Development of Machine Learning Based Analytical Tools for Pavement Performance Assessment and Crack Detection

by

Ju Huyan

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Civil Engineering

Waterloo, Ontario, Canada, 2019

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner **Dr. Yi-Chang James Tsai**

Professor
Georgia Institute of Technology, Civil & Environmental Engineering

Supervisor(s) **Dr. Susan Tighe**

Deputy Provost and Associate Vice-President Int. Planning & Budgeting
Norman W. McLeod Professor in Sustainable Pavement Engineering
University of Waterloo, Civil & Environmental Engineering

Internal Member **Dr. Wei-Chau Xie**

Professor
University of Waterloo, Civil & Environmental Engineering

Internal-external Member **Dr. Alexander Penlidis**

Professor
University of Waterloo, Chemical Engineering

Other Member(s) **Dr. Ningyuan Li**

Adjunct Professor
University of Waterloo, Civil & Environmental Engineering

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included throughout the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapter 4 of this thesis contains parts of a paper co-authored by myself, my supervisor (Dr. Susan Tighe), Wei Li, Liyang Xiao, and Lili Pei. I wrote the paper based on the methodology developed with the cooperation of all the co-authors including myself. All of the co-authors have contributed to the editing and reviewing during the papers production.

Chapter 6 of this thesis has been incorporated into a paper submitted for publication. The paper is co-authored by myself, my supervisor (Dr. Susan Tighe), Wei Li, and Zhaoyun Sun. I developed the methodology and research design and conducted paper writing work. Dr. Tighe and Wei Li help in the methodology design, paper organization and reviewing. Zhaoyun Sun provided addition information regarding the experiments part and helped in the paper reviewing.

Chapter 8 of this thesis has been incorporated into a paper submitted for publication. The paper is co-authored by myself, my supervisor (Dr. Susan Tighe), Wei Li, Junzhi Zhai, Zhengchao Xu, and Yao Chen. I conducted the methodology development, model training, testing, results interpretation, and wrote the paper. Dr. Tighe provide help on the methodology development, paper organization, and conducted paper review. Wei Li, Junzhi Zhai, Zhengchao Xu, and Yao Chen helped on the data collection, hardware system setup, data augmentation and model training.

Chapter 9 of this thesis has been incorporated into a paper submitted for publication. The paper is co-authored by myself, my supervisor (Dr. Susan Tighe), Wei Li, Zhengchao Xu, and Junzhi Zhai. I conducted the methodology development, model training, testing, results interpretation, and wrote the paper. Dr. Tighe provide help on the methodology development, paper organization, and conducted paper review. Wei Li, Zhengchao Xu and Junzhi Zhai helped on the data collection, hardware system setup, data augmentation and model training.

Abstract

Pavement Management System (PMS) analytical tools mainly consist of pavement condition investigation and evaluation tools, pavement condition rating and assessment tools, pavement performance prediction tools, treatment prioritizations and implementation tools. The effectiveness of a PMS highly depends on the efficiency and reliability of its pavement condition evaluation tools. Traditionally, pavement condition investigation and evaluation practices are based on manual distress surveys and performance level assessments, which have been blamed for low efficiency low reliability. Those kinds of manually surveys are labor intensive and unsafe due to proximity to live traffic conditions. Meanwhile, the accuracy can be lower due to the subjective nature of the evaluators. Considering these factors, semi-automated and automated pavement condition evaluation tools had been developed for several years. In current years, it is undoubtable that highly advanced computerized technologies have resulted successful applications in diverse engineering fields. Therefore, these techniques can be successfully incorporated into pavement condition evaluation distress detection, the analytical tools can improve the performance of existing PMSs. Hence, this research aims to bridge the gaps between highly advanced Machine Learning Techniques (MLTs) and the existing analytical tools of current PMSs. The research outputs intend to provide pavement condition evaluation tools that meet the requirement of high efficiency, accuracy, and reliability. To achieve the objectives of this research, six pavement damage condition and performance evaluation methodologies are developed.

The roughness condition of pavement surface directly influences the riding quality of the users. International Roughness Index (IRI) is used worldwide by research institutions, pavement condition evaluation and management agencies to evaluate the roughness condition of the pavement. IRI is a time-dependent variable which generally tends to increase with the increase of the pavement service life. In this consideration, a multi-granularity fuzzy time series analysis based IRI prediction model is developed. Meanwhile, Particle Swarm Optimization (PSO) method is used for model optimization to obtain satisfactory IRI prediction results. Historical IRI data extracted from the InfoPave website have been used for training and testing the model.

Experiment results proved the effectiveness of this method.

Automated pavement condition evaluation tools can provide overall performance indices, which can then be used for treatment planning. The calculations of those performance indices are required for surface distress level and roughness condition evaluations. However, pavement surface roughness conditions are hard to obtain from surface image indicators. With this consideration, an image indicators-based pavement roughness and the overall performance prediction tools are developed. The state-of-the-art machine learning technique, XGBoost, is utilized as the main method in model training, validating and testing.

In order to find the dominant image indicators that influence the pavement roughness condition and the overall performance conditions, the comprehensive pavement performance evaluation data collected by ARAN 900 are analyzed. Back Propagation Neural Network (BPNN) is used to develop the performance prediction models. On this basis, the mean important values (MIVs) for each input factor are calculated to evaluate the contributions of the input indicators. It has been observed that indicators of the wheel path cracking have the highest MIVs, which emphasizes the importance of cracking-focused maintenance treatments.

The same issue is also found that current automated pavement condition evaluation systems only include the analysis of pavement surface distresses, without considering the structural capacity of the actual pavement. Hence, the structural performance analysis-based pavement performance prediction tools are developed using the Support Vector Machines (SVMs). To guarantee the overall performance of the proposed methodologies, heuristic methods including Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are selected to optimize the model. The experiments results show a promising future of machine learning based pavement structural performance prediction.

Automated pavement condition analyzers usually detect pavement surface distress through the collected pavement surface images. Then, distress types, severities, quantities, and other parameters are calculated for the overall performance index calculation. Cracks are one of the most important pavement surface distresses that should be quantified. Traditional approaches are less accurate and efficient in locating, counting and quantifying various types of cracks

initialed on the pavement surface. An integrated Crack Deep Net (CrackDN) is developed based on deep learning technologies. Through model training, validation and testing, it has proved that CrackDN can detect pavement surface cracks on complex background with high accuracy.

Moreover, the combination of box-level pavement crack locating, and pixel-level crack calculation can achieve comprehensive crack analysis. Thereby, more effective maintenance treatments can be assigned. Hence, a methodology regarding pixel-level crack detection which is called CrackU-net, is proposed. CrackU-net is composed of several convolutional, max-pooling, and up-convolutional layers. The model is developed based on the innovations of deep learning-based segmentation. Pavement crack data are collected by multiple devices, including automated pavement condition survey vehicles, smartphones, and action cameras. The proposed CrackU-net is tested on a separate crack image set which has not been used for training the model. The results demonstrate a promising future of use in the PMSs.

Finally, the proposed toolboxes are validated through comparative experiments in terms of accuracy (precision, recall, and F-measure) and error levels. The accuracies of all those models are higher than 0.9 and the errors are lower than 0.05. Meanwhile, the findings of this research suggest that the wheel path cracking should be a priority when conducting maintenance activity planning. Benefiting from the highly advanced machine learning technologies, pavement roughness condition and the overall performance levels have a promising future of being predicted by extraction of the image indicators. Moreover, deep learning methods can be utilized to achieve both box-level and pixel-level pavement crack detection with satisfactory performance. Therefore, it is suggested that those state-of-the-art toolboxes be integrated into current PMSs to upgrade their service levels.

Acknowledgements

I would like to offer my sincere acknowledgements and gratitude to many people who have supported me during this research. Those continuous trust, guidance and encouragement from my supervisor, team members, colleagues, friends and families are the main strength driving me through this important life stage.

I would like to acknowledge my supervisor Dr. Susan Tighe, who has provided tremendous guidance and support for my research which are irreplaceable. She is a true example of inspiration and intellect, which are always combined with her persistence and leadership. I was and will be truly honored to be her student for my whole lifetime.

I would like to give special thanks to Dr. Wei Li, who is in the School of Information Engineering in Chang'An University, China. He has always been encouraging and helping me when I faced with difficulties.

I would like to thank the project team members of York University: Dr. Jianguo Wang, Dr. Boxing Hu and the graduate students of their research team: Gula Abdullahi and Won Mo (Andy) Jung. Special thanks to the team members at MTO: Dr. Ningyuan Li (Senior Pavement Management Engineer, Pavements and Foundations Section, MTO) and Gumisiriza Gideon. Under all their cooperation we completed the Highway Infrastructure Innovation Funding Program (HIIFP) 2016: Calibration and Verification of Pavement Surface Images) Assessment and Improvement of MTO's Imaging Processing Systems for Usage in Pavement Management.

I would also like to thank all the members of the Centre for Pavement and Transportation Technology (CPATT) group (both current and the former), who have always provided immediate help when I needed. Special thanks to Dan Pickel, Eskedil Melese for their help during the project data collection in June 2017, and Raha Wafa for showing me the use of Surpro. Special thanks also to Abimbola Grace Olaleye, Ata Nahidi, Bingqian (Tracy) Zhou, Dandi Zhao, Frank (Yang) Liu, Frank Mi-Way Ni, Guangyuan(Luke) Zhao, Qingfan Liu and Taher Baghaee Moghaddam for their help in the methodology discussion and explanations.

I would also like to thank to the cooperation from Chang' An University, China. Those to be special thanked are: Dr. Zhaoyun Sun; PhD students: Lili Pei, Junzhi Zhai, Zhengchao Xu; and master students: Yao Chen, Zhidan Ma, Yao Shen, Minghang Ding, Ranran Deng for their help in the model training and optimization.

Further thanks to my officemates (both former and present): Dan Pickel, Sunny Singh, Yashar Alamdary, and Zaid Alyami for creating a respectful, clean and nice working environment. Also, thank Jessica Rossi, Bev Seibel, Victoria Tolton and Zoe Tipper for their generous help and patience with me.

Moreover, I would like to express my heartfelt thanks to my dear parents: Jianrong Huyan, Shuling Gao; and my dear brother and sister: Yingcai Huyan and Feng Huyan, for their endless support, love and help during this study.

Dedication

This thesis is dedicated to my partner and family, whose support, patience, and love have made this a possibility.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI – Artificial Intelligence

ARAN – Automated Road Analyzer

ARIMA – Autoregressive Integrated Moving Average

BPNN – Back Propagation Neural Network

CART – Classification and Regression Tree

CNN – Convolutional Neural Network

CrackDN – Crack Deep Network

DBNs – Deep Belief Networks

DCNN – Deep Convolutional Neural Network

DHT – Discrete Haar Transformation

DIP – Digital Image Processing

DLTs – Deep Learning Techniques

DMI – Distress Manifestation Index

DT – Decision Tree

DWT – Discrete Wavelet Transformation

Fast-RCNN – Fast Region-based Convolutional Network

Faster-RCNN – Faster Region-based Convolutional Network

FSM – Finite State Machine

FWD – Falling Weight Deflectometer

GA – Genetic Algorithm

GD – Gradient Descent

GDBT – Gradient Descent Boosting Tree

GPR – Ground Penetrating Radar

GPU – Graphic Processing Unit

HWD – Heavy Weight Deflectometer

ILSVRC – Image Large Scale Visual Recognition Challenge

IOU – Interest Over Region

IRI – International Roughness Index

K-NN – K-nearest Neighbor

KPIs – Key Performance Indicators

LCCA – Life Cycle Cost Analysis

LTPP – Long - Term Pavement Performance

LWD – Lightweight Deflectometer

LWP – Left Wheel Path

MAE – Mean Absolute Error

MIV – Mean Impact Value

MLP – Multilayer Perception

MLS – Moving Least Square

MLTs – Machine Learning Techniques

M&R – Maintenance and Rehabilitation

MSE – Mean Squared Error

MTO – Ministry of Transportation of Ontario

PAC – Probability Approximately Correct

PCA – Principal Component Analysis

PCI – Pavement Condition Index

PMS - Pavement Management System

PSO – Particle Swarm Optimization

RBF – Radial Basis Function

RBMs – Restricted Boltzman Machines

RCI – Riding Comfort Index

R-CNN – Region with CNN features

RE – Root Error

ReLU – Rectified Linear Unit

RF – Random Forest

RMSE – Root Mean Squared Error

ROC – Receiver Operating Characteristic Curve

RPL – Riding Performance Level

RPRN – Region Proposal Refinement Network

RPN – Region Proposal Network

RWP – Right Wheel Path

SAI – Structural Adequacy Index

SBL – South Bend Left

SBR – South Bend Right

SDI – Surface Distress Index

SSD – Single Shot MultiBox Detector

SVM – Support Vector Machine

TAC – Transportation Association of Canada

XGBoost – Extreme Gradient Boosting

2D – Two-dimensional

3D – Three-dimensional

CHAPTER 1. Introduction

1.1. Background

A Pavement Management System (PMS) can be defined as a set of tools or methods that assist decision makers in finding optimum strategies for providing, evaluating, and maintaining pavements in a service condition over a period of time (Huang, 1993). Similar objectives are often noted for other management systems. The reliability of the performance prediction and the usability of the maintenance suggestions depend on the accuracy of pavement condition evaluation, which should be able to provide an objective rating of pavement condition.

After initial construction, the pavement is open to traffic. The pavements should undergo periodic maintenance treatments throughout the life cycle in order to achieve long time service. Those treatments mainly include routine maintenance, preservation, minor and major rehabilitations. Figure 1.1 is a basic illustration of the relationship between different treatment activities and the deterioration of the pavement overtime. There is an overlap between different treatment methods, which is influenced by the complex considerations of treatment planning.

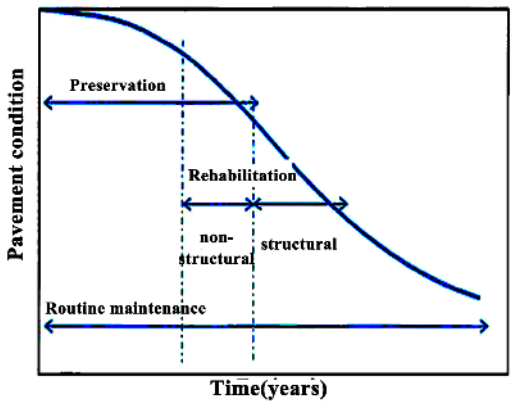


Figure 1.1: Pavement treatments with the deterioration of pavement condition (Tighe, 2013)

The specific pavement treatment methods are selected based on the results of the condition evaluation, performance prediction, and other considerations. Determining the current performance condition of the pavement is the first step towards effective treatment arrangements. Therefore, the development of highly reliable automated pavement condition evaluation tools benefits not only the pavement condition evaluation departments, but also the maintenance implementation and capital allocation agencies. Pavement condition data usually include different types of distresses, roughness, rutting, skid resistance, and structural capacity (AASHTO, 2008). Usually, raw data from original collection devices are incomplete, ambiguous, and contain excessive noise. Moreover, the time intervals and methods for data collection vary among each data collection activity, and the environmental conditions are significantly different from each region. All these factors make the development of highly reliable pavement condition evaluation tools challenging but valuable.

Pavement surface distresses are the results of the joint effects of environmental influences, pavement aging, and traffic loading. As important indicators of pavement performance over time, pavement surface distresses are commonly evaluated manually. However, considering the many problems induced by this traditional method, many agencies have replaced manual evaluations with automated and semiautomated devices for pavement performance data collection and evaluation. The automated pavement condition survey can be considered as systems that include automated pavement condition data collection, pavement distress identification and quantification using software and computer-based technologies. These kinds of techniques have been widely used by many highway agencies to replace traditional manual condition surveys. Considering the budget aspect, data collected from automated pavement condition surveys are worth millions of dollars, and evidently, the quality of collected data are crucial to the performance of PMS and, in turn, to the optimal budgeting and project prioritization of the agency.

High quality pavement condition data collections are important to the reliability of maintenance suggestions output by a PMS. But current data collection methods usually suffer from two main limitations: 1) Having acceptable accuracy but at a very high cost (automated vehicles based), or 2) less expensive but of lower accuracy while being time-consuming because of using manual collection, and possibly with safety concerns. These problems make

it difficult for agencies to make a choice of conducting pavement investigations. Visual inspection of pavement conditions by subjective human experts is the simplest method yet subjective. Moreover, this kind of approach requires high labor involvements, intensive time arrangement, and exposes the inspectors into dangerous working situations, especially when working on highway areas. Recently, both Destructive Testing (DT) and Non-Destructive Testing (NDT) have been used to improve the pavement performance data collection, which can provide additional data for the PMS. An automated PMS can detect and classify diverse types of pavement distress efficiently, regardless of weather conditions. Most road maintenance departments and transportation agencies have started to use the automated pavement management and assessment systems, leading to an increase in the development and utilization of computer vision methods for pavement engineering applications. The developed procedures can be combined with the use of intelligent transportation systems, which can detect and identify vehicle types (e.g., heavy vehicles like tractor trailers), together with the monitoring of traffic intensity, for example, to help identify roads where more frequent pavement status surveys are required.

Deep Learning is a state-of-the-art technology that has witnessed eye-catching advancement in recent years. The main implementation method of Deep Learning is via Deep Neural Networks, which use several hidden layers for multi-level feature extraction. The hierarchy features of the input image can then be obtained for further interpretations. As one important branch, Deep Convolutional Neural Network (DCNN) (Sun, Wang, & Tang, 2013) adopted the methodology of "weights sharing", which made the object recognition more efficient when using this technique. In traditional methods, the images input into a system for feature extraction without any pre-processing being conducted can rarely provide satisfactory results. However, the DCNN technique made it possible for an end-to-end image feature extraction. The reason is that the highly robust characteristics of DCNN in terms of scaling, tilting, etc. makes this architecture much more flexible for practical applications compared to the traditional approaches. The significance of the developed novel technologies lies in their serviceability to engineering practice. Furthermore, in light of the rapid development of mobile devices, incorporating machine learning techniques, especially deep learning technologies, into novel pavement condition evaluation tools is required from municipalities and agencies. Therefore,

bridging the gap between the rapid developing computer technologies and the current pavement condition survey and rating methods is very desirable.

1.2. Problem Statement

During the data collection process, the 2D image acquisition device projects the 3D road entity into a 2D plane by spatial transformation methods. It cannot directly acquire the 3D indicators of road performance, such as rutting and longitudinal profile. Therefore, conventional road condition evaluations based on 2D images can only be used to evaluate the surface damage, including various types of cracks, potholes, and so on. This mechanism tends to lose a large amount of 3D depth information on the pavement. However, with the increase of the impacts from the traffic, environment and human behavior, the load transfer capability of the pavement changes rapidly. The changes of road structure condition can be reflected in the changing of the 3D depth profiles, which can result in an uneven road surface. This phenomenon has a great impact on road users, and therefore, brings serious safety threat to drivers and passengers. Moreover, more serious road damage can initiate, which leads to early failure of the pavement. Therefore, it can be beneficial for all sides in analyzing the relationship between 2D image indicators and the 3D depth condition of the pavement.

Cracks can be considered as the most commonly seen pavement surface defects, which can lead to almost any kind of serious distresses without proper repairing operations. Therefore, efficient detection of cracks initiated in the pavement surface and conducting timely maintenance operations before they propagate into structural related permanent failure are extremely important for pavement management agencies. Researchers have put great effort into the development of crack detection algorithms, especially with respect to digital image processing-based crack detection and classification. From 2D grayscale image processing to 3D pavement depth image analysis, the image based cracking detection approaches have seen significant achievements over the last decades. However, those methods can either deal with crack images that are restricted to specific requirements (camera shooting angle, homogeneous background, etc.), or can only analyze a single image at a time. The emergence of the latest Deep Convolutional Neural Network (Shi, Cui, Qi, Ming & Chen, 2016; Xu, Lou, Wang,

Gilmore & Madabhushi, 2016) technologies and their promising applications in other fields, such as navigation, traffic planning, image recognition, object detection, etc., have given innovative insights to the pavement crack detection research. Hence, the development of DCNN based pavement crack recognition tools is undoubtedly meaningful in terms of filling the gap between the advanced technologies and the less intelligent crack analytical tools.

1.3. Research Hypotheses

Based on the research gaps observed from the literature review, this research is intended to be carried out based on the following hypothesis:

(1) Timely, accurately pavement condition investigation and assessment are significant for the efficient supervising of current in-service pavements, thereby benefit pavement management municipalities and agencies.
(2) Manual pavement condition investigations are time consuming and less detailed than automated pavement condition survey and evaluation.
(3) Pavement condition evaluation tools in current automated PMSs should be updated in pace with rapidly developed technologies to enhance the efficiency and accuracy of the systems.
(4) Machine Learning Techniques (MLTs), especially the recently developed Deep Learning techniques, have great potential of being integrated into automated pavement condition investigation and evaluation tools to improve the overall performance of the system.

1.4. Research Objectives

This research developed pavement performance prediction models and crack detection tools based on data collected from multiple devices. The performance data used in this research are collected by ARAN-7000/9000, Lightweight Deflectometre (LWD), Rival Solutions, and InfoPave. The crack image data are collected by automatic pavement survey vehicles,

smartphones and action cameras. The development of signature image-based pavement cracking detection methods is within the scope of this research. The innovative machine learning technologies are the key methods adopted for model development and optimizations. Moreover, recommendations for usage of the proposed pavement condition investigation and evaluation tools are provided.

The detailed objectives of this research are to:

(1) Develop the analytical tools for IRI prediction based on historical data.

(2) Develop the pavement roughness and the overall performance indices prediction methods based on pavement indicators.

(3) Figure out the dominant image indicators, which have the most significant impacts on the roughness condition and the overall performance of the pavement.

(4) Propose pavement structure performance prediction tools based on pavement surface modulus condition data.

(5) Develop the box-level pavement surface crack detection tools based on crack images collected using diverse devices

(6) Develop the pixel-level pavement crack analytical tools using deep learning technique.

1.5. Thesis Organization

Based on the research conducted throughout the whole research period, the contents of this thesis include ten chapters. Each chapter forms an independent aspect and some chapters are thus related. The first three chapters provide the background information regarding the research of this thesis, the literature and the overall methodologies being developed. Then Chapters 4-7 mainly consist of the methodologies, experiments and results analysis of the structured data based (pavement performance data) analysis. Then, Chapters 8 and 9 are the non-structured data (pavement surface images) based analysis. Finally, the conclusions of the whole research

conducted in this thesis are included in Chapter 10. To be specific:

Chapter 1 introduces the research background, research gaps, and objectives.

Chapter 2 is the literature part. Literature regarding this research is discussed. Pavement surface distresses and their measurements are introduced. Pavement surface deformation measurement indicators, Key Performance Indices (KPIs), automatic pavement condition inspection and performance assessment systems & methods are discussed as well. Meanwhile, the current engineering application condition of machine learning technologies are discussed.

Chapter 3 introduces an overall methodology framework fulfilling the objectives of this thesis.

Chapter 4 focuses on the modeling of the International Roughness Index (IRI) prediction.

Chapter 5 and 6 are carried out based on pavement condition data collected from ARAN-9000. Chapter 5 aims to propose the pavement roughness condition and the overall indices prediction models based on the 2D image indicators. Chapter 6 aims to provide engineering practices with the insights of the dominant pavement surface distresses that have significant influences on the overall pavement performance.

Chapter 7 develops the pavement structural performance prediction model based on pavement surface modulus data collected by LightWeight Deflectometer (LWD).

Chapter 8 proposes the deep convolutional neural network-based pavement surface crack detection architecture. Different kinds of pavement surface crack images are used as the input data for model training and testing. This research aims at evaluating the effectiveness of the deep learning-based pavement crack detection.

Chapter 9 develops the pixel-level pavement surface crack segmentation architecture utilizing deep learning technology. This chapter aims at addressing whether and how well deep neural networks can be used for pixelwise crack analysis.

Chapter 10 is the conclusion and recommendations part. All the findings of this thesis are discussed and summarized in this part, along with recommendations for future work.

CHAPTER 2. Literature Review

2.1. Pavement Performance Measurements

2.1.1 Pavement Surface Distress

The failure of a pavement structure usually results from the combination of internal or external influences which have harmful impacts on the whole pavement structure. Surface distresses in a pavement can be caused by a number of factors such as heavy traffic, environmental effects, bad materials and poor construction. Overall pavement performance KPIs are calculated based on the weighting of various pavement distresses. In the 1980s, the Ministry of Transportation of Ontario established the first Pavement Management System (PMS). Limited by the high computational cost and the less advanced computer technologies, the early version of PMS used data from manual surveys. Based on performance levels, maintenance and rehabilitation strategies were developed. After several years of improvements, MTO upgraded the original PMS to PMS-2, which involved comprehensive road performance data and therefore, integrated more up-to-date distress rating principles for the overall performance evaluation. In terms of surface distress classification, MTO's PMS-2 defined and evaluated 15 surface distresses which can be seen in Table 2.1.

It can be concluded that pavement distresses are categorized by three groups, which include (1) cracking, i.e. fatigue cracking, transverse cracking, and longitudinal cracking (2) surface distresses including flushing, raveling, rippling, and (3) surface deformation, including rutting and shoving.

I. Fatigue Cracking

The formation of fatigue cracking can be considered as the result of two main reasons. Firstly, the interconnection of cracks at the surface, which are initiated as a result of excessive tensile stress beneath the asphalt layer. These can eventually result in one or more longitudinal cracks

at the top surface. This process is commonly named as ‘bottom up’ or ‘classical’ fatigue cracking. The cracks can also be caused by repetitive traffic loads, where tensile stress leads to fatigue cracking that has patterns similar to the back of an alligator, so fatigue cracking is also referred to as “alligator cracking”. Figure 2.1 shows the general measurement method of fatigue cracking. The cracked areas of different severity levels are measured, usually using the unit of squared metre. Low severity (Figure 2.2a) fatigue cracking indicates the affected area has few connecting cracks, which are not spalling, and the pumping phenomenon is not evident. Moderate severity (Figure 2.2b) represents a fatigue cracking with complete pattern, slightly spalling, but no evident pumping. High severity (Figure 2.2c) fatigue cracking indicates the cracked area has an interconnected cracking pattern, which is often caused by traffic and shows signs of pumping.

Table 2.1: Pavement surface distresses defined by MTO PMS-2 (Chong., et al.,1989)

No.	Distress Type
1	Raveling and Coarse Aggregate Loss
2	Flushing
3	Rippling
4	Wheel Track Rutting
5	Distortion and Shoving
6	Longitudinal Wheel Track: Single and Multiple
7	Longitudinal Wheel Track: Alligator
8	Centreline: Single and Multiple Cracking
9	Centreline: Alligator Cracking
10	Pavement Edge: Single and Multiple Cracking
11	Pavement Edge: Alligator Cracking
12	Transverse: Half, Full and Multiple Cracking
13	Transverse: Alligator Cracking
14	Longitudinal Meandering and Midlane Cracking
15	Random Cracking

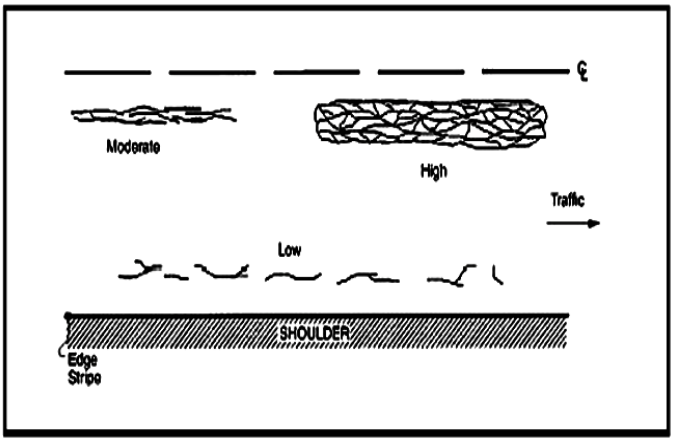


Figure 2.1: Fatigue cracking measurement (Miller & Bellinger, 2014)

(a) Low severity **(b) Moderate severity**

(c) High severity

Figure 2.2: Fatigue cracking on the pavement surface (Chong., et al.,1989)

II. Transverse Cracking

Transverse cracks usually extend across the pavement centerline perpendicular to the traffic flow. Transverse cracks can be divided into two categories: temperature-induced and load-induced transverse cracks. The temperature changes and aging related shrinkage of the asphalt binder leads to the layer's movements and these can cause temperature-induced transverse cracks. Load-induced transverse cracks can propagate on the overlays that have unfilled joints or gaps. These cracks allow freeze-thaw from water penetrating the pavement under-layers, which therefore cause more serious damages. Meanwhile, the rideability of the pavement decreases because of the increase in the surface roughness. Transverse cracking is measured by recording the number of occurrence and the total length of all cracks in different severity levels (Figure 2.3). The severity levels are classified as low severity (mean cracking width <6mm) (Figure 2.4a), moderate severity (mean cracking width exceeds 6mm but is less than 19mm) (Figure 2.4b), and high severity (mean cracking width exceeds 19mm) (Figure 2.4c).

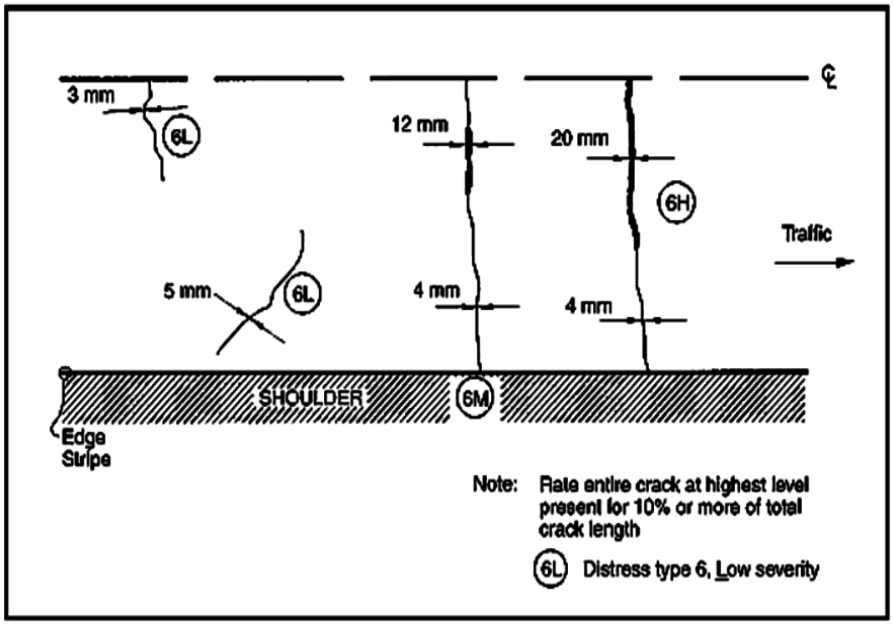


Figure 2.3: Transverse cracking measurements (Miller & Bellinger, 2014)

(a) Low severity **(b) Moderate severity**

(c) High severity

Figure 2.4: Transverse cracking on the pavement surface (Chong., et al.,1989)

III. Longitudinal Cracking

Longitudinal cracking is when cracks propagate along the traffic direction, hence perpendicular to the transverse cracking. The severity levels are classified based on the same principle (Figure 2.5) as transverse cracking. According to the occurrence location difference, longitudinal cracking can be classified into wheel path longitudinal cracking and non-wheel path longitudinal cracking. Both wheel path (Figure 2.6a) and non-wheel path (Figure 2.6b) longitudinal cracking are recorded by the length with the unit of metre for every severity level.

Low severity cracks are less than 6mm in width while high severity cracks can be wider than 19mm.

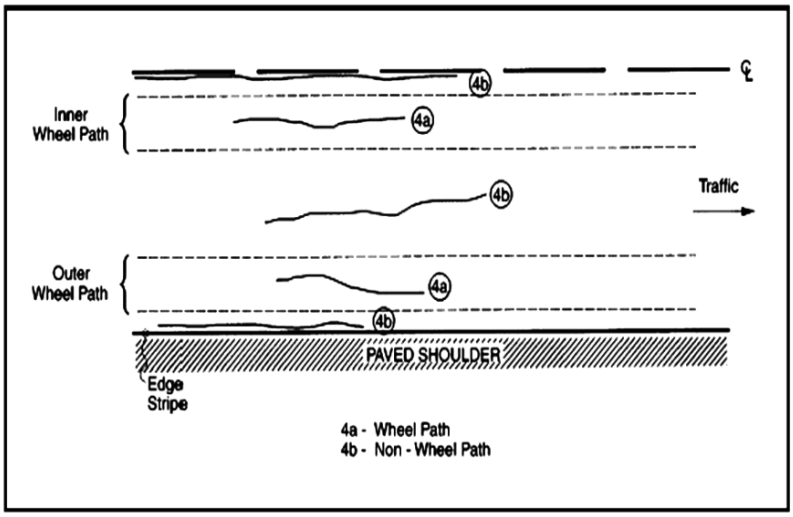
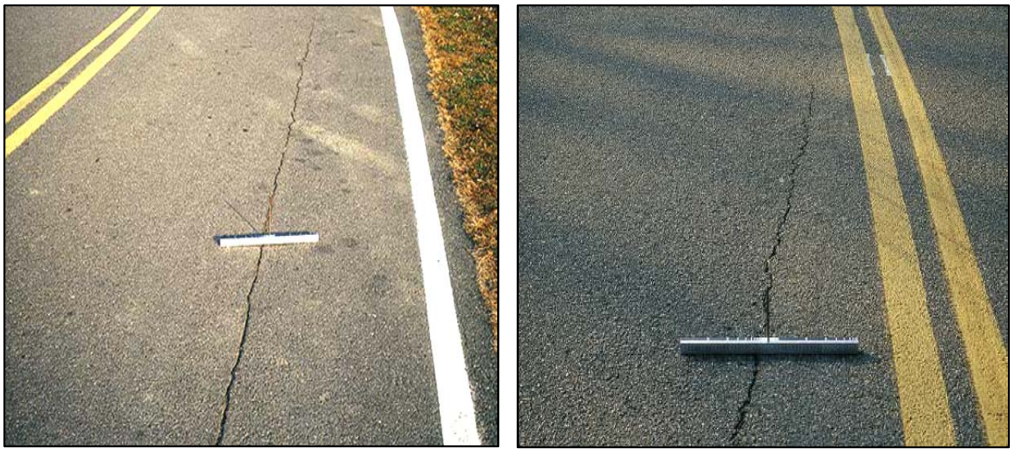


Figure 2.5: Longitudinal cracking measurements (Miller & Bellinger, 2014)



(a) Non-wheel path longitudinal cracking (b) Wheel path longitudinal cracking

Figure 2.6: Longitudinal cracking on the pavement surface (Chong., et al.,1989)

IV. Potholes

Potholes are small, bowl-shaped defects on the pavement surface that penetrate all the way through the HMA layer down to the base layers. This kind of commonly seen distress generally

has sharp edges and vertical sides near the top of the hole. Many reasons contribute to the formation of potholes, such as the propagation of crocodile cracking which is featured by the interlocked and fatigue fractures. The condition becomes more serious in cold regions where water tends to freeze, thus providing the strength increasing tendency of the inner soil. This phenomenon usually speeds up the initial crack to propagate. And this leads to pothole formation.

Potholes occur on HMA surfaces (see Figure 2.7 and 2.8). The presence of water inside these potholes can further lead to serious damage such as washing the larger areas of the pavement when several vehicles passing through. However, they can expand if not treated in time. Severe accidents tend to happen due to the invisibility of the potholes to the drivers, thus causing hazards both to road users and their vehicle suspension systems. Pothole patching is mostly conducted in order to repair the imperfection. Potholes are measured by recording the number as well as the square metres of the areas for every severity level. The depth is recorded as the maximum depth with reference to the pavement surface. Low severity potholes have depths which are less than 25 mm; moderate severity potholes have measured depth that range from 25mm to 50mm; high severity potholes refer to those that have measured depths more than 50mm.

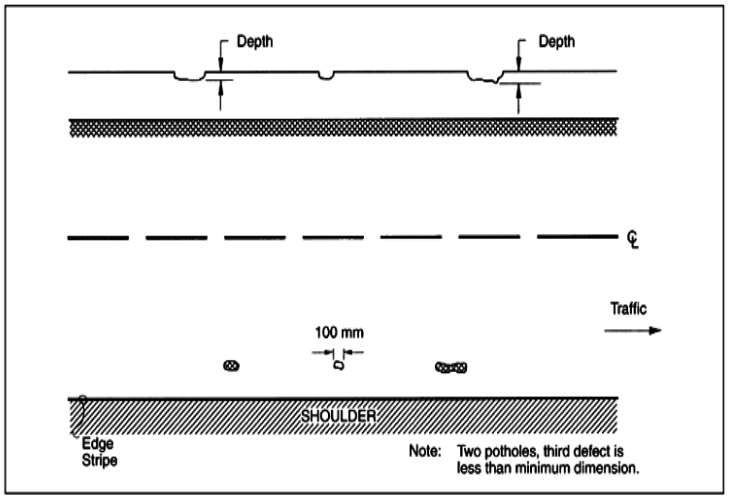


Figure 2.7: Pothole measurements (Miller & Bellinger, 2014)

(a) Low severity

(b) Moderate severity

(c)Moderate severity

(d) High severity

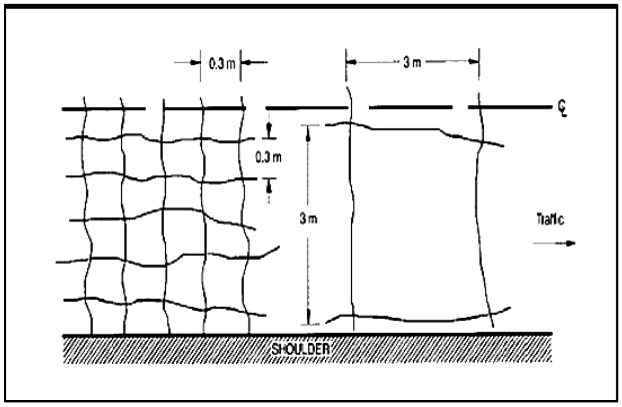
Figure 2.8: Potholes on the pavement surface (Miller & Bellinger, 2014)

V. Block Cracking

Block cracking is more severe to handle as compared to the single longitudinal cracks. In block cracking, a series of cracks are interconnected to each other and shape into a rectangular form. These cracks are simply connected to the longitudinal cracks which are perpendicular to the transversal path of a pavement, thus dividing the pavement into several pieces (Figure 2.9). The possible reasons might be due to the shrinkage of asphalt that has poor resistance against varying temperature conditions. The closer spaces among the cracks indicate a more advanced

aging caused by shrinkage and hardening of the asphalt with time. Block cracking must be repaired in a timely fashion, otherwise it can accelerate moisture infiltration and roughness. The same strategies are adopted in this deterioration process as done in the longitudinal cracking. Block cracking is measured by recording the square metres of the cracked area for each severity level as shown in Figure 2.9b. The severity of block cracking is classified into three levels: low severity indicates the mean crack width does not exceed 6 mm (millimetre); moderate severity indicates the mean crack width ranges between 6mm and 19mm; and high severity indicates the mean crack width exceeds 19mm.

(a) Block cracking



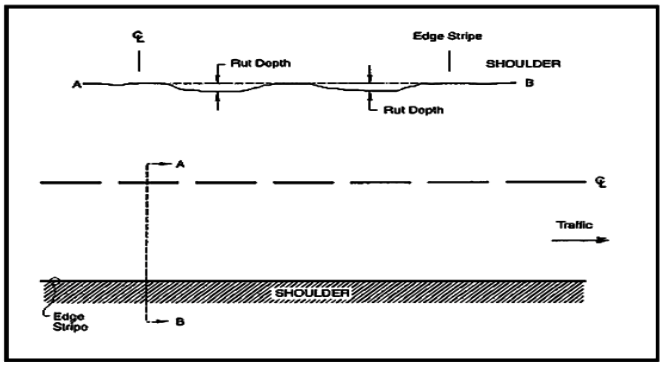
(b) Block cracking measurement

Figure 2.9: Block cracking and measurements (Miller & Bellinger, 2014)

2.1.2 Rutting Depth

Pavement permanent deformation, which is also called rutting, can be defined as “longitudinal depressions in the wheel paths after repeated loadings, combined with a sideways shoving of the pavement material” (Chatrchyan, Khachatryan, Sirunyan, Tumasyan, Adam, Aguilo, & Friedl, 2012). It has a significant effect on pavement performance. Optimally designed pavement combined with periodic repairing activities can reduce the occurrence of rutting on pavement surface. Hence, pavement service life can be prolonged. Figure 2.10a shows commonly seen rutting, regarding which manually measurement methods usually employ a 1.2m straight edge with the depth being recorded. Other methods include using Dipstick profiler, SurPro, etc., to automatically measure the IRI values of a certain pavement section.

(a) Rutting on the pavement surface



(b) Rutting measurements

Figure 2.10: Rutting and measurements (Miller & Bellinger, 2014)

2.1.3 Roughness

A profile refers to the vertical deviations of the road surface along a longitudinal line of traveling in the wheel path. This profile of the pavement measured in the longitudinal direction is important because iterations help to interpret the levels of rutting, roughness and other related distresses.

Profilometers have been used to measure the smoothness, or roughness, of the road surface. Based on previous research, the smoother the pavement, the better the ride quality. Hence, less energy consumption and environmental pollution can be achieved. Therefore, the International Roughness Index (IRI) has been developed and used worldwide for measuring the general roughness of the road surface. IRI is defined by the International Road Roughness Experiment (Sayers, 1987) as a mathematical simulation of a passenger vehicle suspension system. Once proposed, IRI has been used to evaluate the roughness levels of any pavement type. IRI values normally vary from 0 mm/m (m/km) and 4 mm/m (m/km). The 0 mm/m (m/km) indicates an ideal smooth surface which is defined by initial construction. IRI values higher than 4mm/m (m/km) indicate super rough surface roads that need maintenance operation. Based on the accuracy and reliability of the measured IRI values using different devices and methods, research conducted by the World Bank has aimed at standardizing the data collection and analysis techniques. According to diverse types of methods, the IRI measurement devices are classified into four categories as listed in Table 2.2.

In Table 2.2, four classes of IRI data collection devices are cited, which are the most commonly used measurement methods. The accuracies among those devices vary significantly. Class 1 methods can collect the perfect pavement profile data; hence they are the most accurate. In contrast, Class 4 measures pavement profile in the roughest ways, which is subjective. That means the accuracy of collected data decreases from Class 1 to Class 4. With the development of computerized technologies, several specifications and standards have been developed to improve the correctness and repeatability of IRI measurements. Widely used devices are the inertial profilers that use height sensor technology to measure the roughness. The concurrent benefits brought by this method are that it can measure roughness of the highly textured HMA

or PCC pavement surfaces.

Table 2.2: World Bank Roughness Measurement Classification (Sayers, 1987)

Class 1	***Precision Profiles*** ***(most accurate)***
	Rod and level, Total Station, Digital Incremental Profiler, Rolling Surface Profiler
Class 2	***Other Profilometric Devices***
	Inertial Profilers, Profilographs
Class 3	***Response Type Devices***
	Mays Ride Metre (vehicle or trailer version), Portable Universal Roughness Device (PURD)
Class 4	***Subjective Ratings*** ***(least accurate)***
	Panel Ratings (Riding Comfort Index, Present Serviceability Rating or Index)

Since 1997, MTO has started to use the International Roughness Index (IRI) for measurement of pavement roughness. IRI is computed from a longitudinal profile measurement using quarter car simulation at a speed of 80 k m /h. The IRI is a measurement scale for pavement roughness based on the response of a generic motor vehicle to a single longitudinal profile of the road surface. The IRI was initially developed in 1986 based on the results of the international road roughness experiments held in Brazil in 1982. Since then, the IRI has become a well-recognized standard for measuring road roughness.

2.2. Pavement Key Performance Indices (KPIs)

Highways systems use several Key Performance Indices (KPIs), such as PCI, DMI, RCI, to evaluate the performance of pavement conditions. Each KPI can be used individually to reflect

the performance of the tested pavement section. PCI is usually considered as a comprehensive indicator, which is calculated based on DMI and IRI. The selection of KPIs for a specific road section are based on data availability, the requirements of the pavement management department and the corresponding budget allocation. After the quantity evaluations, specific pavements maintenance and rehabilitation (M&R) operations can be scheduled for cost-effective pavement management. In Ontario, the most widely used KPIs are shown in Table 2.3.

Table 2.3: KPIs in MTO PMS-2

Key Performance Index	**Measurement description**
International Roughness Index (IRI)	Assessing the riding quality of pavement in terms of roughness condition
Pavement Condition Index (PCI)	Evaluation of the overall pavement surface condition, structural strength and functional capability of the pavement.
Distress Manifestation Index (DMI)	Measuring the overall surface condition of the pavement
Riding Comfort Index (RCI)	Evaluating the riding quality of pavement in terms of user comfortableness condition
Rutting depth (RUT)	Measuring the transverse profile of the pavement. Rut depth is measured in both left and right wheel paths

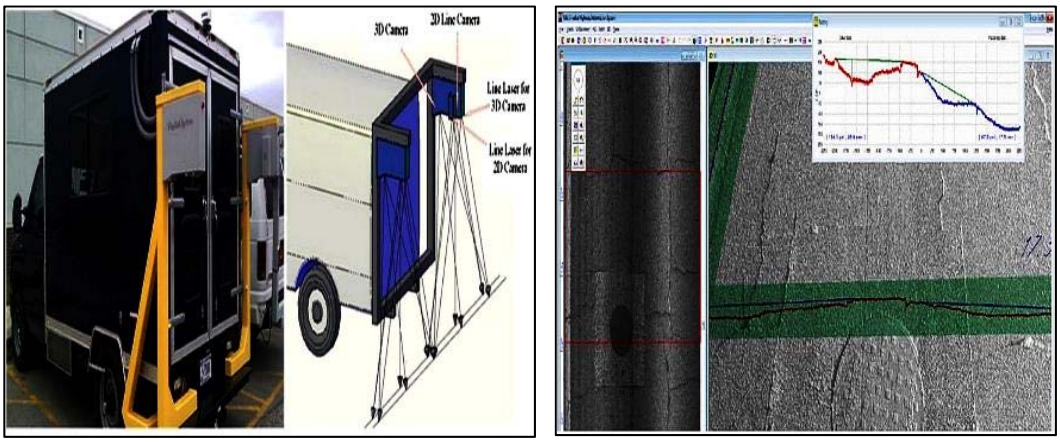
2.3. Pavement Distress Detection and Condition Evaluation

2.3.1 Automatic Pavement Distress Detection Systems

Automated pavement condition surveys and evaluation techniques have been examined since 1990, when a general framework of automated pavement surface distress characterization was proposed (Haas & Hendrickson, 1990). Then, automated crack sealing and path planning methodologies are developed assuming crack maps have been acquired (Kim, Haas, & Greer, 1998). In the1990s, researchers led by Cheng from Utah State University cooperated with Utah DOT to develop the automated pavement distress analysis systems (Cheng & Miyojim, 1998). After several years of development, their systems have upgraded from the original low speed, low accuracy systems to relatively high speed but still less accurate systems due to poor quality of images acquired. Another research team made great contributions to the development of automated pavement distress detection systems is Wang's research team at Oklahoma State University. Wang's research team that have conducted intensive work in developing 3D detection method for the pavement surface distresses. During the research period, they developed a high-resolution line laser imaging system, which can achieve 1mm crack detection (Figure 2.11). However, their system has specific requirements for pavement surface condition and pavement type (rural roads or urban roads). What's more, the system is expensive for small scale companies, even though the system has been commercialized. Tsai's research team, from Georgia Institute of Technology, is another important branch of investigating automated pavement distress analysis. They have made significant contributions since 2010, when 3D line laser imaging techniques were introduced for pavement cracking detection (Tsai & Li, 2012) methodologies (Figure 2.12). After that, they made continuous improvements for the accuracy of their cracking detection algorithms (Tsai, Jiang, & Huang, 2012; Tsai, Jiang, & Wang, 2014; Tsai, Wu, & Lewis, 2013). However, significant false-positive and false-negative errors still remains as challenging issues.

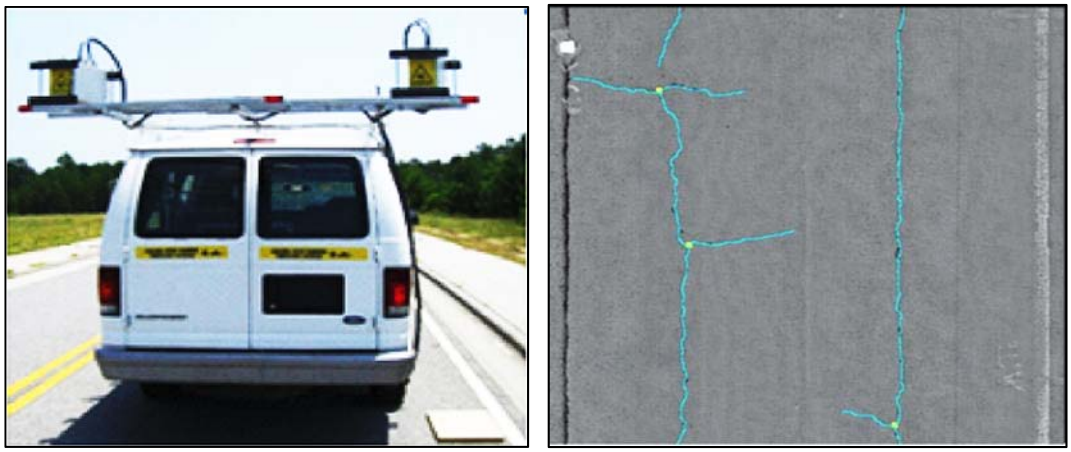
Overall, although almost all the research teams have put enormous effort into the crack analysis, they are still facing problems and trying to find more intelligent, accurate, robust and reliable

pavement distress detection algorithms to optimize the systems. However, none of those automated pavement surface condition detection and evaluation systems have been successfully integrated with artificial intelligent algorithms, i.e., machine learning techniques, especially recently developed Deep Learning techniques, for high efficiency pavement condition assessment. Thus, this should be an promising alternative for consideration.



(a) Hardware system of PaveVision 3D ultra **(b) Software system**

Figure 2.11: 3D ultra-system for data collection (Qiu, Wang, Li, & Moravec, 2018)



(a) Vehicle collecting data **(b) An example of detected cracking**

Figure 2.12: Georgia Tech's sensing vehicle for data collection (Tsai, 2014)

2.3.2 Image Based Pavement Surface Crack Detection

In recent years, departments of transportation are more interested in investigating automatic systems for road condition evaluations (Zakeri, Nejad, Fahimifar, Torshizi, & Zarandi, 2013). The primary objective of these methodologies is achieving intelligent non-destructive pavement condition evaluation and assessment employing innovative techniques. The novel techniques are characterized by digital image processing techniques (Huang, Liu, & Sun, 2014), sensor technologies (Chapeleau, Blanc, Hornych, Gautier, & Carroget, 2014), laser light systems, ground penetration radar (GPR) and hybrid systems. Researchers have proven that embracing those technologies can significantly improve the overall performance of the pavement condition evaluating systems (Adu-Gyamfi, Okine, Garateguy, Carrillo, & Arce, 2011). Most of those road crack assessment systems are based on traditional machine vision and image processing techniques (Chua & Xu, 1994). However, huge challenge remains for the accurate crack detection and quantification due to the complex pavement conditions (Cord & Chambon, 2012, Nishikawa, Yoshida, Sugiyama, & Fujino, 2012, Zhang, Sha, Sun, & Gao, 2009). Though some achievements have been made in more advanced road crack detection, most of them are built based on complex computational algorithms. After automatic pavement crack image acquisition, the general crack computing methods include image preprocessing, crack object segmentation, and feature extraction. A body of research has been conducted by scholars regarding each procedure of the digital image processing-based crack analysis. A summary of some significant literature efforts, including their pros and cons, is shown in Table 2.4.

Table 2.4: Summary of literature for image-based crack detection

Method type	Method	Main purpose/key method	References	Pros	Cons
Image pre-processing	Point operation	Noise reduction	(Yao, Zhao, Yao, & Xu, 2015)	Novel imaging system for removing noise	Application based, not robust.
	Spatial operation	Smooth the texture and enhance the linear features	(Gavilán, Balcones, Marcos, Llorca, Sotelo, Parra, Ocaña, Aliseda, Yarza, & Amírola, 2011)	Facilitated pavement crack detection and classification	Application based, not robust.
	Transformation equation based	Electromagnetism-like Mechanism	(Zakeri, Nejad, & Fahimifar, 2017)	Both local and global condition is considered	Computational complexity
	Image enhancement	Enhancement and segmentation	(Zhou, Yang, & Zhu, 2010)	Illumination invariant	Computational complexity
	Entropy based	Grey entropy method for image contrast characterization	(Li, 2013)	More effective compared to the traditional method for road crack detection	Computational complexity
	Multi-scale analysis based	Multi-resolution image extraction	(Adu-Gyamfi, Kambhamettu, & Okine, 2013)	Gaussian pyramids being utilized	Application based, not robust
	Edge detection-based segmentation	Detection operators	(Tsai, Kaul, & Mersereau, 2009)	Compared to six image segmentation techniques	High dependent on the quality of the image
	Gradient based	gradient histogram analysis	(Mahler, Kharoufa, Wong, & Shaw, 1991)	The pavement management aspect of the developed model is discussed	Only Pavement Serviceability Index (PSI) being considered.

Image segmentation	Threshold segmentation-based segmentation	OSTU	(Salari & Bao, 2011; Salari & Ouyang, 2012)	Complex background being considered	High computational complexity
	Region based segmentation	Genetic Algorithm	(Salari & Yu, 2011)	A kind of adaptive automated image analysis-based method	High computational complexity
	Transformation based	DWT	(Wu & Liu, 2012)	Robust and high-efficiency	Assumes that crack pixels are darker than the background
		Hausdorff	(Kaul, Tsai, & Yezzi, 2010)	The method is easy for application	Not suitable for blurred images
		Hough transformation	(Cheng, Wang, Hu, Glazier, Shi, & Chen, 2001)	High-efficiency compared to tradition manual method	Dependent on the selection of threshold values
		Fast Haar Transform (FHT),	(Abdel-Qader, Abudayyeh, & Kelly, 2003)	Efficient method	Dataset is not big enough
	Clustering based segmentation	Parzen density estimation	(Oliveira, Caeiro, & Correia, 2010)	The overlapping phenomenon of the images being considered	Cannot deal with more than two dimensions
	Matching based segmentation	Markovian	(Chambon, Gourraud, Moliard, & Nicolle, 2010; Chambon & Moliard, 2011)	Multi-scale Markovian segmentation method being proposed	More comparative experiments needed
	Hybrid method	dynamic programming-based	(Huang & Tsai, 2011)	Multi-scale characteristic being considered	The method is only validated in

		Genetic programming	(Nishikawa, Yoshida, Sugiyama, & Fujino, 2012)		certain regions
	Texture-based	regions pixels of coherent texture	(Varadharajan, Jose, Sharma, Wander, & Mertz, 2014)	Detailed information of the images is considered	The images tend to be over segmented
Feature extraction	Gray levels based	Spatial distribution analysis	(Forsyth & Ponce, 2003)	The algorithms are simple the easy to interpret	Can only deal with high quality images
		Histogram	(Nejad & Zakeri, 2011)		
	Transformation based	Wavelet transformation	(Zhou, Huang, & Chiang, 2005; Zhou, Huang and Chiang, 2006)	Multi-perspective information extracted	Computational complexity
		Discrete Fourier Transformation	(Ling, Peikang, Xiaohu, & Xudong, 2009)		
		Radon Transformation, Ridgelet Transformation	(Lin & Liu, 2010; Zhang, Qu, He, & Shi, 2009)		
		Curvelet Transformation	(Wu, Lu, Chen, & Wang, 2011)		
	Operator based	Canny, LOG, Sobel, Prewitt, and Roberts's	(Maini & Aggarwal, 2009; Sharifi, Fathy, & Mahmoudi, 2002)	Simple and easy to use	Cannot deal with complex images
		Modified operator	(Ayenu-Prah & Attoh-Okine, 2008)	bi-dimensional empirical mode decomposition being developed	Some false features being extracted
	Multi-scale information extraction based	Multiresolution Information Mining	(Attoh-Okine, Barner, Bentil, & Zhang, 2008)	Performs well for certain kind of pavement images	Performance not consistent

According to the literature review, it has been found that current automatic crack analysis algorithms are based on several assumptions, which can be summarized as follows:

(1) The differences between the cracked regions and the road backgrounds are significantly correlated with the gray-level distributions of the pavement image (Grandsaert, 2015).

(2) The cracked regions in an image are much darker than the road background (Amhaz, Chambon, Idier, & Baltazart, 2014).

(3) A crack is a continuous object in an image (Tang & Gu, 2013).

(4) Significant differences exist between the cracks and background in the transformed domain (Salman, Mathavan, Kamal, & Rahman, 2013).

2.4. Machine Learning Technologies in Engineering

Based on the realization mechanism difference of each method, MLTs can be classified into supervised learning techniques, unsupervised learning techniques, semi-supervised learning techniques, and the reinforced learning techniques (Alpaydin, 2009; Ivanović & Radovanović, 2014). Among those methods, supervised and unsupervised learning algorithms (Jahanshahi, Jazizadeh, Masri, & Becerik-Gerber, 2012) are the most widely used approaches in engineering. Supervised learning methods mainly refer to the approaches that use per-defined labels in the training datasets for the model optimization. Support Vector Machines (SVMs) (Barakat & Bradley, 2010; Diederich, 2007), Artificial Neural Networks (ANNs) (Andrews, Diederich, & Tickle, 1995; Tsukimoto, 2000), and Decision Trees (DT) are the representatives for the supervised learning. Unsupervised learning methods refer to the approaches that classify the datasets without labeling the datasets before the training process, whereas different rules are defined for the model optimization. Rough set theory (Greco, Matarazzo, & Słowiński, 2008; Sugimura, Obayashi, & Jeong, 2007; Sugimura, Obayashi, & Jeong, 2010), association rule mining, and clustering algorithms are the widely used unsupervised learning methods.

Table 2.5 shows a summary of literature that has focused on the MLTs based problem solving

in civil engineering. It can be seen that MLTs act as important tools for the development of intelligent infrastructure management systems. Among those different kinds of MLTs, it can be found that the most widely used algorithms are Support Vector Machines (SVM), Artificial Neural Networks (ANN), Genetic Algorithms (GA), Trees methods, Clustering algorithms, and the recently advanced Deep Convolutional Neural Networks (DCNN).

Table 2.5: Literature for the application of MLTs in Civil Engineering

Researchers	Research objective	Algorithm
(Wang, Zhuang, & Zhang, 2019)	Intelligent road crack detection	Support vector machines (SVM); Neural Networks (NN); Random Forests (RF); Boosted Tree (BT)
(Staniek & Czech, 2018)	Road monitoring, management	Neural Networks (NN)
(Ashrafian, Taheri Amiri, Rezaie-Balf, Ozbakkaloglu, & Lotfi-Omran, 2018)	Prediction of compressive strength	Heuristic regression methods
(Yang, Li, Zhan, Fei, & Zhang, 2018)	Pavement Texture analysis: Friction Model	Convolutional Neural Network (CNN)
(Cheng & Wang, 2018)	Sewer pipe defects detection	Faster Region-based Convolutional Neural Network (Faster R-CNN)
(Agathos, Chatzi, & Bordas, 2018)	Crack detection	Genetic Algorithms (GA)
(Turkan, Hong, Laflamme, & Puri, 2018)	Crack detection	Adaptive wavelet neural network

(Cha, Choi, Suh, Mahmoudkhani, & Büyüköztürk, 2018)	Autonomous Structural Visual Inspection	Region-Based Deep Learning
(Dorafshan, Thomas, Coopmans, & Maguire, 2018)	Structural Inspections	Deep Learning
(Xue & Li, 2018)	Shield Tunnel Lining Defects	Region- Convolutional Neural Networks (RCNN)
(Wang, Gopalakrishnan, Smadi, & Somani, 2018)	Pavement crack detection	Polynomial curve fitting
(Nabian & Meidani, 2018)	Seismic Reliability Analysis	Deep Learning (DL)
(Rafiei & Adeli, 2018)	Structure condition assessment	Convolutional Neural Networks (CNN)
(Yeum, Dyke, & Ramirez, 2018)	Disaster evaluation	Deep Convolutional Neural Networks (DCNN)
(Hadjidemetriou, Vela, & Christodoulou, 2018)	Pavement Patch Detection	Support Vector Machine (SVM)
(Hoang, 2018)	Asphalt Pavement Pothole Detection	Support Vector Machine (SVM); Neural Network (NN)
(Molina-Cabello, Luque-Baena, López-Rubio, & Thurnhofer-Hemsi, 2018)	Vehicle type detection	Convolutional Neural Networks (CNN)

(Gopalakrishnan, Khaitan, Choudhary, & Agrawal, 2017)	Pavement distress detection	Deep Convolutional Neural Networks (DCNN)
(Maeda, Sekimoto, Seto, Kashiyama, & Omata, 2017)	Road maintenance decision making	Decision Trees (DT)
(Ouma & Hahn, 2017)	Pothole detection	Fuzzy c-means clustering
(Wang, Qiu, Wang, Xiao, & Wang, 2017)	Cracking Classification	Support Vector Machine (SVM)
(Maeda, Takahashi, Ogawa, & Haseyama, 2017)	Distress classification of road structures	Adaptive Bayesian Network
(Zhang, Jiang, Long, & Han, 2017)	Fault diagnosis	Decision Tree (DT); Clustering algorithm
(Liu, Zhao, Ju, Shi, Shi, & Shi, 2017)	Materials design	Support Vector Machine (SVM); Artificial Neural Network (ANN)
(Todkar, Le Bastard, Ihamouten, Baltazart, Dérobert, Fauchard, Guilbert, & Bosc, 2017)	Detection of deboning	Support Vector Machine (SVM)
(Hasni, Alavi, Jiao, & Lajnef, 2017)	Detection of fatigue cracking	Support Vector Machine
(Tedeschi & Benedetto, 2017)	Automatic pavement crack and pothole recognition	Neural Networks (NN)
(Radopoulou & Brilakis, 2017)	Automated detection; Pavement assessment;	Semantic Textron Forests (STFs)

(Jang, Yang, Smyth, Cavalcanti, & Kumar, 2017)	Automated pavement detection and assessment;	Clustering
(Wang, Zhang, Li, Fei, Chen, & Li, 2017)	Asphalt pavement cracking recognition	Deep Convolutional Neural Networks (DCNN)
(Shi, Cui, Qi, Meng, & Chen, 2016)	Automatic road crack detection	Random Forests (RF)
(Lempert, Sidorov, Zhukov, & Nguyen, 2016)	Automated detection; Pavement assessment;	Neural Networks (NN)
(Rafiei, Khushefati, Demirboga, & Adeli, 2016)	Concrete material characterization	Neural Networks (NN)
(Ziari, Sobhani, Ayoubinejad, & Hartmann, 2016)	Prediction of IRI	Artificial Neural Network (ANN)
(Schlotjes, Henning, Burrow, & Evdorides, 2015)	Predict the probability of pavement failure	Support Vector Machines (SVM)
(Yan, Goto, Miyamoto, & Zhao, 2014)	Rating for corrosion states of weathering steel	Wavelet transformation; PSO-SVM
(Cheng, Firdausi, & Prayogo, 2014)	Concrete compressive strength prediction	Genetic Weighted Pyramid Operation Tree (GWPOT)
(Tabatabaee, Ziyadi, & Shafahi, 2013)	Pavement performance modeling	Support Vector Machine (SVM), Recurrent Neural Network (RNN)

(Terzi, Karaşahin, Gökova, Tahta, Morova, & Uzun, 2013)	Asphalt concrete stability estimation	Artificial Neural Networks (ANN)
(O'Byrne, Schoefs, Ghosh, & Pakrashi, 2013)	Pavement damage detection	Artificial Neural Networks (ANN)
(Hsie, Ho, Lin, & Yeh, 2012)	Asphalt pavement cracking modeling	Genetic algorithm; Tree models
(Moussa & Hussain, 2011)	Flexible pavement crack detection;	Support Vector Machines (SVM)
(Fan, Zhang, Ding, Xie, & Xu, 2010)	Pavement crack detection	Back Propagation Neural Network (BPNN)
(Li, Hou, Yang, & Dong, 2009)	Automation pavement surface distress recognition	Support Vector Machine (SVM)
(Tapkin, Çevik, & Uşar, 2009)	Strain prediction of polypropylene modified Marshall specimens	Artificial Neural Networks (ANN)
(Farran & Zayed, 2012)	Life-cycle costing for sustainable infrastructure rehabilitation	Markov Decision Process, Genetic Algorithms (GA)
(Ng, Tran, & Osman, 2006)	Modelling serviceability deterioration of the concrete stormwater pipe	Genetic Algorithm (GA); Neural Networks (NN)
(Huang, Hsu, Lee, & Chang, 2005)	Airport maintenance strategies selection	Artificial Neural Network (ANN)

Machine Learning Techniques are developed based on the advancement of big data, which provides the foundation for discovering intrinsic knowledge through intelligent learning algorithms. Generally, an effective machine learning model is developed based on four aspects: (i) enough training data (more than 100 samples); (ii) an experienced machine learning trainer; (iii) a meaningful target function; and (iv) an effective training algorithm. Meanwhile, most of the current developed Deep Learning (DL) techniques are based on deep neural networks, which belongs to the supervised learning. However, DL techniques have become the most advanced technique compared to traditional shallow learning techniques. Therefore, more explanations are provided regarding deep learning techniques.

2.4.1 Supervised Learning Techniques

Supervised learning techniques are models trained based on given desired output values. In the training database, both the input instances and the corresponding ground truth outputs are included for the model to learn intrinsic knowledge. The objective functions are usually built based on the concept of calculating the differences between the model outputs and the corresponding ground truth values. On this basis, the model training process is implemented with the purpose of minimizing the constructed objective function. The most significant characteristics of these methods are the existence of the specific 'teacher' and the paired input-output datasets. In terms of problem-solving categories, supervised learning techniques can deal with two types of problems, which are regression and classification problems. If the ground truth outputs are continuous targets, the task can be considered as a regression problem. Otherwise, if dealing with discrete variables, the task should be regarded as a classification problem. Some machine learning algorithms can deal with both regression and classification problems, such as neural networks, ensemble algorithms, etc. A rough illustration of the most widely used algorithms is shown in Figure 2.13. It should be noted that there is not a hard division border between these two kinds of tasks, which means that they are interchangeable under some circumstances.

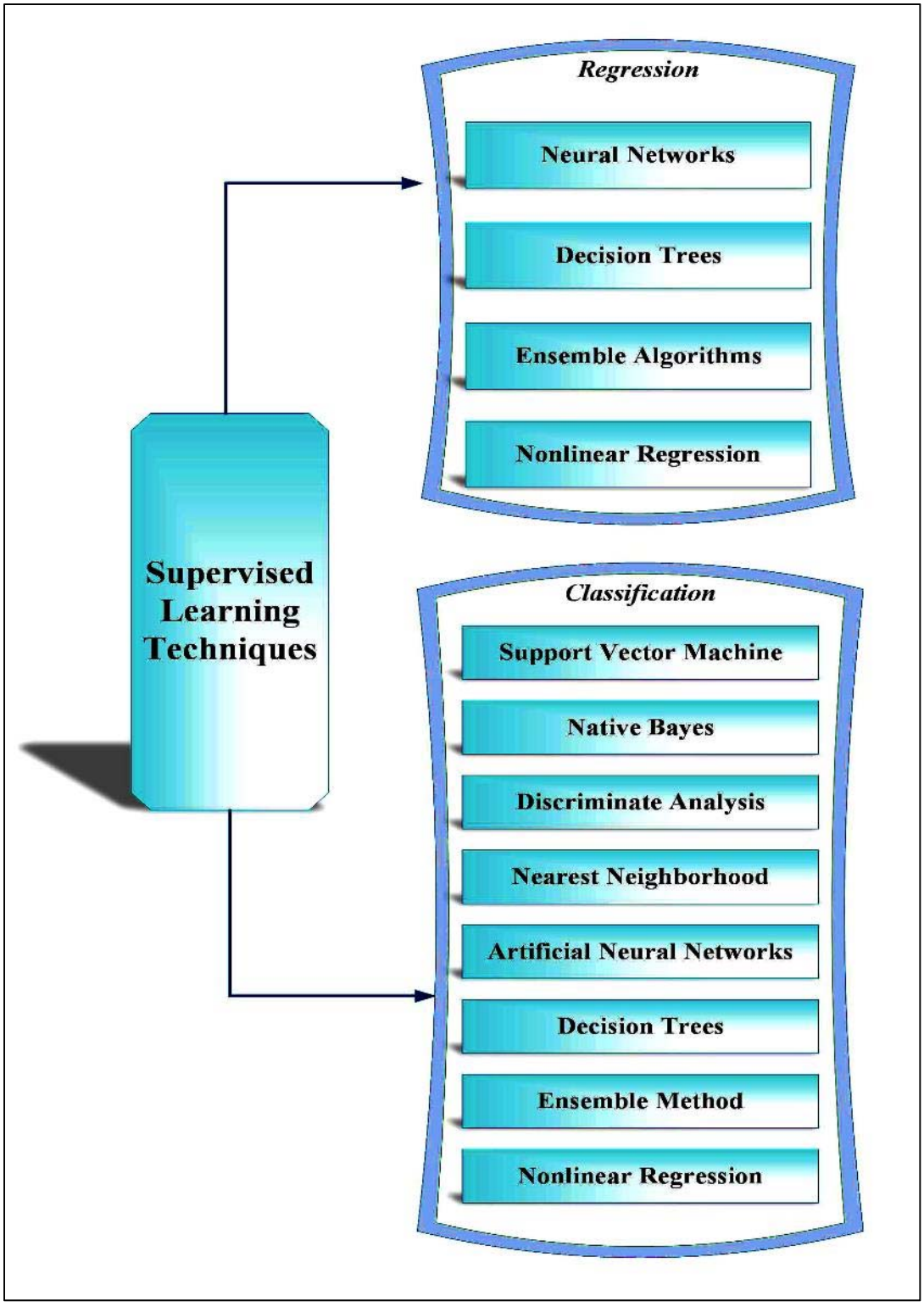


Figure 2.13: Supervised learning

2.4.2 Unsupervised Learning Techniques

Different from the supervised learning methods, the most significant characteristic of unsupervised learning is that no ground truth values are given for the model training. That means the algorithm should learn the intrinsic knowledge only based on input instances. For example, clustering algorithms, which are the most advanced representatives for the unsupervised learning techniques, aim to divide given datasets into several clusters with each cluster represents a certain category. The objective function is usually formulated by calculating the distances among different clusters. The most widely used unsupervised learning algorithms are summarized in Figure 2.14.

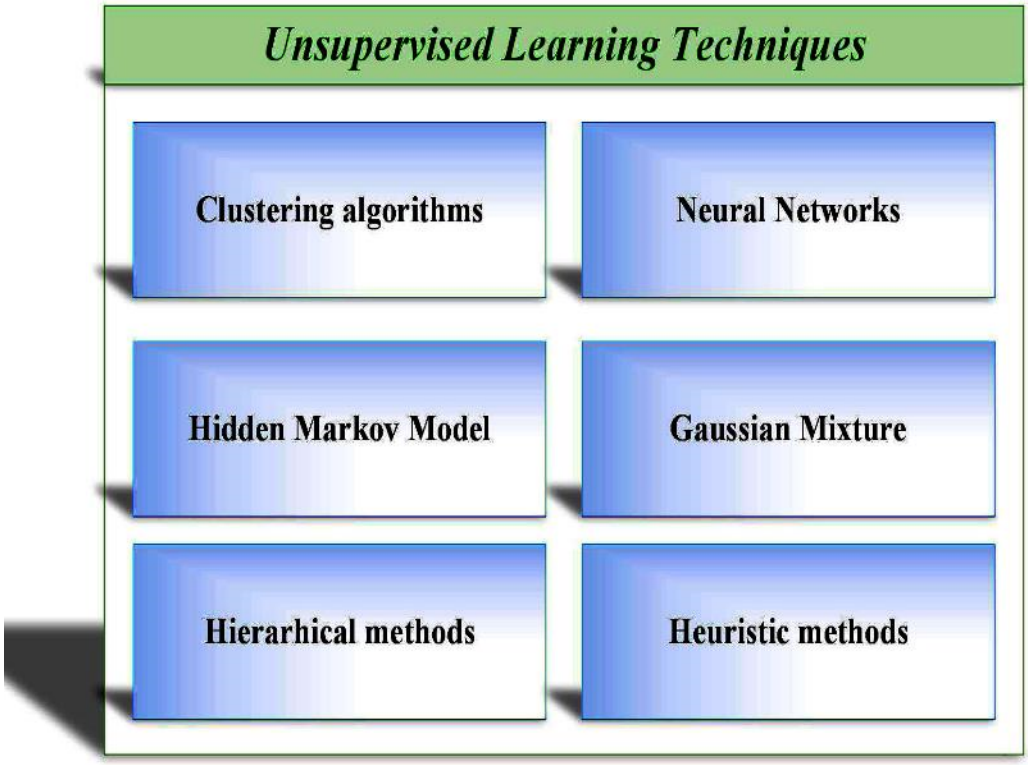


Figure 2.14: Unsupervised learning

2.5. Deep Learning Technology

Traditional Machine Learning based road distress images extraction studies are mainly based on the unsupervised learning approaches, such as k-means clustering and the threshold-based methods. Most of those methods (SVM, DT, Clustering, etc.) are shallow learning methods, sensitive to the intensity condition of the images. Moreover, the size of the non-structured image data can be thousand times of the traditional structured data. Traditional shallow machine learning methods do not perform well in dealing with large quantities of the non-structured image data. That's why Deep Learning (or Deep Neural Network) has been developed for dealing with the problems which traditional ML techniques cannot solve. In recent years, Deep Learning (DL) techniques has been successfully utilized in several fields such as object recognition, speech recognition (Hinton, Deng, Yu, Dahl, Mohamed, Jaitly, & Kingsbury, 2012), data denoising (Vincent, Larochelle, Bengio, & Manzagol, 2008) and feature extraction.

With the rapid development of Deep Learning technology, pavement engineering researchers have conducted researches using Deep Learning Techniques for road network feature extraction based on images from very-high-resolution (VHR) aerial and satellite imaging systems. Similar work can be found from the work of Wang et al (Wang, Li, Yang, Zhan, & Qiu, 2015), who proposed a framework using CNN and Finite State Machine (FSM) to detect roads from pavement images. Wang's work has then been extended by other scholars to conduct research of automated road point initialization analysis.

Compared with traditional shallow learning techniques, such as support vector machines and regression analysis, deep learning techniques are more intelligent as they are based on the exploration of higher-level features of large quantity input images. Traditionally, simple issues can be solved by constructing linear classifiers, which are only based on the analysis of limited features. However, most images collected from the physical world contain complex information and the quality varies significantly between each other. In other words, pavement surface images, for instance, always contain complex background information and are usually influenced by the roadway environment. Therefore, most of those problems are nonlinear and

cannot be modeled by traditional concepts. Deep Learning techniques are straightforward ways aim at nonlinear system modeling. This objective is fulfilled by adding more network layers with a large number of neural nodes. The weights assigned to each neural node act as unit tools for nonlinearity modeling and multi-scale information extraction.

The resurgence of deep neural network techniques is due to the availability of big data and the advancement of computer technologies. The big data provide resources for the model to learning more comprehensive information of a certain object. The highly advanced computer technology made it possible for parallel data processing, which can improve the efficiency of model training.

The most widely used deep learning architectures are deep convolutional neural networks (DCNN). Those kinds of architectures represent specific systems that only the neurons in the field of view are connected with each. By this way, the computational complexity is significantly reduced compared to traditional complete fully connected networks. Meanwhile, since the input layer contains overlaps among different neurons, the feature regions of the image can be completely covered without missing any point.

The data labeling operations should be conducted in advance, which is usually performed by professional operators. Then, the labelled training dataset can be uploaded into the architecture. By defining the loss function and the selection of appropriate training method, the model training process can be performed successively. The model validations are conducted along with the training process, then the final model can be obtained when the training operation is completed. The model can be used to handle the classification or regression problems of the new datasets.

2.5.1 Artificial Neural Networks

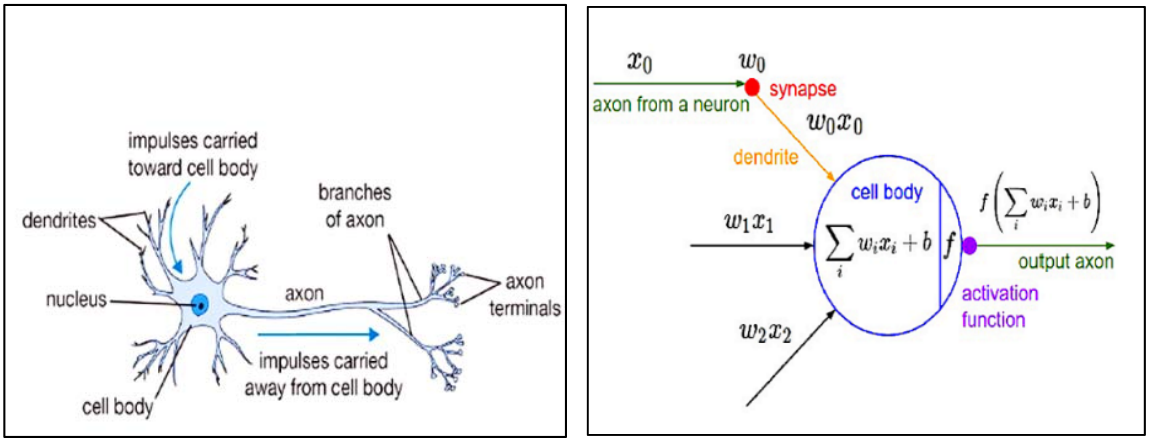
Besides Decision Trees, Support Vector Machines, Radial Basis Functions (RBF), and Bayesian networks, Artificial Neural Networks (ANN) should be another most widely used MLT. The initial idea of ANN is inspired by the biological neural of human beings. However, they are not actually identical. Figure 2.15 is an illustration of the biological neuron in the left

side (Figure 2.15a) and its mathematical model in the right side (Figure 2.15b). ANN systems consist of a group of unit processors with several interconnections, thereby forming the interconnected architecture which is similar to parallel computing systems. There are two aspects showing how ANNs resemble human brains (Haykin, 1999): the networks are able to learn the intrinsic knowledge through the learning process; and the synaptic weights which interconnect the strengths between neurons are used to store knowledge.

A complete ANN consists of several layers, which are organized by a batch of neurons connected with each other by the weights, biases, and activation functions. Equation (2.1) represents the operations conducted on each node (neuron).

$$y_i = f_i(\sum_{j=1}^{n} w_{ij} x_j + b_j) \tag{2.1}$$

where f_i represents the activation function of the i^{th} node; y_i represents the output of the i^{th} node in the current layer, which should be used as the input value of the next layer; x_j represents the j^{th} input to the i^{th} neuron; W_{ij} represents the weight between the i^{th} and j^{th} neurons. b_j represents the bias of the i^{th} neuron.



(a) A cartoon drawing of the biological neuron **(b) Mathematical model**

Figure 2.15: ANN demonstration

Feed-forward propagation is the basic neuron connection method used in ANNs. In this method, the nodes of each layer are only connected to the next layer that is closer to the output layer. Another layer connection method of ANNs is the cascade-forward approach, in which the nodes of a specific layer can be connected to all subsequent layers. The topology of an ANN represents the number of layers, nodes, and the connection methods between each layer. Activation function acts as the non-linear transformation operator which can simulate complex problems. Feed-forward multi-layer perception (MLP), a typical kind of ANN architecture, has been used in diverse fields to solve problems such as pattern recognition, regression, and function approximations (Hornik, Stinchcombe, & White, 1989).

Learning, or training process, aims to adjust the weights and biases of the connections to minimize the error between the network outputs and the expected values. With this purpose, the Mean Squared Error (MSE) is often used as an indicator for the network optimization. Back-propagation gradient descent is the most popular method in training the ANNs, during which the weights and biases are updated simultaneously.

Three important aspects (Jain, Mao, & Mohiuddin, 1996; Samarasinghe, 2016) should be taken into consideration in order to ensure accuracy of the trained ANNs. They are: construction of the appropriate ANN architecture (number of layers, nodes, etc.) with proper selected activation function (Table 2.6); definition of suitable learning rules; and efficient model training implementations (weights and biases).

2.5.2 Deep Convolutional Neural Network (DCNN)

Conventional machine learning techniques have been widely used in diverse fields for data mining and knowledge discovery practices. However, professional engineering expertise considerations are required when designing the feature extractors. Those extractors are used to transfer the input datasets into multiple scales of feature maps. After that, the classifiers should be designed to achieve the final objective based on the extracted multi-scale features. The traditional manual selection of feature extraction and classification algorithms are often empirical and subjective. Therefore, the trained models are often prone to unexpected noise.

Multi-layer Neural Networks (NN) should be a possible approach in solving this problem, because these kinds of architectures allow the network to find the best feature extractors by learning the weights from the input datasets. Characterized by the capability of self-leaning, NN has been regarded as a powerful machine learning technique which can be trained to approximate complex non-linear functions for diverse kinds of high-dimensional inputs.

Table 2.6: Activation functions and their characteristics

Name	**Plot**	**Equation**	**Derivative**	**Monotonic**	**Derivative Monotonic**	$f(x) \approx 0$ **when** $x \approx 0$
Identity		$f(x) = x$	$f(x)' = 1$	Yes	Yes	Yes
Binary step		$f(x) = \begin{cases} 0 \ for \ x < 0 \\ 1 \ for \ x \geq 0 \end{cases}$	$f(x)' = \begin{cases} ? \ for \ x = 0 \\ 0 \ for \ x \neq 0 \end{cases}$	Yes	No	No
Logistic (Sigmoid)		$f(x) = \frac{1}{1 + e^{-x}}$	$f(x)' = \mathrm{f(x)(1 - f(x))}$	Yes	No	No
Tanh		$f(x) = tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f(x)' = 1 - \mathrm{f(x)}^2$	Yes	No	Yes
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 \ for \ x < 0 \\ x \ for \ x \geq 0 \end{cases}$	$f(x)' = \begin{cases} 0 \ for \ x < 0 \\ 1 \ for \ x \geq 0 \end{cases}$	Yes	Yes	No

NN has been recognized as a high reliable class of functions in dealing with knowledge discovery tasks based on big data. However, when used in dealing with image datasets, increasing layers are required for complex feature extraction. But for NN, increasing input dimensions leads to an increase of hidden layers at the same time, because the layers are fully connected with each other. These kinds of characters made it seems incredible for NN being used in image detection because the size of the image datasets tends to increase by thousands of times However, Convolutional Neural Network (CNN) is proposed as an approach which tries to alleviate the mentioned problems. The basic concepts of CNNs are utilizing the principle of weight sharing approach which can drastically reduce the number of model parameters.

CNN has been widely applied to various fields from the early 1900s. Representatives efforts include the development of speech recognition time delay neural networks (Waibel, Hanazawa, Hinton, Shikano, & Lang, 1990) and document reading networks (LeCun, Bottou, Bengio, & Haffner, 1998). However, not until the early 2000s CNNs were successfully used in image segmentation, object detection and recognition, especially in the applications of medical image segmentation (Ning, Delhomme, LeCun, Piano, Bottou, & Barbano, 2005). Other successfully applicable areas include but are not limited to human face detection and recognition, and pedestrian detection in transportation engineering (Sermanet, Kavukcuoglu, Chintala, & LeCun, 2013; Vaillant, Monrocq, & Le Cun, 1994; Nowlan & Platt, 1995; Garcia & Delakis, 2004; Nasse, Thurau, & Fink, 2009; Tompson, Goroshin, Jain, LeCun, & Bregler, 2015). Despite those successful achievements, CNN was forsaken by the mainstream of computer-vision and machine-learning communities until 2012. In the Image Large Scale Visual Recognition Challenge (ILSVRC) competition (Deng, 2009), the winner Alex Krizhevsky's team proposed a deep convolutional neural network based on the datasets of ImageNet, which is called AlexNet (Krizhevsky, Sutskever, & Hinton, 2012). The most notable characteristics of AlexNet are reflected in four aspects: the efficient use of GPUs; the innovative use of non-linear function called Rectified Linear Unit (ReLU) as activation function; the proposal of novel regularization technique called dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014); and the state-of-the-art data augmentation techniques. After that breakthrough, ZFNet (Zeiler & Fergus, 2014) (Figure 2.17), which is the winner of ILSVRC

in 2013, has made improvements based on original AlexNet. In ZFNet, the most significant feature is the visibility which is achiveved by adjusting the architecture of the original AlexNet (Figure 2.16). In 2014 ILSVRC, Google joined the competition and won first place by the proposed Google Net. Google Net increased the layers of the network to 22 layers in total and adopted a multi-scale data training approach for more accurate object detection. Another change of Google Net that differs from conventional networks is that it dropped fully connected layers in the network architecture, which decreased 90% of the parameters of the whole model. In 2014, ILSVRC winner was the Visual Geometry Group (VGG), who developed the VGGNet (Figure 2.18) (Simonyan & Zisserman, 2014). This architecture used 3*3 convolution kernels combining with deeper layers, hence VGG has higher potential of being used in different fields. Nowadays, CNNs have become dominant methods for dealing with all kinds of image and video-based object detection and recognition tasks (Taigman, Yang, Ranzato, & Wolf, 2014).

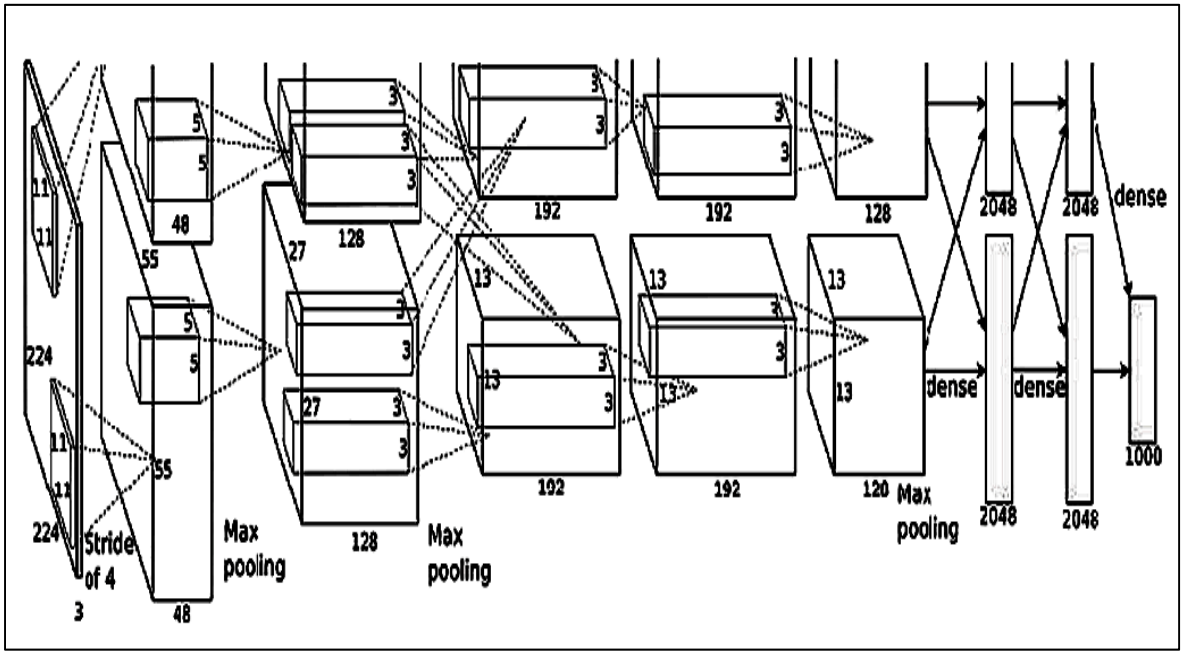
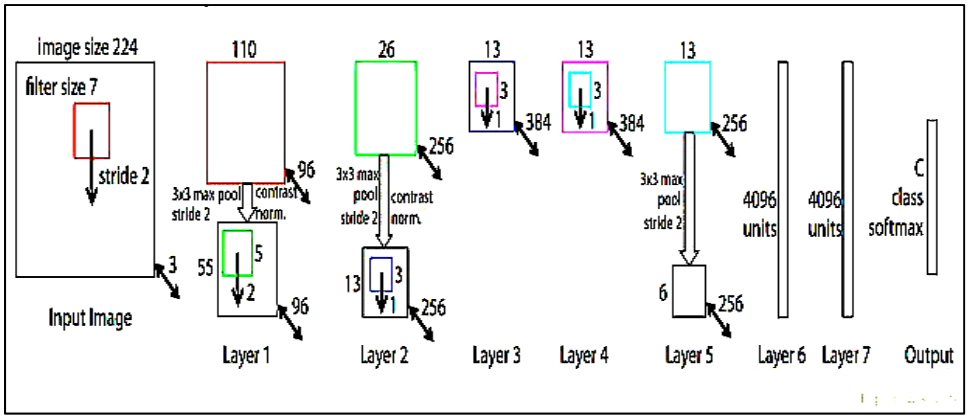


Figure 2.16: AlexNet (Krizhevsky, Sutskever, & Hinton, 2012)



(a) Architecture

Layer	Input	$Size_{kernel}$	Stride	Output	$Num_{parameters}$	Memory
CONV1	3@227 × 227	7 × 7	2	96@111 × 111	96 × (7 × 7 × 3 + 1)	96 × 111 × 111
POOL1	96@111 × 111	3 × 3	2	96@55 × 55	0	96 × 55 × 55
CONV2	96@55 × 55	5 × 5	2	256@26 × 26	256 × (5 × 5 × 96 + 1)	256 × 26 × 26
POOL2	256@26 × 26	3 × 3	2	256@13 × 13	0	256 × 13 × 13
CONV3	256@13 × 13	3 × 3	1	512@13 × 13	512 × (3 × 3 × 256 + 1)	512 × 13 × 13
CONV4	512@13 × 13	3 × 3	1	1024@13 × 13	1024 × (3 × 3 × 512 + 1)	1024 × 13 × 13
CONV5	1024@13 × 13	3 × 3	1	512@13 × 13	512 × (3 × 3 × 1024 + 1)	512 × 13 × 13
FC6	86528@1 × 1	1 × 1	1	4096@1 × 1	4096 × (86528 + 1)	4096 × 1 × 1
FC7	4096@1 × 1	1 × 1	1	4096@1 × 1	4096 × (4096 + 1)	4096 × 1 × 1
FC8	4096@1 × 1	1 × 1	1	1000@1 × 1	1000 × (4096 + 1)	1000 × 1 × 1
total					386548840	2044840

(b) Incarnation of Architecture Inception

Figure 2.17: ZFNet (Zeiler & Fergus, 2014)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

(a) Architecture

Layer	Input	$Size_{kernel}$	Stride	Output	$Num_{parameters}$	Memory
CONV3-64	3@224 × 224	3 × 3	1	64@224 × 224	64 × (3 × 3 × 3 + 1)	64 × 224 × 224
CONV3-64	64@224 × 224	3 × 3	1	64@224 × 224	64 × (3 × 3 × 64 + 1)	64 × 224 × 224
MAXPOOL	64@224 × 224	2 × 2	2	64@112 × 112	0	64 × 112 × 112
CONV3-128	64@112 × 112	3 × 3	1	128@112 × 112	128 × (3 × 3 × 64 + 1)	128 × 112 × 112
CONV3-128	128@112 × 112	3 × 3	1	128@112 × 112	128 × (3 × 3 × 128 + 1)	128 × 112 × 112
MAXPOOL	128@112 × 112	2 × 2	2	128@56 × 56	0	128 × 56 × 56
CONV3-256	128@56 × 56	3 × 3	1	256@56 × 56	256 × (3 × 3 × 128 + 1)	256 × 56 × 56
CONV3-256	256@56 × 56	3 × 3	1	256@56 × 56	256 × (3 × 3 × 256 + 1)	256 × 56 × 56
CONV3-256	256@56 × 56	3 × 3	1	256@56 × 56	256 × (3 × 3 × 256 + 1)	256 × 56 × 56
MAXPOOL	256@56 × 56	2 × 2	2	256@28 × 28	0	256 × 28 × 28
CONV3-512	256@28 × 28	3 × 3	1	512@28 × 28	512 × (3 × 3 × 256 + 1)	512 × 28 × 28
CONV3-512	512@28 × 28	3 × 3	1	512@28 × 28	512 × (3 × 3 × 512 + 1)	512 × 28 × 28
CONV3-512	512@28 × 28	3 × 3	1	512@28 × 28	512 × (3 × 3 × 512 + 1)	512 × 28 × 28
MAXPOOL	512@28 × 28	2 × 2	2	512@14 × 14	0	512 × 14 × 14

(b) Incarnation of Architecture Inception

Figure 2.18: VGGNet (Simonyan, K., and Zisserman, A.,2014)

CNN is proven to be robust and has the capability of overcoming tilt, noise, and displacement of objects in the images (Xu, Luo, Wang, Gilmore, & Madabhushi, 2016; Sun, Wang, & Tang, 2013). Those capabilities are feasible for detecting objects from both 2D and 3D images (Hu, Chang, Nian, Wang, & Li, 2016). Based on those characteristics, CNN has been extensively used in several digital images-based feature extraction and object detection fields, which have yield satisfactory results. A basic CNN process includes architecture establishment, network training, and testing. Based on the specific requirement, original constructed CNN should be tested, re-structured and re-trained for new datasets, if necessary.

For the purpose of object identification, Region Proposal Convolutional Neural Networks (R-CNN) (Figure 2.19) has been proposed (Girshick, Donahue, Darrell, & Malik, 2014) by RBG team in 2013. R-CNN can predict the bounding box of an object together with accurate probability in an image. Selective Search (Uijlings, Van De Sande, Gevers, & Smeulders, 2013) is used to extract region proposals that act as the proposed object in an image. After that, CNN is trained for feature extraction using region proposals as inputs, and then feeding the results into SVM for object classification. To summarize, the primary purpose of R-CNN is object detection, which is realized by the searching strategy-based region proposal extraction and the CNN based object classification. Region proposals are acquired by selective search and convolutional neural network layers are constructed using several convolutional layers, max-pooling layers, and the fully connected layers. The input of R-CNN is the image to be detected, and the outputs are images including the bounding boxes and labels for each object. In the pre-trained R-CNN network, a selective search is adopted firstly to obtain a total of 2k region proposals from original images. After that, the extracted region proposals are warped to the specified size, say, 227×227(AlexNet input required). Thereafter, each region proposal is put into CNN for feature extraction to obtain the desired feature maps, which are presented by feature vectors. Finally, those feature vectors are used for object classification by SVMs. This research can be considered a great breakthrough using CNN for image detection ever since convolutional neural network architecture was proposed by Alex Krizhevsky. The achievement of R-CNN has attracted significant attention in the field for its satisfactory performance regarding the image classification. However, some disadvantages are observed as well, which include: 1) R-CNN requires a forward CNN (AlexNet) for every individual region proposal of

certain image; 2) the methods for image feature extraction, classification and regression are trained with different models, which makes the model training process complex and time consuming.

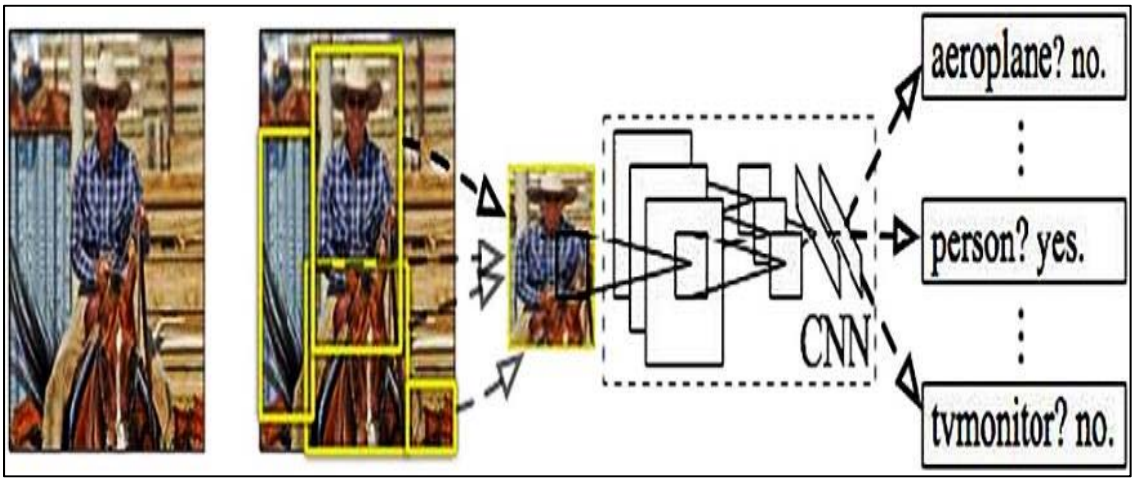


Figure 2.19: R-CNN (Girshick, Donahue, Darrell, & Malik, 2014)

The improved version of R-CNN is called Fast-RCNN (Figure 2.20), which was proposed in 2015 (Grishick, 2015). Faster-RCNN adopted multi-task regression model instead of bounding box regression (bbox regression) and obtained the region proposals in the fifth layer. The significant improvements of Fast R-CNN are: 1) feature maps are acquired from the whole images using selective search method; 2) ROI pooling layer is added to the last convolution layer; 3) multi-task loss functions are introduced to calculate the total loss of the network; 4) SoftMax function and bounding box (bbox) regression are used for classification and regression, respectively. Meanwhile, the ROI pooling, SoftMax classification, and bbox regression are implemented in an end-to-end CNN.

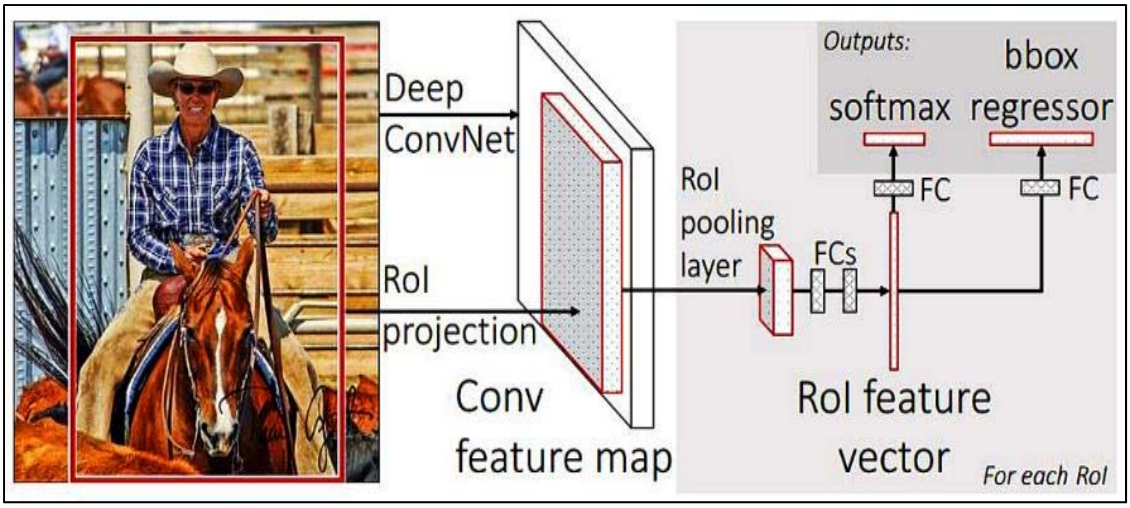


Figure 2.20: Fast-RCNN framework (Girshick, 2015)

The main bottleneck of Fast R-CNN models that the region proposals generating method still relays on the Selective Search method. This problem significantly slowed down the whole network. Therefore, the improved version, which is called Faster-RCNN (Figure 2.21), is developed. The most significant feature of Faster-RCNN is the development of Region Proposal Network (RPN), which conducts the candidate regions selection by the network itself. Specifically, Faster R-CNN (Ren, He, Girshick, & Sun, 2015) only includes two components, which are Region Proposal Network (RPN) and Fast R-CNN. RPN is the remarkable innovation of Faster R-CNN, which features the main difference between Faster R-CNN and Fast R-CNN. RPN can deal with input images of arbitrary-size and then generate the reliable region proposals. The utilization of anchors (Figure 2.21) can be considered as one of the most important characteristics of RPN. By using anchors, multi-scale region proposals can be extracted, which means RPN can obtain the multi-scale knowledge incorporation from the input images. Those changes have significantly improved the speed of the network training process. However, improvement room still exists. Since in this architecture, the step for proposal extraction is not integrated into the whole network.

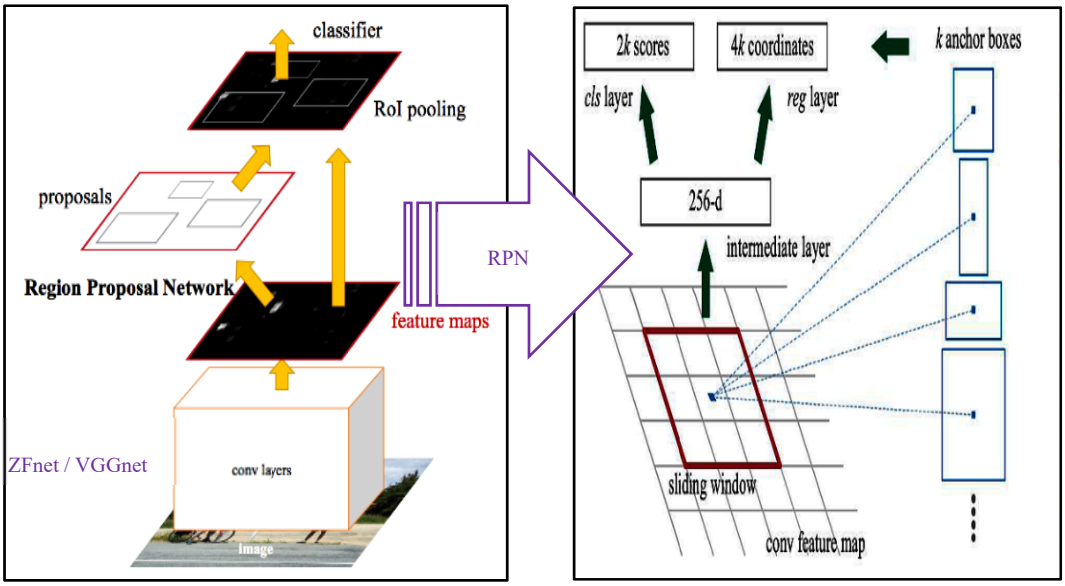


Figure 2.21: Faster-RCNN (modified from Ren, He, Girshick, & Sun, 2015)

In facing the rapid development of all those technologies, traditional digital image processing based automated pavement cracking detection algorithms call for improvement to meet the requirements of both industries and research institutes. Meanwhile, developing innovative methodologies for automated pavement cracking detection can highly improve the accuracy and efficiency of existing methods, thereby facilitating automated pavement management systems. Therefore, this research proposes to train network for pavement cracking detection and classification based on Faster-RCNN and developing innovative CNN framework for automated pavement condition evaluation.

2.5.3 Basic Procedure of CNN

Traditionally, different layers of ANN are fully connected regardless of the number of the layers being chosen. These kinds of operations might perform well on small images which represent small datasets. However, for large images (e.g., 128×128 images) or large datasets, parameters in different layers being fully connected should be extremely computationally expensive. Moreover, the parameters to be trained tend to increase incredibly. That's why ANN has once been considered by researchers as impossible of going deeper, and therefore, cannot

be used in dealing with large quantities of data, especially for large images. Then, Convolutional Neural Network (CNN) architectures were proposed to solve the problem which ANNs faces in the object detection. CNN adopted the concept of locally connected area convolution, together with other innovations such as weight sharing and pooling operation. Those innovations made CNNs attract researchers' attention again since they provided high potential for intelligent object detection. Compared with traditional image detection methods, CNN has its own advantages that other methods do not have, which include (1) error tolerance ability and self-learning ability; (2) high generalization ability that can be easily applied in areas such as pattern classification, object detection, and recognition; and (3) multi-layer perceptron, which can learn the features of the detected images comprehensively. Those features made the models become more robust to different kinds of images and do not involve complex pre-processing operations. The basic structure of a CNN mainly includes the following layers: convolutional layer, pooling layer, and fully connected layer. Each layer has several feature maps, which aim at extracting certain types of features by the convolution filters. CNNs are usually built following the principle of: the first layer after input is a convolutional layer, in which every neuron is locally connected with the input images and extracts their local features; after the convolutional layers is the pooling layer, which is used to reduce the parameters of the model without affecting the accuracy. The convolutional layer and pooling payer are usually repeated for comprehensive feature extraction. Finally, the fully connected layer is used for object classification and regression.

Convolution process is illustrated in Figure 2.22. It can be seen that a convolution kernel should be selected to slide across the image before convolution operation. The size of the kernel can be selected according to the specific problems, which is usually defined by a matrix such as $3 \times 3,\ 5 \times 5,\ 8 \times 8$, etc. After the convolution kernel has been decided, sliding the kernel template across the images is done, and followed by multiplying the image data with template values. Then the specific feature relating to the certain convolution kernel can be extracted from the image, obtaining the extracted feature maps.

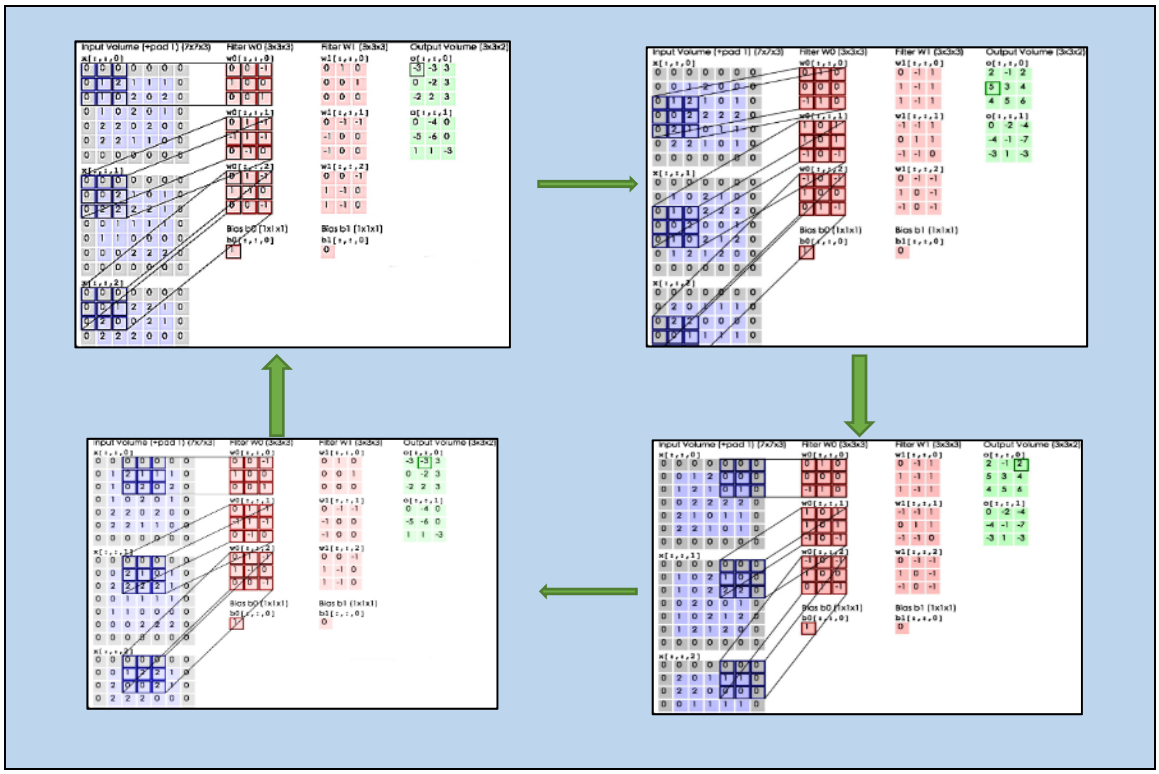
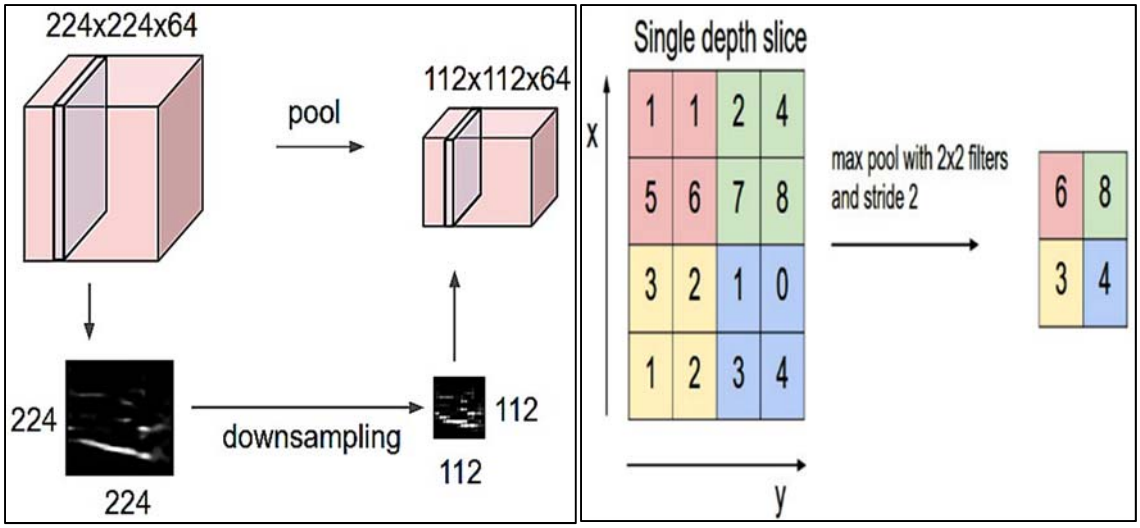


Figure 2.22: Illustration of the convolution process

Convolution layers are always followed by pooling layers, which can be used to reduce the parameters for learning and avoid overfitting on the training data. For example, if dealing with an image with the size of 48×48 pixels, 500 feature maps can be obtained with the template size of 5×5 after the convolution layer. Thus, the output size after convolution is $(48 - 5 + 1) \times (48 - 5 + 1) = 1936$. Therefore, if there are 500 feature maps, the total parameters of the vector can be up to $44 \times 44 \times 500 = 968{,}000$. Training the classifier for such a small image can drive the computer mad, let alone for the larger images, which tend to make the machine unwieldly and the model prone to overfitting. Researchers noticed the 'stationarity' feature of the image. Based on that they proposed the method of 'pooling', which computes the mean (or max) value of a feature over a certain region of the image as the result of the output value of that region. Thus, the parameters are reduced significantly. The most commonly used pooling operations are the Max-pooling (Figure 2.23) and mean pooling. Figure 2.23 shows the mechanism of the Max-pooling operation. Figure 2.23a shows the Max-pooling over the feature image. Figure 2.23b shows Max-pooling over 4-non-overlapping feature images.



(a) Max-pooling over feature image (b) Max-pooling over 4 non-overlapping images

Figure 2.23: Max-pooling illustration

The last several layers before the output layer are always 1-3 fully connected layers, which are fully connected to all the input features and then output to the classifiers. The classification and regression are usually achieved by SoftMax regression method. The probability of the input vector $\boldsymbol{x}$ can be calculated by Equation (2.2):

$$h_0(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)};\theta) \\ p(y^{(i)} = 2|x^{(i)};\theta) \\ p(y^{(i)} = 3|x^{(i)};\theta) \\ \dots \\ p(y^{(i)} = k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ e^{\theta_3^T x^{(i)}} \\ \dots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{2.2}$$

where $\boldsymbol{p(y = j|x)}$ represents the probability of class $\boldsymbol{j}$ for input vector of $\boldsymbol{x}$; $\theta_1, \theta_2, \theta_3, \dots\dots, \theta_k \epsilon R^{n+1}$ are model parameters; $\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}$ can be considered as normalization items, which control the sum of all the values in the vector equal to one. The loss function of SoftMax regression can be expressed by Equation (2.3):

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] \tag{2.3}$$

where $\theta = \begin{bmatrix} \theta_1^T \\ \theta_2^T \\ \theta_3^T \\ \cdots \\ \theta_k^T \end{bmatrix}$.

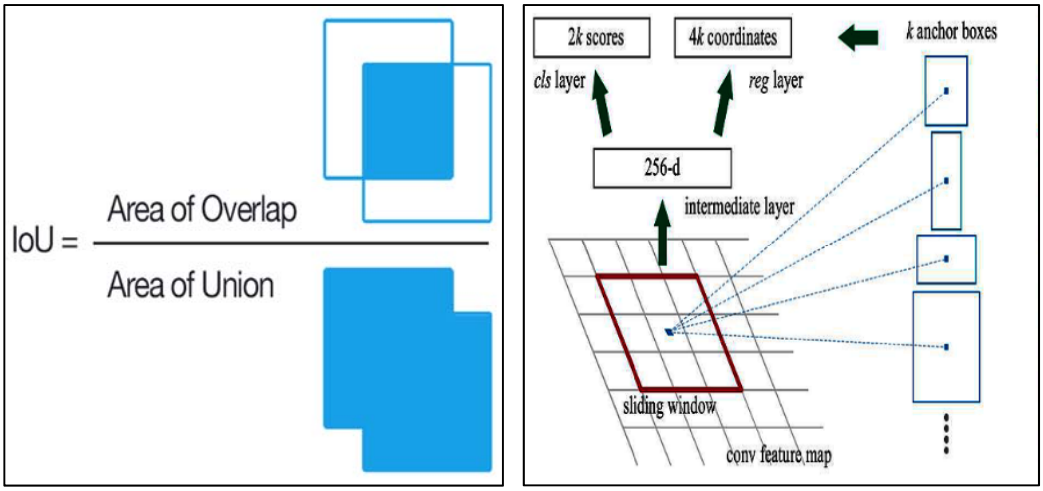
2.5.4 Training Procedure of RPN

The pre-trained classification neural networks models, such as ZF, VGG, are used to initialize the parameters of the designed architecture. Then, by training the model using the crack images, the feature maps can be obtained. As shown in Figure 2.24, the training process of RPN to obtain the feature maps includes the following steps:

(1) Calculating anchors: there are totally nine potential region proposals that should be considered for every location in an image. The sizes of the feature maps are a combination of images areas of $\{128^2,256^2,512^2\}$and the length-to-width ratios of {1:1,1:2,2:1}. Those areas with different scales and ratios are called anchors.

(2) Identifying the positive and negative regions: This step tries to find the relationship between the ground truth box and all the anchors. The following principles are used to identify the positive and negative regions:

   1) For each labelled ground truth box, regard the anchor that has the maximum Interest Over Region (IOU) with the ground truth (Figure 2.24a) as the positive anchor;
   2) For the rest of the anchors apart from 1), if IOU>0.7 the corresponding anchor should be considered positive.
   3) If an anchor has the value of IOU lower than 0.3, it should be labelled as negative;
   4) Leave out the rest of the anchors that are not included in 1), 2) and 3).



(a) Calculation of IOU **(b) RPN**

Figure 2.24: Illustration of IOU and RPN (Ren, He, Girshick, & Sun, 2015)

The loss function of the network is used to evaluate the performance of the proposed network. For every anchor, there are two SoftMax classifiers that can generate two output scores representing the probability (p_i) of whether it is the object or not. Meanwhile, a bounding box regressor is included for every anchor. This bounding box is defined by the four coordinates (t_i) of the anchor. Therefore, the total loss function of RPN can be defined by Equation (2.4):

$$L(\{p_i\}\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (2.4)$$

where i represent the $i-th$ anchor. If this anchor is positive, p_i^*=1, otherwise p_i^*=0. t_i^* represents the coordinates of the ground truth box that has a positive relationship with the anchor. N_{cls}, N_{reg} denote normalization factors, *i.e.* batch size and anchor numbers, for classification and regression respectively. L_{cls} represents classification loss evaluation part. $p_i^* L_{reg}(t_i, t_i^*)$ represents the regression loss evaluation part, which is calculated using the smooth function, Equation (2.5)

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if|x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (2.5)$$

Bounding box regression method is used for objective location or detection. The four coordinates for defining the bounding box of an object are calculated by transferring operations and scaling operations, defined in Equation (2.6):

$$\begin{matrix} t_x = \frac{x-x_a}{w_a} & t_y = \frac{y-y_a}{h_a} & t_x^* = (x^* - x_a)/w_a & t_y^* = \frac{y^*-y_a}{h_a} \\ t_w = \log\left(\frac{w}{w_a}\right) & t_h = \log\left(\frac{h}{h_a}\right) & t_w^* = \log\left(\frac{w^*}{w_a}\right) & t_h^* = \log\left(\frac{h^*}{h_a}\right) \end{matrix} \quad (2.6)$$

where x, y, w, and h represent the centre coordinates, width, and height of the bounding box. X, x_a, and x^* denote the coordinates of the predicted box, anchor box, and the ground truth box, respectively.

2.6. Chapter Summary

This chapter conducted a literature review regarding current pavement performance measurements, pavement surface distress types, KPIs, automated pavement distress detection and evaluation methods and the machine learning technologies in current engineering.

Pavement performance measurements include several surface distresses, rutting depth and the overall performance indices. Those measurements can be divided into surface defects, pavement profile conditions and the overall performance indices.

In terms of automated methods for pavement condition evaluation and performance analysis, 2D image processing-based pavement damage analysis methods can be used for the detection and evaluation of surface defects. This aspect of the research has been conducted for several decades throughout the world according to the literature. However, the measurements of pavement surface roughness condition require the information of pavement profile changing data, which is hard to obtain from the analysis of 2D pavement surface images. Therefore, most current automated pavement management systems only used the information obtained from the collected 2D images, which is less reliable.

Machine learning technologies have seen advanced development in recent years. Meanwhile, several engineering fields have adopted machine learning to improve the performance of their

traditional analysis systems. In literature, however, fewer researches have been found in applying machine learning techniques into pavement management. Therefore, extensive research on machine learning techniques-based pavement distress detection and performance analysis is highly required to fill this gap.

CHAPTER 3. Research Methodology

To address the gaps observed in the previous chapters, a comprehensive research methodology framework is proposed in Figure 3.1. This methodology framework includes the development of six analytical tools. These six analytical tools can be broadly classified into two categories: 1) Structured data modeling-based methods, which include historical data based IRI prediction, image indicator data-based network level performance prediction, dominant distress indicators analysis, and surface modulus based structured performance prediction; 2) Non-structured image data processing-based methods, which include deep convolutional neural network-based pavement surface crack detection and crack image segmentation.

3.1. Structured Data Analysis-based Methods

Structured data refers to the data that can be organized by the data table and stored in the spreadsheet. For the analysis of these kinds of data, shallow machine learning methods can be used.

In the development of historical data based IRI prediction model, the training and testing data are extracted from the InfoPave website. The raw IRI data stored in this database are obtained through automated pavement condition investigations. High speed road inspection analyzers are the main measurement devices. Therefore, both the left wheel path and the right wheel path IRIs are recorded for the same road. Hence the most significant aspect of the model developed in this chapter differs from traditional methods is that multi-granularity similarity analyses are considered. Meanwhile, the method considers the historical trends of the IRI changing as fuzzy time series. The reason lies in that IRI reflects the longitudinal profile changing of pavement surface over the pavement service time. These two aspects of considerations made the model developed in this thesis outperform traditional methods, which has been proved by the comparative experiment of Chapter 4.

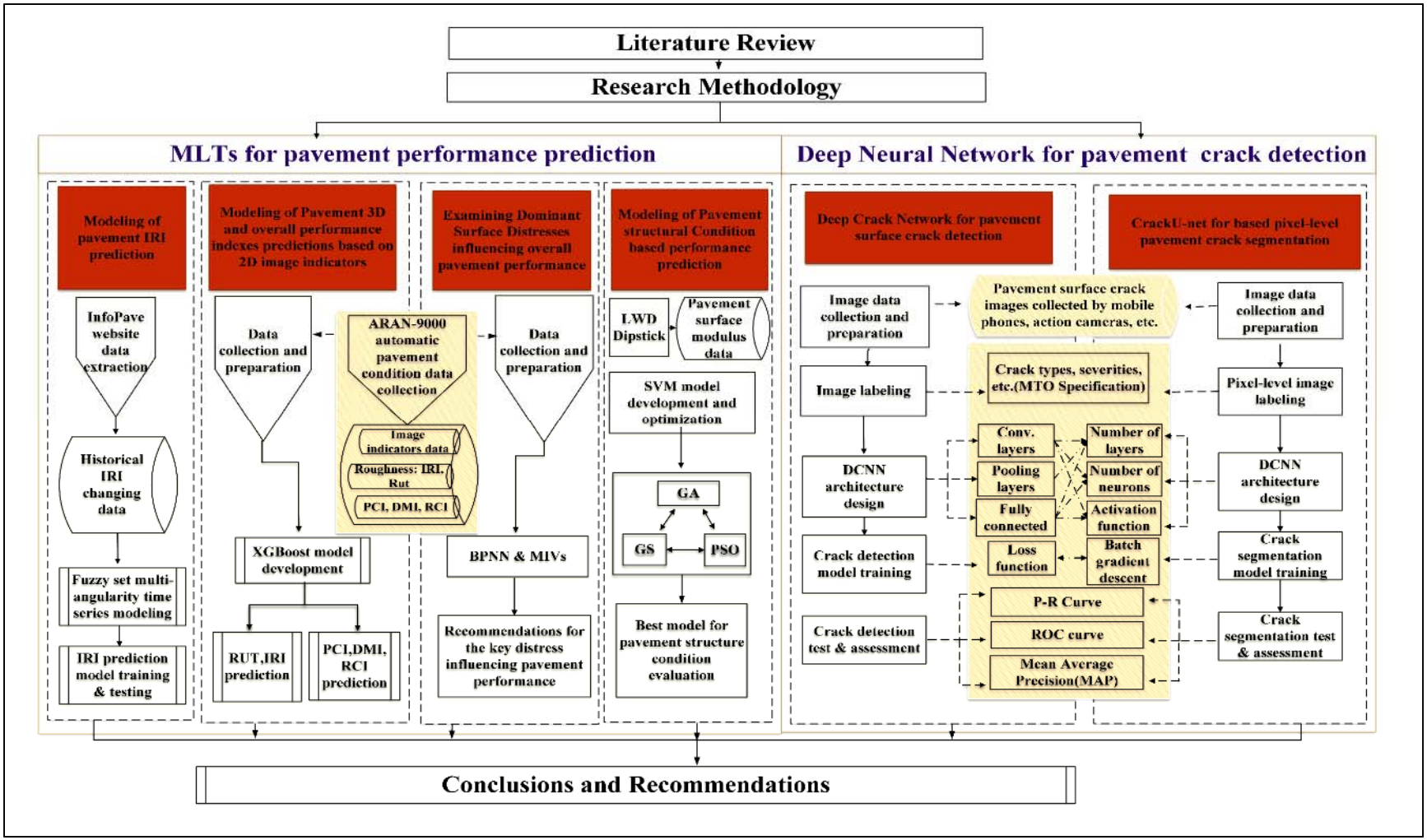


Figure 3.1: Methodology

The modeling of network level indices predictions using the image indicators aims to seek the answer of whether the overall pavement performance can be predicted without collecting road roughness data. Meanwhile, considering that pavement is a physical entity, the performance of the three dimensions should be correlated. The predictions of roughness condition indices based on image indicators are analyzed. Therefore, the input indicators are the pavement surface distress indicators, such as the length, width, percent, and severity of the cracks. The output data are the most widely used network level indices, including RUT, IRI, PCI, DMI, and RCI. All those data are collected using ARAN 9000 and provided by the technical staff of MTO. The state-of-the-art XGBoost method is used to achieve the objectives. Detailed methodologies, model training/testing, and the related experiments are discussed in Chapter 5.

Using the same data as the network level performance prediction which is discussed in Chapter 5, the impact condition of different surface distresses on the pavement overall KPIs are analyzed. Back Propagation Neural Networks (BPNNs) are being used to train the prediction models. On this basis, the Mean Impact Values are calculated for each input parameter. The values of MIVs reflect the impact condition of the corresponding input parameter on the output indices. Therefore, the dominant distress indicators can be obtained from the high impact image indicators. The obtained results can provide significant guidance to the M&R treatment scheduling. Detailed implementations of this part are discussed in Chapter 6.

The pavement structural performance prediction model is developed using the optimized SVM methods. This aspect of research aims to develop pavement surface modulus data-based performance prediction methods. Pavement surface modulus data are collected using LWD from the highway test sections. These surface modulus data can reflect the structural condition of the whole pavement layers at different points. Higher pavement surface modulus indicates better load transfer capacity of the whole pavement, hence representing

higher performance. According to this basic principle, the surface modulus is labeled into four categories with each representing a certain pavement service level. In the model training procedure, heuristic methods are used to optimize the parameters of the SVM models. Meanwhile, comparative experiments are carried out. The implementations of this part can be found in Chapter 7.

3.2. Non-structured Data(image)-based Methods

Non-structured data refers to the pavement surface images collected by different kinds of pavement condition survey devices. There are two significant differences between the image data and the structured data:

1) the sizes of the image data are much higher than the traditional structured data;

2) the image data are stored non-structured, and the images obtained from different devices are significantly different from each other.

Hence, using the traditional shallow machine learning methods is hard to obtain satisfactory results. With these considerations, the state-of-the-art deep convolutional neural networks are used to analyze pavement surface image data.

Pavement surface image-based road crack detection is one of the most broadly eye-catching research aspects. An innovative deep convolutional neural network-based crack detection model, CrackDN, is proposed. The proposed CrackDN is composed of two main parts, which are sensitivity detection network and the region proposal refinement network. The input training pavement surface images are collected from several commonly used devices, including high speed road monitor vehicles, smart phones, action camera, etc. The model is programmed by TensorFlow with GPU configuration. Adam method is selected for model optimization to find the best model parameters. The model is trained and tested on

the pavement surface crack images under complex background conditions with satisfactory performance. The detailed methodologies are explained in Chapter 8.

Image based Pixel-level pavement crack segmentation is another extensively developed research branch. Correctly segmented pavement crack information can guarantee the effectiveness of t= crack parameter calculation that follows. Therefore, reliability of the calculated overall network level pavement performance indices can be improved. To bring the state-of-the-art technologies into the application of pixel-level crack segmentation, innovative CrackU-net architecture is developed. CrackU-net follows the basic concept of deep convolutional neural network. The architecture is composed of several convolutional layers, pooling layers and up-convolutional layers forming the “U” shape. This kind of architecture shape can maintain the size of the input image unchanged. The final model realized crack segmentation accuracy around 0.99, and the average error around 0.03, which proved the satisfactory performance of the model. Chapter 9 provides a detailed explanation for this part.

CHAPTER 4. IRI Prediction Using Multi-Granularity Fuzzy Time Series Method

4.1. Introduction

In the 1970s, the World Blank proposed the use of International Roughness Index (IRI) to evaluate the roughness condition of the pavement surface. Since then, various kinds of mathematical models have been developed by researches for the prediction of IRI, thereby guiding maintenance prioritizations.

The Department of Transportation (DOT) of Ontario is a responsible for pavement maintenance and rehabilitation prioritizations and optimizations based on a limited budget. Therefore, using pavement performance models acts as a key in helping DOT to manage future expenses from an economical perspective. In the literature, several researchers across the world have made significant contributions to IRI predictions. Treat-effectiveness based, and performance deterioration-based models are the two main categories of the methods. Treatment-effectiveness based methods usually define a performance jump measurement to analyze the performance improvement before and after certain treatment activities. Performance deterioration-based methods usually develop a deterioration curve to predict pavement performance changing with time. Those IRI prediction models can be grouped into three categories: exponential function models, power function models, and polynomial function models. According to research results verified in the literature, exponential function based IRI prediction models outperform all the other kinds of methods (Hajj, Loria, Sebaaly, Borroel, & Leiva, 2011; Oliveira, Vedaie, & Kennedy Jr, 1990). However, climate related input variables are hard to obtain, and high computational complexity exists in current developed approaches.

However, for simplicity, most of the IRI prediction models in usage are developed based on traditional linear or nonlinear regression methods, which often fail to obtain satisfactory results if the roadways environment and the pavement types changes. For example, the IRI changing mechanisms between asphalt pavement and concrete pavement should analyzed by different methods. In recent years, highly advanced computerized technologies significantly have facilitated the tasks of developing more effective IRI prediction models. Machine Learning technologies embraced the advantages of supercomputer technologies to benefit both the scientific and engineering fields. Granular Computing (GrC) concepts provide a convenient way for multiple granularity spaces data analysis. Therefore, it is utilized in this chapter for more effective IRI analysis. Meanwhile, since IRI values are time dependent variables, fuzzy time series analysis method has been used. Firstly, an automatic clustering technique is utilized to form different intervals. Next, the second-order fuzzy trend matrix (SFTM) and fuzzy trend relationship classification (FTRC) are proposed to predict the fuzzy trend of each factor. After that, the fuzzy trend status for multiple granular spaces are generated with full consideration of several aspects of uncertainties. Finally, particle swarm optimization (PSO) technique is employed to optimize the IRI forecasting results.

4.2. IRI Prediction Framework

In this research, IRI values are considered as time dependent indices which tend to increase with the increase of pavement service life if no M&R activities being applied. Figure 4.1 shows the flowchart of the multi-granularity fuzzy time series based IRI prediction methodology developed in this research.

In the beginning, the raw historical IRI data are automatically assigned as principal factor granular space and the subfactor granular spaces. Next, automatic clustering operations are conducted for each granular space to obtain multiple intervals. Those intervals act as unit

tools for measuring the fuzzy trends of the IRI values. After that, the fuzzy trends values of each interval are calculated based on the proposed fuzzy trend relationship classification (FTRC) principle. Then, the fuzzy trend matrix is obtained, which is combined with the PSO technique to calculate the final predicted IRI value.

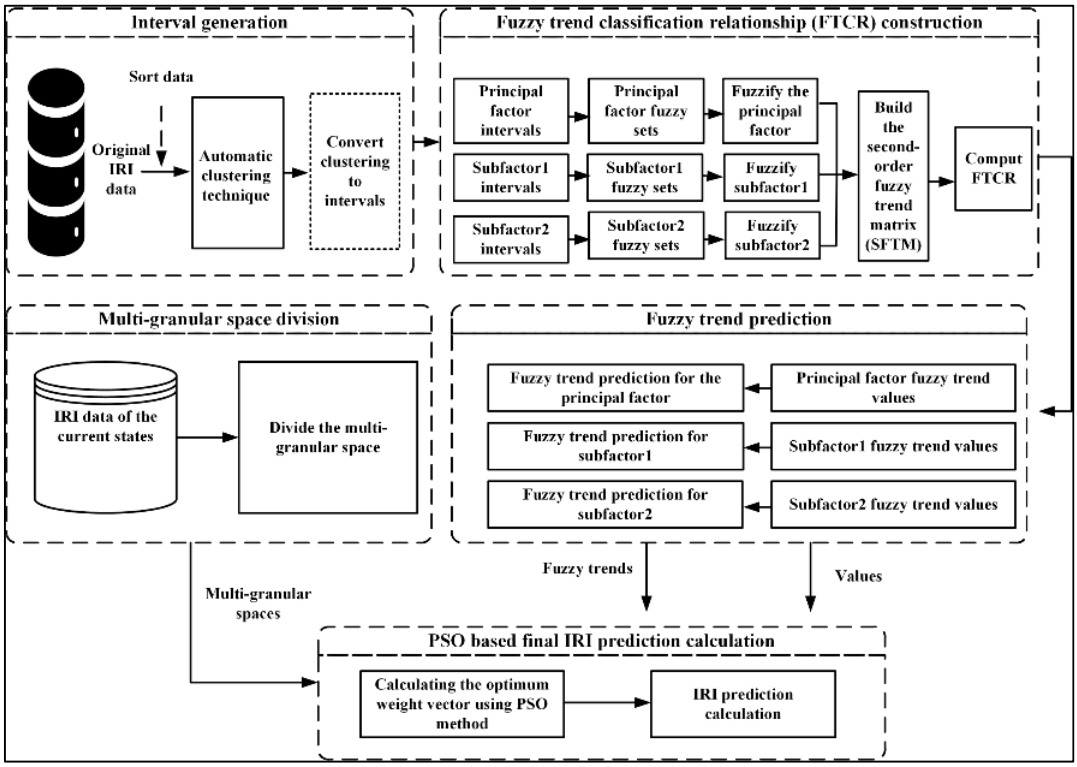


Figure 4.1: IRI prediction methodology

4.2.1 Fuzzy Time Series

The concept of fuzzy time series (Song & Chissom, 1993) is developed based on the definitions of fuzzy set theory, which was first proposed by Zadeh in 1965 (Sets & Zadeh, 1965). The definition of a fuzzy set is explained as follows:

Let U be a universe of discourse, which can be defined as $U=\{u_1,u_2,u_3,...,u_n\}$. Then,

a fuzzy set A which is within U can be expressed by Equation (4.1):

$$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + ... + f_A(u_n)/u_n \quad (4.1)$$

where f_A denotes the membership function. A denotes a fuzzy set. $f_A(u_i)$ denotes the membership degree of u_i, $1 \le i \le n$.

Assume that there is a group of fuzzy sets that have been predefined as $f_i(t)$ $(i = 1, 2, ...)$. On this basis, the n^{th} order fuzzy time series relationships for the state series of $F(t-1), F(t-2), ..., F(t-n)$ can be expressed by Equation (4.2):

$$F(t-n), F(t-n+1), ..., F(t-1) \rightarrow F(t) \quad (4.2)$$

where $F(t) = \{f_1(t), f_2(t), ...\}$ represents the collection of $f_i(t)$ $(i = 1, 2, ...)$. According to this, the collection can be called as a fuzzy time series set on the discourse of $Y(t)$ $(t = ..., 0, 1, 2, ...)$.

Therefore, if the values of $F(t) = A_j, F(t-1) = A_{i_1}, ... F(t-n) = A_{i_n}$ are given, then the logical relationship of Equation (4.3) can be obtained

$$A_{i_n}, ..., A_{i_2}, A_{i_1} \rightarrow A_j \quad (4.3)$$

where $A_{in}, ..., A_{i2}, A_{i1}$ denote the current state. A_j denotes the next status, which is to be predicted based on current state.

4.2.2 Auto-clustering Technique for Interval Division

Several intervals of the input data can be divided using Auto-clustering technique (ACT), which has been applied in different engineering fields. The framework of the ACT process

is illustrated in Figure 4.2. Data sorting, distance calculations, and CTI transformations are the key procedures for this methodology.

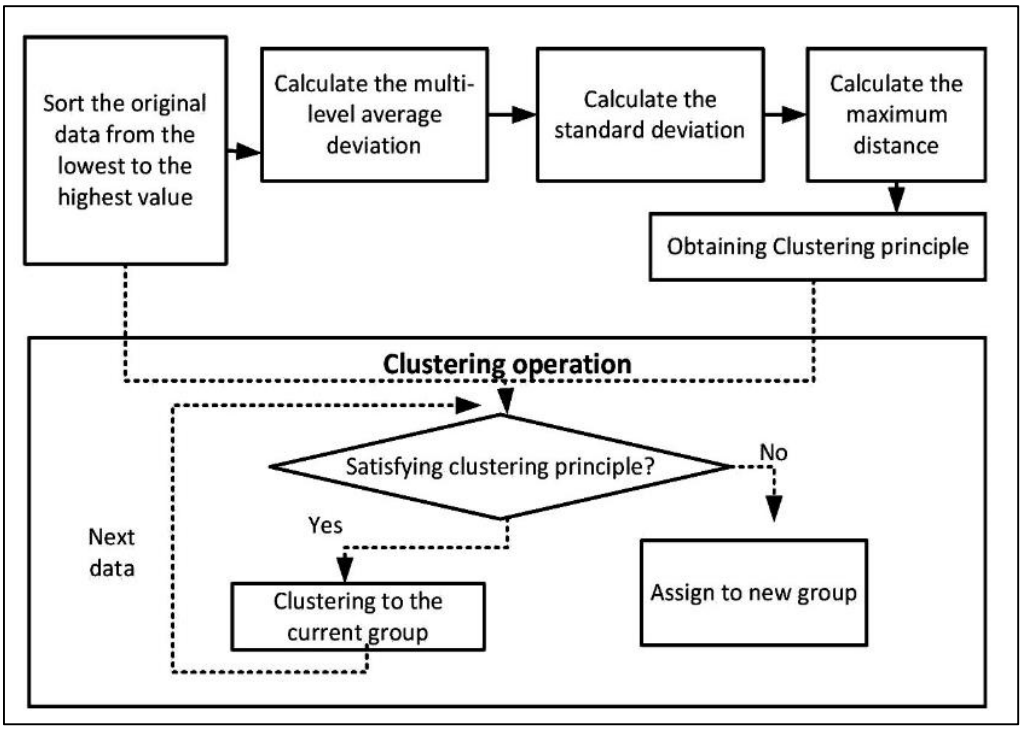


Figure 4.2: Framework of ACT

Step 1：Sort the original dataset from the lowest value to the highest value, the sorted data series can be expressed as:

$$D = \{d_1 < d_2 < d_3 < \cdots < d_n\} \tag{4.4}$$

where n denotes the number of total data points.

Step 2: Calculate two measurement criteria: the multi-level average deviation (avg_{diff}) and the standard deviation (dev_{diff}), which are expressed in Equation (4.5) and (4.6)

$$avg_{diff} = \frac{1}{n-1}\sum_{k=1}^{n-1}\frac{\sum_{i=1}^{n-k}(d_{i+k} - d_i)}{n-k} \tag{4.5}$$

$$dev_{diff} = \sqrt{\frac{\frac{1}{n-1}\sum_{k=1}^{n-1}\frac{\sum_{i=1}^{n-k}(d_{i+k} - d_i - agv_{diff})^2}{n-k}}{\frac{n(n-1)}{2}}} \tag{4.6}$$

Step 3: Calculate the maximum distance use Equation (4.7)

$$\max\_dis\tan ce = c \times dev_{diff} \tag{4.7}$$

where c is a constant value within the range of 0~1. In this research, $c = 0.5$ has been used.

Step 4: Assign the sorted data into different clusters with reference to the calculations of Step 2 and Step 3. Detailed process can be explained as:

(1) Denote the current cluster to be d_i. Then, use the principle of Equation (4.8) to judge whether the next data point should be considered as belonging to the current cluster. If the distance assessment satisfies this principle, then d_{i+1} is grouped into cluster d_i. Otherwise, assign a new cluster with the cluster centre of d_{i+1}.

$$d_{i+1} - d_i \le \max\_dis\tan ce \tag{4.8}$$

(2) Conduct this assessment to each data point until all of them have been processed. The clustered results can be obtained as Equation (4.9) expresses.

$$\{d_{11},...d_{1a}\},\{d_{21},...d_{2b}\},...\{d_{i1},...d_{in}\},...\{d_{j1},...d_{jm}\} \tag{4.9}$$

Step 5: Convert each cluster to an individual interval using the cluster to interval (CTI)

method illustrated in Figure 4.3.

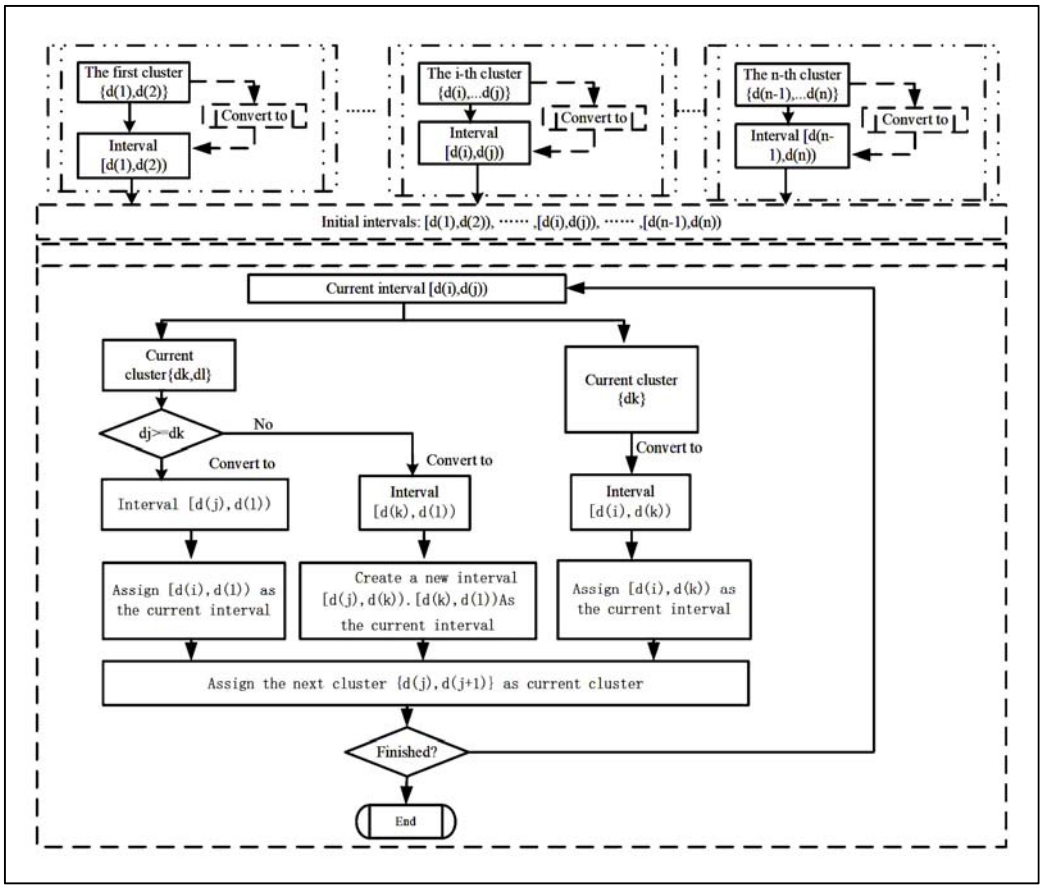


Figure 4.3: Methodology of CTI

In the beginning, the clusters can be automatically obtained from the ACT method, which can be denoted as $\{d_{11},...d_{1a}\},\{d_{21},...d_{2b}\},...\{d_{i1},...d_{it}\},\{d_{j1},...d_{jm}\}...\{d_{n1},...d_{ns}\}$. On this basis, the number of intervals is assigned in the same way as the number of clusters. The intervals can be denoted as $K_1, K_2, K_3,...K_i,..., K_n$. Then, the continuous intervals are generated based on three different cases: (1) Case I: The left bound of the first interval can be calculated by subtracting the maximum distance from the first cluster; (2) Case II: The right bound of the last interval can be calculated by adding the maximum distance to the last cluster; (3) Case III: The intervals between the first and the last ones. The value of left

bound in later intervals should be equal to the right bound of their previous interval, while the right bound values can be calculated by the average of their previous cluster and their next cluster. Finally, the divided continuous intervals can be obtained. The medium values of each interval can be calculated by the average of their left and right bound values.

4.2.3 Granular computing

Granular Computing (GrC) is a kind of intelligent learning concept which is proposed by simulating human thinking mechanism. GrC method has been used by several intelligent system design and implementation areas and received high evaluations. The most eye-catching advantage of GrC is that it can solve the problem with the consideration of both certainty and uncertainty conditions.

Three basic elements of GrC methods are granules, granular space, and their corresponding hierarchy structures. The granules are the unit elements being used in conducting granular computing. Granular spaces are the spaces populated by the granules. Meanwhile, several granular spaces can be combined to form more global sized granular spaces with hierarchy structures. Therefore, multi-perspective information of the data can be learned. In this research, the original data include three categories, which are the left wheel path IRI, right wheel path IRI, and the average IRI. According to this condition, three granular spaces are selected in this research. Meanwhile, in current automatic pavement performance assessment systems, the overall pavement performance indices are usually calculated based on the average IRI values. Therefore, the principal factor is assigned to the average IRI values, which represent the first granular space. Two subfactors are assigned as: (1) subfactor1: the second granular space: average IRI and one of the left or right wheel path IRI, and (2) subfactor 2: all three kinds of IRI measurements. The principal factor is the values to be predicted. The subfactors are the supplementary pieces of information that help the model to achieve more reliable results.

4.3. Implementation of Multi-granularity Fuzzy Times Series Based IRI Prediction

4.3.1 LTPP Data Extraction and Analysis

To provide reliable pavement data for long life pavement design and management, Long-Term Pavement Performance (LTPP) program collects comprehensive data of the studied regions annually since the initial construction. Different road types, climate conditions, traffic and roadway environment conditions have significant influences on the actual IRI distributions. Therefore, to provide a more robust method for IRI prediction, this research used data from different provinces/states. The data extraction, model development and programming are conducted by myself. The model optimization is conducted with the help of Wei Li at Chang' An University of China, and the results analyses are helped by my supervisor.

The climate conditions in the North America regions have a strong influence on pavement performance. Therefore, climate regions are always being selected as an important factor when conducting performance modeling and M&R prioritizations. On this consideration, this research selected IRI data from different climate regions with different roadway functional classes. As shown in Table 4.1and Figure 4.4, 38 provinces and states with functional classes of rural principal arterial, rural major collector, urban principal arterial, and rural minor arterial are considered.

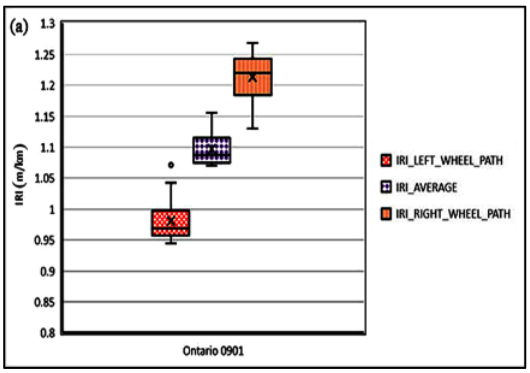
Two important phenomena have been observed by the analysis of the extracted IRI data. The findings are:

Figure 4.4: Regions of dataset selection

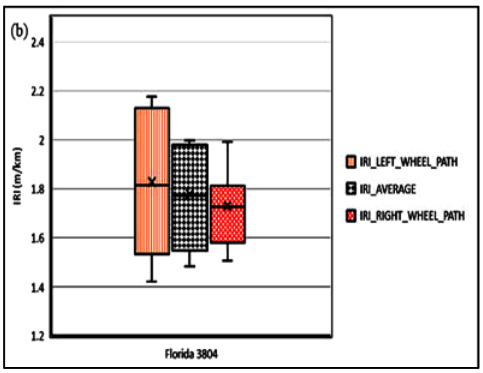
Table 4.1: IRI Data selection in different regions

Count	Climate Region	Roadway Functional Class
12	Wet, Freeze	Rural Principal Arterial
1	Wet, Non-Freeze	Rural Principal Arterial
16	Wet, Freeze	Urban Principal Arterial
		Rural Principal Arterial
		Rural Minor Arterial
		Rural Principal Arterial
		Urban Principal Arterial
9	Dry, Freeze	Rural Principal Arterial
		Urban Principal Arterial
		Rural Minor Arterial
		Urban Principal Arterial
		Rural Minor Arterial

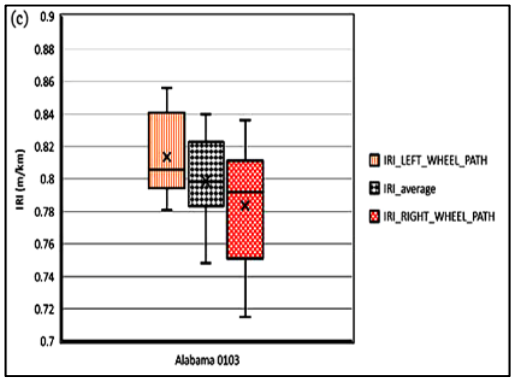
(1) Region differences can have a significant impact on the condition of IRI values, as can be seen in Figure 4.5, which provides examples of IRI distribution in 6 road sections. By comparing (a) (b) and (c), it can be found that in Ontario, the IRI values between the left and right wheel path have a significant difference. However, in Florida and Alabama, the IRI of the left and right wheel path are closer to each other. By observing the IRI distributions in Figure 4.5 (c), (d), (e) and (f), one can see the variance of IRI values between road sections of the same region, though these differences may not so significant than that observed between regions. Therefore, including IRI data of different regions as the influencing factors when doing model development should be considered.



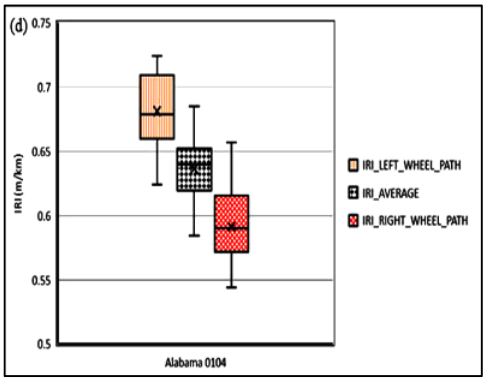
(a) Ontario 0901



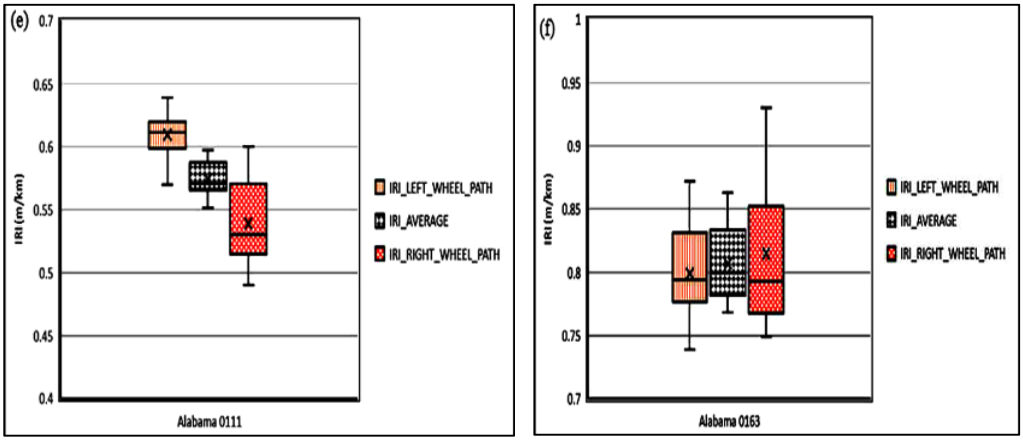
(b) Florida 3804



(c) Alabama 0103



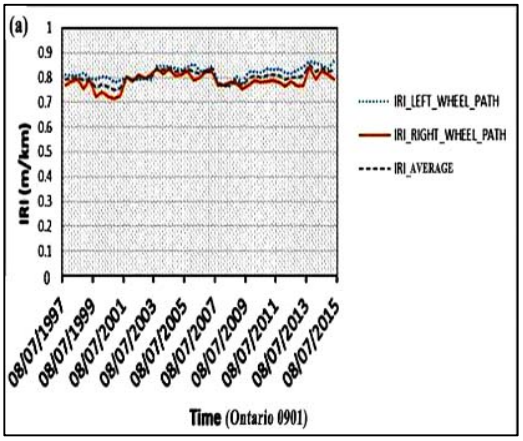
(d) Alabama 0104



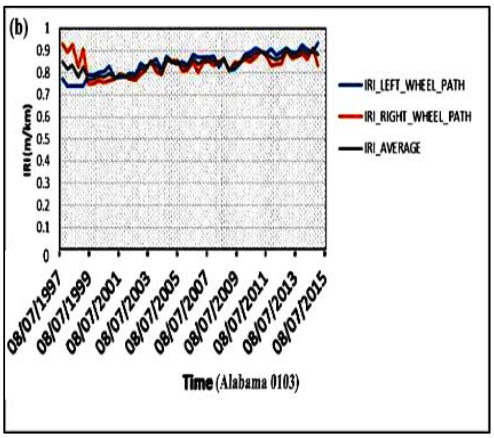
(e) Alabama 0111 (f) Alabama 0163

Figure 4.5: IRI distributions

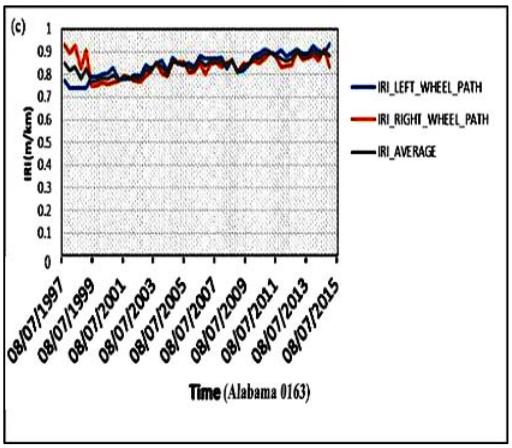
(2) Without timely road repairing being conducted, the IRI tends to increase with time. If proper road maintenance operations are conducted, the IRI values can be decreased immediately. At the same time, safer roads can be open to serve the traffic, and the service life of the corresponding pavement is prolonged. In the example shown in Figure 4.6, both road sections with and without repairing work being conducted are demonstrated. In section Alabama 0103 (Figure 4.6b), it can be seen that IRI increases gradually with the increase of road service life, though some fluctuations are observed. Those minor fluctuations might be caused by the variances of the environment conditions, or errors caused during data collections. In section Florida 3804 (Figure 4.6d), two road repairing implementations have been applied. The first one is crack sealing operation, which only changed the IRI from 2.5 to 2.0 in 2011. Two years later, in 2013, micro surfacing rehabilitation was conducted, which reduced the IRI to around 1.0. Therefore, timely and appropriate road maintenance activities have strong influences on the overall pavement performance. Thereby the initial purpose of long-life pavement can be achieved.



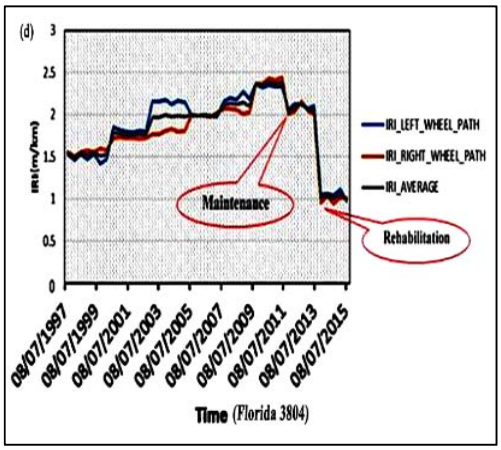
(a) Ontario 0901



(b) Alabama 0103



(c) Alabama 0163



(d) Florida 3804

Figure 4.6: IRI changing with time

4.3.2 Discourse Interval Division

In the beginning, a series of discourse intervals should be generated for both the principal factor and the subfactors. Firstly, the raw data should be arranged in an ascending sequence. Then, the ACT method is used to several data clusters. After that, the discourse intervals can be generated by the proposed CTI method.

Table 4.2 shows the interval division results for the IRI data collected from the road section of Ontario 0901. 7 intervals, 9 intervals, and 8 intervals are obtained for the principal factor, subfactor 1, and subfactor 2, respectively.

Table 4.2: The results of dividing intervals

Factor	Interval
Principal factor	[0.991,1.216], [1.216,2.182], [2.182,2.268], [2.268,2.653], [2.653,2.682], [2.682,2.917], [2.917,3.075]
Subfactor 1	[0.895,1.137], [1.137,1.791], [1.791,1.825], [1.825,1.926], [1.926,2.006], [2.006,2.07], [2.07,2.076], [2.076,2.146], [2.146,2.205]
Subfactor 2	[1.025,1.304], [1.304,2.561], [2.561,2.615], [2.615,3.207], [3.207,3.366], [3.366,3.732], [3.732,3.845], [3.845,4.027]

4.3.3 Establish Fuzzy Trend Relationship Classification (FTRC) Principle

In this stage, the fuzzy sets for IRI data are defined in the beginning. After that, the dataset is fuzzified based on data fuzzifying method. On this basis, the second-order fuzzy trend matrix (SFTM) is calculated. Finally, the FTRCs are assigned to each investigated dataset. Detailed operations are explained as follows:

1). In the beginning, the fuzzy sets of the principal and subfactors are defined by Equation (4.10) and Equation (4.11). Herewith, the fuzzy set of the principal factor is denoted as $A_1, A_2, \dots, A_p$. The fuzzy sets of the subfactors are denoted as $B_{j,1}, B_{j,2}, \dots, B_{j,mj}$

$$
\begin{aligned}
A_1 &= 1/u_1 + 0.5/u_2 + 0/u_3 + 0/u_4 + ... + 0/u_p \\
A_2 &= 0.5/u_1 + 1/u_2 + 0.5/u_3 + 0/u_4 + ... + 0/u_p \\
A_3 &= 0/u_1 + 0.5/u_2 + 1/u_3 + 0.5/u_4 + ... + 0/u_p \\
&..... \\
A_p &= 0/u_1 + 0/u_2 + 0/u_3 + 0/u_4 + ... + 1/u_p
\end{aligned}
\tag{4.10}
$$

$$
\begin{aligned}
B_{j,1} &= 1/v_{j,1} + 0.5/v_{j,2} + 0/v_{j,3} + 0/v_{j,4} + ... + 0/v_{j,m_j} \\
B_{j,2} &= 0.5/v_{j,1} + 1/v_{j,2} + 0.5/v_{j,3} + 0/v_{j,4} + ... + 0/v_{j,m_j} \\
B_{j,3} &= 0/v_{j,1} + 0.5/v_{j,2} + 1/v_{j,3} + 0.5/v_{j,4} + ... + 0/v_{j,m_j} \\
&... \\
B_{j,m_j} &= 0/v_{j,1} + 0/v_{j,2} + 0/v_{j,3} + 0/v_{j,4} + ... + 1/v_{j,m_j}
\end{aligned}
\tag{4.11}
$$

where u_i and v_{j,m_j} $i = 1,2,\dots,p$ denote the intervals.

2). Fuzzifying the IRI data. In this procedure, the originally collected IRI values are fuzzified according to interval assignments. First, all the data are evaluated and assigned to different intervals. Then, fuzzifying functions are used to calculate the fuzzified data of those intervals.

3). Construct the second-order fuzzy trend matrix (SFTM). Since the trend conditions (increase, decrease, or equal) of each logical relationship have to be counted, the SFTM is built based on the following two conditions:

a. For the principal factor, the SFTM can be represented as: $A_{i2}, A_{i1} \rightarrow A_m$. A_{i2}, A_{i1} and A_m represent the fuzzy sets of the principal factor at the $(\mathrm{t}-2)^{th}$, $(\mathrm{t}-1)^{th}$ and the t^{th} time points.

b. For the subfactors, the SFTM should be related to the principal factor. Hence, the SFTM of subfactors is expressed as $B_{j,k2}, B_{j,k1} \rightarrow A_m$. $B_{j,k2}$ and $B_{j,k1}$ represent the fuzzy sets of

the subfactors at the $(t-2)^{th}$ and $(t-1)^{th}$ time points. A_m represents the fuzzy set of the principal factor at the t^{th} time point.

4). Build the fuzzy trend relationship classification (FTRC). This research defined three groups, which are descending (G1), equal (G2) and ascending (G3). Specifically, Equation (4.12) is defined to construct FTRC for each data point.

$$\begin{cases} if\ \ i2 > i1 (or\ k2 > k1),\ classify\ SFTM\ \ to\ group\ 1 \\ if\ \ i2 = i1 (or\ k2 = k1),\ classify\ SFTM\ \ to\ group\ 2 \\ if\ \ i2 < i1 (or\ k2 > k1),\ classify\ SFTM\ \ to\ group\ 3 \end{cases} \tag{4.12}$$

Based on the method of Equation (4.12), the FTRC assignment principles are shown in Table 4.3.

Table 4.3: Fuzzy trend relationship classification principle

Classification	Relationship (i2/k2 and i1/k1)
G1	Descending
G2	Equal
G 3	Ascending

Consider road section of Ontario 0901, which is used as an example. The fuzzy trend classification results for each factor are shown in Table 4.4. It can be seen that most of the fuzzy sets are classified into the same group. This phenomenon is reasonable because the observed road section is under fair condition and the variances among all those measured IRI values are not significant. It should be noted that the first two fuzzy sets in Table 4.4 are the basis values that are used to classify the following fuzzy sets. Therefore, they cannot be assigned into any groups.

Table 4.4: Fuzzy trend relationship assignment method

Average IRI	Fuzzy set	Group	Left wheel path IRI	Fuzzy set	Group	Right wheel path IRI	Fuzzy set	Group
1.070999	A1	/	0.964999	B1,1	/	1.177000	B2,1	/
1.080000	A1	/	0.953999	B1,1	/	1.20599	B2,1	/
1.100999	A1	2	0.968999	B1,1	2	1.233000	B2,1	2
1.105999	A1	2	0.968999	B1,1	2	1.243000	B2,1	2
1.082999	A1	2	0.975000	B1,1	2	1.190999	B2,1	2
1.080000	A1	2	0.944000	B1,1	2	1.215999	B2,1	2
1.085999	A1	2	0.952000	B1,1	2	1.218999	B2,1	2
1.103000	A1	2	0.958000	B1,1	2	1.248999	B2,1	2
1.072000	A1	2	0.962000	B1,1	2	1.182000	B2,1	2
1.088000	A1	2	0.954999	B1,1	2	1.220999	B2,1	2
1.075999	A1	2	0.992999	B1,1	2	1.159000	B2,1	2
1.070000	A1	2	0.949000	B1,1	2	1.190999	B2,1	2

4.3.4 Fuzzy Trend Matrix Development

The fuzzy trend matrix is defined to predict the fuzzy trend of each factor. To ensure the accuracy of the prediction results, both values and the groups of observed IRI values are evaluated. Detailed process is discussed as follows:

Consider the principal factor whose SFTM can be represented as $A_{12,}A_{11} \rightarrow A_m$.

a. If $i_1 > m$, the method defined in Equations (4.13) through (4.16) should be used to calculate $S_{d(i)}$ and $N_{d(i)}$.

$$S_{d(i)} = S_{d(i-1)} + (mid\_value_m - mid\_value_{i1}) \quad (4.13)$$

$$N_{d(i)} = N_{d(i-1)} + 1 \quad (4.14)$$

$$S_{e(i)} = 0 \quad (4.15)$$

$$N_{e(i)} = N_{e(i\text{-}1)} + 1 \quad (4.16)$$

b. If $i_1 < m$, Equation (4.17) and (4.18) should be used to calculate $S_{a(i)}$ and $N_{a(i)}$

$$S_{a(i)} = S_{a(i-1)} + (mid\_value_m - mid\_value_{i1}) \quad (4.17)$$

$$N_{a(i)} = N_{a(i\text{-}1)} + 1 \quad (4.18)$$

The denotations of the parameters in Equations (4.13) - (4.18) are:

$S_{d(i)}$ = the sum of the descending trend for group i;

$N_{d(i)}$ = the total number of descending relations;

$S_{e(i)}$ = the sum of the equal trend for group i;

$N_{e(i)}$ = the total number of equal relations;

$S_{a(i)}$ = the sum of the ascending trend for group i;

$N_{a(i)}$ = the total number of ascending relations.

Based on the parameters calculated from the above procedure, the matrix shown in Table 4.5 can be obtained.

Then, the fuzzy trend prediction can be conducted based on the results of Table 4.5. Assuming that SFTM of the principal factor in the current status is G_i, the fuzzy trend at the next status can be calculated by Equation (4.19).

$$\Delta_{trend} = \frac{S_{d(i)}}{N_{d(i)}} \times \frac{N_{d(i)}}{N_{d(i)} + N_{e(i)} + N_{u(i)}} + \frac{S_{u(i)}}{N_{u(i)}} \times \frac{N_{u(i)}}{N_{d(i)} + N_{e(i)} + N_{u(i)}} \tag{4.19}$$

Table 4.5: Fuzzy trend matrix

Group	Relationship					
	Descending		Equal		Ascending	
	sum	count	sum	count	sum	count
G 1	$S_{d(1)}$	$N_{d(1)}$	$S_{e(1)}$	$N_{e(1)}$	$S_{a(1)}$	$N_{a(1)}$
G 2	$S_{d(2)}$	$N_{d(2)}$	$S_{e(2)}$	$N_{e(2)}$	$S_{a(2)}$	$N_{a(1)}$
G 3	$S_{d(3)}$	$N_{d(3)}$	$S_{e(3)}$	$N_{e(3)}$	$S_{a(3)}$	$N_{a(3)}$

The constructed fuzzy trend matrixes of the observed IRIs regarding the principal factor, subfactor 1 and subfactor 2 are shown in Table 4.6 (a), (b) and (c).

After that, the fuzzy trends can be calculated using the method of Equation (4.19). Based on those values, the forecasting trends of IRIs can be evaluated. The calculated trends for three factors are:

$$\Delta principal = 0.688,\ \Delta_{subfactor1} = 0.127,\ \Delta_{subfactor2} = 0.030$$

It can be seen that the fuzzy trends of both the principal factor and the two subfactors tend to increase at the next status. Moreover, the most significant increase seems to occur for the principal factor.

Table 4.6: Fuzzy trend matrix for three factors

(a) Principal factor

Group	Descending		Equal		Ascending	
	sum	count	sum	count	sum	count
1	0	0	0	1	1.375	1
2	-1.717	2	0	30	1.520	3
3	0	0	0	4	0.733	2

(b) Subfactor1

Group	Descending		Equal		Ascending	
	sum	count	sum	count	sum	count
1	-0.107	1	0	1	0.996	5
2	-0.725	3	0	20	1.483	4
3	-0.623	3	0	5	0.068	1

(c) Subfactor 2

Group	Descending		Equal		Ascending	
	sum	count	sum	count	sum	count
1	0	0	0	0	1.029	2
2	-0.148	1	0	31	1.211	3
3	-0.323	1	0	4	0.656	1

4.3.5 Multi-granularity Spaces Fuzzy Trend Calculation

Assume that n subfactors are selected, then and the dataset should be divided into $(n+1)$ granular spaces. Therefore, three granular spaces should be assigned to the IRI data, which are:

(1) First granular space: principal factor (average IRI);

(2) Second granular space: the principal factor the one of the subfactor (average IRI and the left or right wheel path IRI);

(3) Third granular space: principal factor and the two subfactors.

In general, considering the fuzzy trend prediction of the $(m+1)^{th}$ granular space. Equations (4.20) -(4.23) can be used to calculate the fuzzy trend in the current state (the $(t-1)^{th}$time point) and the predicted value in the next state (the t^{th} time point)

$$FW_M = \frac{1}{1+\sum_{j=1,2,...,m.r_j>0} r_j} \tag{4.20}$$

$$FW_{s(j)} = \frac{r_j}{1+\sum_{j=1,2,...,m.r_j>0} r_j} \tag{4.21}$$

$$\Delta_{m+1} = FW_M \times \Delta_M + \sum_{j=1}^{m} (FW_{s(j)} \times \Delta_{s(j)}) \tag{4.22}$$

$$R_{m+1}(t) = R(t-1) + \Delta_{m+1} \tag{4.23}$$

where $R(t-1)$ =the value of the principal factor in the current state;

Δ_m= the fuzzy trend of the principal factor.

$\Delta_{S(j)}$= the fuzzy trend of the j^{th} subfactor.

r_j = the average fuzzy trend variation correlation coefficient of the principal factor ($V_M = (S_{d(i)} / N_{d(i)}, S_{e(i)} / N_{e(i)}, S_{u(i)} / N_{u(i)})$) and the j^{th} subfactor ($V_{S(j)} = (S_{d(k)} / N_{d(k)}, S_{e(k)} / N_{e(k)}, S_{u(k)} / N_{u(k)})$).

According to multi-granularity space analysis method discussed previously, the fuzzy tends for different granular spaces are calculated. Then, the forecasted IRI values in different granular spaces can be obtained. These results are:

$$\Delta_1 = 0.688, \Delta_2 = 0.4145, \Delta_3 = 0.248$$
$$R_1(t) = 3.684, R_2(t) = 3.411, R_3(t) = 3.244$$

4.3.6 Best Weight Vector Calculation Based on PSO Method

In this stage, the method of finding the optimal transmit weight vector is introduced. The elements of the weight vector represent the transition parameters from the previous state to the future state. Based on the weight vector, the predicted value can be calculated. The processes in this stage include the procedures as Figure 4.7 shows.

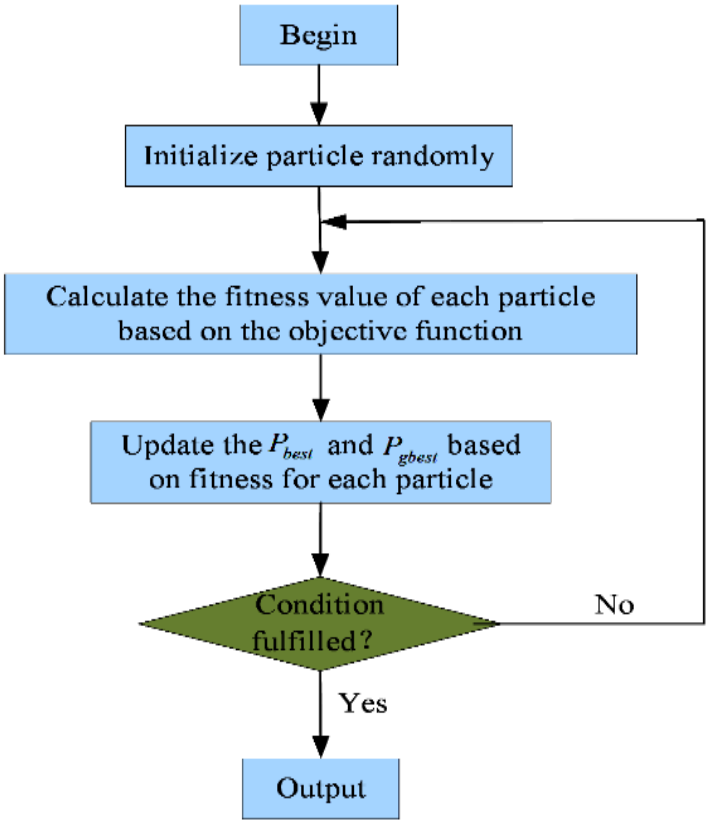


Figure 4.7: PSO algorithm

At first, assume the collected historical dataset can be defined by Equation (4.24), which can be used to train the model and get the best weight vector.

$$X = \begin{bmatrix} R_1(t-1) & \dots & R_{m+1}(t-1) & \dots & R_{n+1}(t-1) \\ \dots & & & & \\ R_1(t-i) & \dots & R_{m+1}(t-i) & \dots & R_{n+1}(t-i) \\ \dots & & & & \\ R_1(t-p) & \dots & R_{m+1}(t-p) & \dots & R_{n+1}(t-p) \end{bmatrix}, \quad Y = \begin{bmatrix} R(t-1) \\ \dots \\ R(t-i) \\ \dots \\ R(t-p) \end{bmatrix} \tag{4.24}$$

where X = the predicted value matrix.

Y = the ground truth.

p = the total observed time points.

$(n+1)$ = the total number of granular spaces.

$R_{m+1}(t-i)$ = the predicted value of the $(m+1)^{th}$ space at the i^{th} time point

$R(t-i)$ = the ground truth values at the i^{th} time point

Based on the constructed matrix, the PSO method can be used to find the best weight matrix. The particles should be generated at the beginning of each granular space. Then, the position and the speed of each particle can be updated according to the optimization of the fitness function. Herein, the following denotes are used:

$W_i = \{w_{i,1}, w_{i,2}, w_{i,3}, ..., w_{i,n+1}\}(w_{i,j} \in [0,1], w_{i,1} + w_{i,2} + w_{i,3} + ... + w_{i,n+1} = 1)$: the position vector of the i^{th} particle;

$V_i = \{v_{i,1}, v_{i,2}, v_{i,3}, ..., v_{i,n+1}\}(v_{i,j} \in [0,1], i \le j \le n+1)$: the speed of the i^{th} particle

The velocity and position of each particle are updated by Equations (4.25) and (4.26)

$$V_{i,t} = w \times V_{i,t-1} + C_1 \times r_1 \times (P_{best,i} - W_{i,t-1}) + C_2 \times r_2 \times (P_{gbest} - W_{i,t-1}) \quad (4.25)$$

$$W_{i,t} = W_{i,t-1} + V_{i,t} \quad (4.26)$$

where w is the inertia weight parameter. C_1 and C_2 are the constant values for the speed updating. r_1 and r_2 are random numbers that follow Normal distribution. $P_{best,i}$ denotes the best position vector of the i^{th} particle. P_{gbest} denotes the best position vector of all the particles.

The fitness function of Equation (4.25) is defined to find the best weight values.

$$f = \frac{1}{n+1} \times (Y - X \times W_i) \tag{4.27}$$

where f denotes the fitness value. X is a matrix which is formed by the predicted values of p days for each granular space. Y is the corresponding ground truth value.

In the beginning, the W_i and V_i are initialized with random values. Then, $P_{best,i}$ and P_{gbest} are updated following two judgement methods, which are:

1) If $f < P_{best,i}$ in the current iteration, then assign $P_{best,i} = W_i$;

2) If $P_{best,i} < P_{gbest}$, then assign $P_{gbest} = P_{best,i}$.

After that, use Equations (4.25) and (4.26) to update V_t and W_t. The updating operations can be finished when the predefined maximum iteration is reached. Then, P_{gbest} is regarded as the final best weights vector.

When the prediction results of all the particles are obtained through the previous PSO steps, the final predicted changing trend and the predicted value can be calculated by the Equations (4.28) and (4.29)

$$\Delta_z = \sum_{i=0}^{n} w_{i+1} \Delta_{i+1} \tag{4.28}$$

$$FR(t) = R(t-1) + \Delta_z \tag{4.29}$$

where $FR(t)$ denotes the final predicted value. Δ_z denotes the changing trend from the $(i-1)^{th}$ time point to the i^{th} time point. To be more specific,

$\Delta_z > 0$ indicates an ascending tendency;

$\Delta_z < 0$ indicates a descending tendency;

$\Delta_z = 0$ indicates no significant changing between two time points.

In this research, 25 particles are defined to find the best weight vector. Optimum results were obtained when processing used1200 iterations. The optimal weighting vector here obtained is $w = (0.26, 0.11, 0.63)$. The final predicted IRI value can be obtained based on Equation (4.28) and (4.29), which are:

$$\Delta_z = 0.26 \times 0.6875 + 0.11 \times 0.4145 + 0.63 \times 0.2477$$

$$FR(t) = 2.996 + 0.3804$$

where the calculated value Δ_z indicates that IRI tends to increase since it is higher than 0. FR is the forecasted IRI value at the next state.

4.4. Comparative Experiments and Model Assessment

In this part, the effectiveness of the proposed method is compared with polynomial fitting algorithm, autoregressive integrated moving average (ARIMA) method and back-propagation neural network (BPNN) method. 20% of the total dataset is used as the test dataset. Those methods are compared by calculating the two most widely used errors indices. Root mean squared error (RMSE) can provide the overall error of the model in the test set. Relative error (RE) can give insight to the extent of the errors by the percent of the difference between the forecasted values and the corresponding ground truth values. Equation (4.30) and (4.31) are the methods for calculating these two indices.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (p_i - gt_i)^2}{N}} \tag{4.30}$$

$$RE = \frac{|p_i - gt_i|}{gt_i} \times 100\%, i = 1, 2, 3, \ldots, N \tag{4.31}$$

where N (equals to 1566 in this research) is the number of the data in the dataset. p_i and gt_i are the forecasted and the ground truth values for the i^{th} data point.

The errors of each method are shown in Figure 4.8. The proposed method outperforms all the other approaches for both error measurements. ARIMA method can also provide acceptable results with RE equals to 6.44% and RMSE equals to 0.193. Polynomial fitting method ranks with higher than 10% of error being observed. BPNN method is slightly better than the polynomial method. These results indicate that traditional linear fitting method should not be selected in pavement management systems for performance prediction given that enough historical data can be obtained for machine learning based model training. By this way, the reliability of the system can be maintained, and effective maintenance activities can be scheduled.

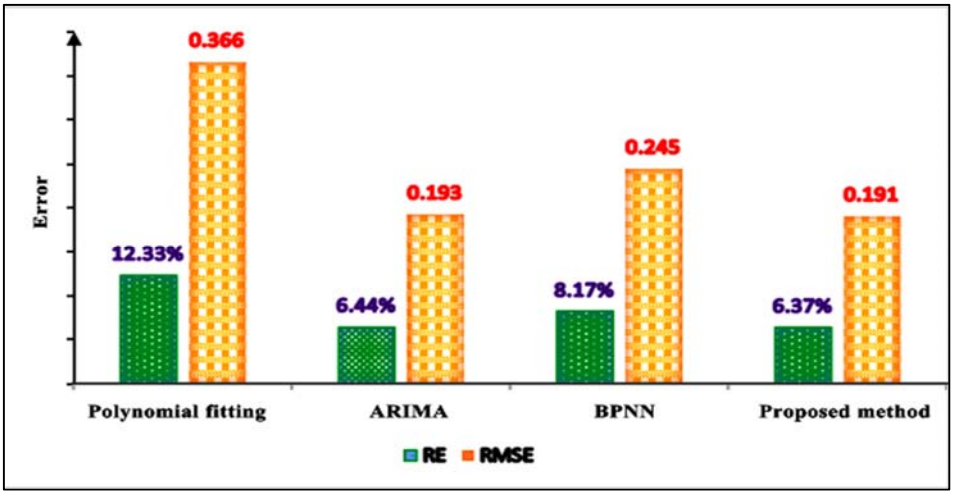


Figure 4.8: The performance of each method

4.5. Chapter Summary

An innovative IRI prediction model is developed in this chapter. The model is composed of the processing of automatic clustering, fuzzy trend analysis, multi-granularity similarity analysis, and PSO optimization.

Original IRI data are divided into different lengths of intervals using ACT. On this basis, three 3×6 fuzzy trend matrixes are constructed to forecast the fuzzy trend. The method being used is defining the fuzzy sets and establishing a fuzzy trend relationship classification. To deal with the uncertainty and ensure the model accuracy, initial IRI data are divided into three granular spaces. Then, the IRI predictions are conducted in every granular space. The final predicted IRI is obtained through particle swarm optimization technique.

The errors of the proposed model are compared with the traditional polynomial fitting method, ARIMA method, and BPNN model. Results showed that the proposed method can provide the lowest RMSE (0.191) and RE (6.37%), which proved the effectiveness of the proposed approach.

CHAPTER 5. Network Level Performance Indices Prediction Based on Image Indicators

5.1. Introduction

Optimizing budget allocation strategy and providing the best available level of services to the users require effective multi-layer pavement preservation plans. The effective plan of treatment activities depends not only on the current condition of the pavement but also depends on the reliability levels of the performance forecasting in the future years.

Figure 5.1 shows the significance of performance prediction for the service life of different pavements. As this figure shows, pavement A and B are in the same condition level currently. However, they might have different ages, suffer from a different environment and roadway conditions, which resulted in different deterioration speeds. Pavement A has a much lower deterioration speed than pavement B, therefore it can service traffic longer before it arrives the minimum acceptable level. In other words, pavement A has more remaining service life than pavement B. Therefore, pavement B should be applied with preservation or rehabilitation treatments earlier. This awareness is based on effective performance prediction.

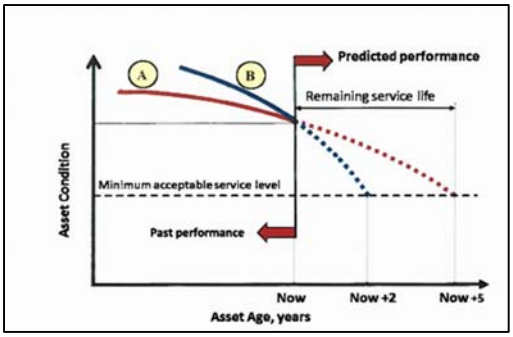


Figure 5.1: Need for pavement performance prediction (Tighe, 2013)

In terms of network level pavement management, the most widely used KPIs in current pavement management systems are RUT, IRI, DMI, PCI, and RCI. More specifically, these indexes can be classified as pavement deformation indices and the overall performance indices. The deformation indicators include RUT and IRI, which indicate the transverse and longitudinal profile variations of the pavement surface. These two deformation indices cannot be directly measured based on the analysis of 2D images. DMI, PCI, and RCI can be considered as the overall performance indices.

Rut depth is measured at the network level using high-speed devices. When driving through a certain pavement, the transverse profiles of the left and right wheel paths can be automatically measured, which are recorded as rutting depth in mm. In current PMS, IRI is usually measured by automatic high-speed equipment, which is called inertial profiler. The collected IRI can be directly used to evaluate the roughness condition of pavement. Or it can be converted to RCI if the calculation of PCI is required. DMI is the overall road surface performance evaluation index, which is calculated with of the aim of supporting the calculation of PCI. DMI can also be used in individual pavement sections for the assessments of lower-class roads.

5.2. Existing KPIs Used in Current PMS and the Formulations

DMI is a subjective index evaluating the surface condition of the pavement. RCI is an objective index which is calculated based on the measured IRI on provincial highways. PCI is the overall pavement performance condition index used by MTO, which is calculated based on the RCI and DMI values. Therefore, PCI is partially subjective, which is caused by the subjective characteristic of DMI.

The calculations of different KPIs are related to different pavement surface types. In Ontario, the calculation methods used in current PMS were defined by MTO in 2009. There

are minor differences between different surface types of pavement with regard to the calculation of the same index.

1) DMI formula

Equation (5.1) expresses the detailed methods of calculating DMIs for different types of pavements:

$$\begin{cases} ACpavement: \quad DMI = 10 * (208 - \sum_{k}^{N} (S_k + D_k) \times W_k) / 208 \\ PCCpavement: \quad DMI = 10 * (196 - \sum_{k}^{N} (S_k + D_k) \times W_k) / 196 \\ CompositePavement: \quad DMI = 10 * (220 - \sum_{k}^{N} (S_k + D_k) \times W_k) / 220 \\ SurfaceTreatedPavement: \quad DMI = 10 * (135 - \sum_{k}^{N} (S_k + D_k) \times W_k) / 135 \end{cases} \tag{5.1}$$

where N = the number of distresses counted for a specific pavement type

S_k = the severity level of the distress k

D_k = the density of the distress k

W_k = the weight factor of the distress k

It should be noted that the S_k and D_k values used in the above formula are not directly from the survey results. What's more, they must be modified from the original survey results based on the following principles:

(1) Let SEV_k be the severity rating of the distress *k* from the initial survey. If $SEV_k = 1$, then $S_k = 0.5$; If $SEV_k \geq 2$, then $S_k = SEV_k - 1$.

(2) Let DEN_k be the density rating of the distress *k* from the initial survey. If $DEN_k = 1$, then $D_k = 0.5$; If $DEN_k \geq 2$, then $D_k = DEN_k - 1$.

However, when dealing with surface treated pavements, S_k and D_k are used directly from the survey values. That means $S_k = SEV_k$ and $D_k = DEN_k$.

Detailed pavement condition survey forms can be seen in Appendix I.

2) RCI formula

The RCI values for different kinds of pavements are calculated from the measured IRI. Once the IRI values are obtained, formulas in Equation (5.2) are used to calculate the corresponding RCIs for different pavements:

$$\begin{cases} ACpavement: \quad RCI = Max(0, Min(10, 8.5 - 3.02 \times \ln(IRI))) \\ PCCpavement: \quad RCI = Max(0, Min(10, 10.37 - 4.77 \times \ln(IRI))) \\ CompositePavement: \quad RCI = Max(0, Min(10, 8.49 - 2.44 \times \ln(IRI))) \\ SurfaceTreatedPavement: \quad RCI = Max(0, Min(10, 9.94 - 3.46 \times \ln(IRI))) \end{cases} \tag{5.2}$$

Therefore, RCIs are within the range of 0-10. RCI=10 indicates excellent riding quality while RCI=0 represents the worst riding quality.

3) PCI formula

In Ontario, the most widely used index is PCI, which is an overall measurement calculated based on DMI and RCI. The formulas used to calculate PCIs for different types of pavements are shown in Equation (5.3):

$$\begin{cases} ACpavement: \quad PCI = Max(0, Min(100, 13.75 + 9 \times DMI - 7.5 \times e^{(8.5 - RCI)/3.02})) \\ PCCpavement: \quad PCI = Max(0, Min(100, 35.5 + 7 \times DMI - 11 \times e^{(10.37 - RCI)/4.77})) \\ CompositePavement: \quad PCI = Max(0, Min(100, 20.5 + 8.5 \times DMI - 11 \times e^{(8.49 - RCI)/2.44})) \\ SurfaceTreatedPavement: \quad PCI = Max(0, Min(100, 12.75 + 9 \times DMI - 5.5 \times e^{(9.94 - RCI)/3.46})) \end{cases} \tag{5.3}$$

Moreover, RCI is calculated from the measured IRI in the specific pavement. Therefore, Equation (5.3) can be revised as:

$$\begin{cases} ACpavement: \quad PCI = Max(0, Min(100, 13.75 + 9 \times DMI - 7.5 \times IRI) \\ PCCpavement: \quad PCI = Max(0, Min(100, 35.5 + 7 \times DMI - 11 \times IRI) \\ CompositePavement: \quad PCI = Max(0, Min(100, 20.5 + 8.5 \times DMI - 11 \times IRI) \\ SurfaceTreatedPavement: \quad PCI = Max(0, Min(100, 12.75 + 9 \times DMI - 5.5 \times IRI) \end{cases} \tag{5.4}$$

It is noteworthy that PCI values vary from 0 to 100 based on the performance condition of the pavement. PCI=0 indicates failure of the pavement, while PCI=100 indicates excellent pavement.

5.3. Trigger Values and Performance Prediction

It is impossible for all pavements to be maintained under excellent condition throughout their service life, though it might be desirable from all aspects. Usually, large quantities of the capital investments should be assigned to pavement treatments in order to maintain their performance under acceptable condition. Therefore, with the deterioration of pavement performance, trigger values should be defined for the cost-effective selection of appropriate pavement repairing operations. Figure 5.2 shows a demonstration for different types of pavement service levels with the corresponding trigger values.

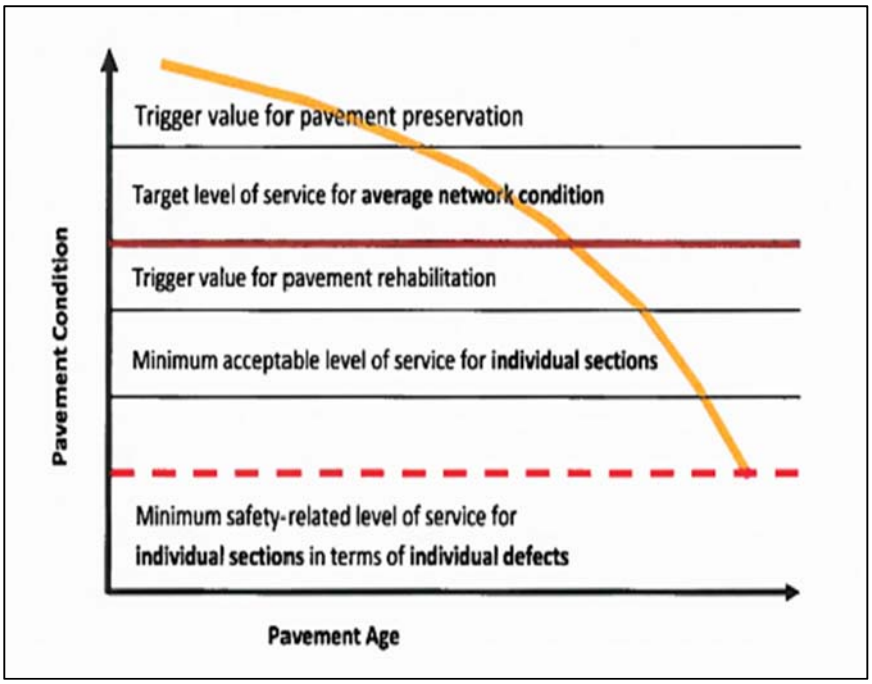


Figure 5.2: Types of pavement serviceability condition and trigger levels (Tighe, 2013)

As shown in Figure 5.2, the trigger values are the performance drop limitations defined for timely pavement maintenance & rehabilitation treatments. In other words, when the performance measurement indices drop/increase to the certain trigger values, the corresponding treatments should be applied to improve pavement condition and prolong the service life.

Since different KPIs reflect pavement performance in different perspectives, they are therefore used as guidance for different pavement treatments. Generally, PCIs are used for the scheduling of pavement rehabilitation treatments, such as milling and overlay. Several other treatment specific trigger values are used for pavement preservation maintenance, such as crack sealing and seal coat. Those trigger values are defined to provide detailed pavement preservation suggestions at the best suitable time, preventing the pavement from deteriorating to a more severe condition which requires more expensive treatment.

In addition to PCI, other most widely used indicators being utilized for trigger value-based treatment planning include RCI, IRI, Structural Adequacy Index (SAI), Surface Distress Index (SDI) and composite indices. According to different requirements and budget conditions, municipalities, provinces and other agencies have various preferences in selecting the condition indicators. The percentage of indicator selection preferences in different agencies of Ontario is shown in Table 5.1, which derives from a survey of Canadian agencies conducted in TAC 2013 (Transportation Association of Canada (TAC), 2013).

Table 5.1: Percent of pavement performance indicators used in Canada (Tighe, 2013)

Condition Indicator	Percent of Agencies Using Specific Indices	
	Municipalities (%)	Provincial/Federal/Territorial (%)
RCI	40	54
IRI	60	85
SAI	47	31
SDI	73	69
Composite Index	60	62

The pavement serviceability level should be established based on pavement condition indicators before assigning certain trigger values. Meanwhile, the establishment of pavement performance measures, service levels, and the trigger values should be based on the various decision levels, pavement types and priority conditions. Roughly speaking, the pavement service levels are classified into the target level, minimum acceptable level and

the safety level. Target level is the desired service level which usually should be exceeded in order to ensure that the pavement can function smoothly for a long time. The minimum acceptable level (or mandatory level) refers to the service level which the specific pavement must meet when it is supposed to open to traffic. The safety level is not used for the overall performance assessment. Therefore, the target level and the minimum acceptable level are the key values for trigger value establishment, since the pavements are usually divided into several segments as individual survey units. Then pavement condition rating and index calculations are conducted within individual segments. Therefore, the averaged values of all the individual segments in the same pavement are used to define the service levels for different functional classes of pavement. Table 5.2 shows an example of the target level assignment for different types of pavements. Multi-level of serviceability definition provides detailed references for pavement maintenance operations from individual roads upward to high-level pavement networks. Meanwhile, the treatments should be conducted on individual road sections firstly if their service conditions are below these limitations.

Table 5.2: Example levels of service for a pavement network (Transportation Association of Canada (TAC), 2013)

Roadway Type	Target or Desirable (Average PCI for all sections)	Minimum Acceptable (Average PCI for all sections)	Minimum Acceptable (PCI for individual sections)
Arterial	80	65	55
Collector	70	60	45
Local	60	55	40

5.4. Problems and Research Gaps

A comprehensive evaluation of road condition is critical to the design, construction, and maintenance of long-life pavement, which is also an important research area of all road maintenance departments, universities, and other agencies. Traditional manual condition survey and road damage condition measurement approaches are inefficient, subjective and less accurate. Most of the current road condition data collection, performance evaluation and treatment priority scheduling are conducted by automated or semi-automated road performance inspection equipment. The successful application of those technologies is facilitated by the advancement of computerized digital image processing methodologies. Generally, the automated road management systems can be divided into 2D and 3D pavement monitoring systems based on their differences in the data collection. Therefore, the following gaps can be summarized from the current pavement condition assessment and treatment of prioritization systems:

(1) The 2D detection method is mainly based on mature 2D digital image processing technology. However, lacking access to 3D depth information of the pavement leads to the low detection accuracy, unsatisfactory performance evaluation and the less effective treatment prioritization characteristics of the traditional 2D based methods. In comparison, the 3D equipment can acquire road condition information in more comprehensive ways, which however, are cost expensive and computationally complex. Most of the world's automated road performance detection and management systems are still using 2D image processing methods.

(2) The pavement deformation condition should be included in the performance indices to provide a more reliable overall performance evaluation. That’s why IRI or rutting depth values are significant components of the network level KPIs, such as DMI, RCI, and PCI. However, single 2D image processing cannot get access to depth information directly.

Therefore, it can be beneficial if the model is developed with the ability to predict the pavement depth deformation condition indices and the network level indices based on image indices, such as length, width and the percent of surface cracks.

(3) In recent years, highly advanced AI technologies have been successfully utilized in several areas for intelligent object detection and feature extraction. However, a few types of research are conducted in using machine learning techniques for intelligent pavement management. It is promising that machine learning technique based automatic pavement management techniques can benefit both the academic institution and industry applications to a large extent.

Therefore, taking the advantages of machine learning techniques into account, this research aims to provide the pavement management departments with the ability to predict the pavement network level performance indices through learning knowledge from 2D image indices.

5.5. XGBoost Method for Network Level Indices Prediction

5.5.1 Feature Engineering

The model training and test data used in this research are collected by ARAN-9000 automated pavement inspection system. To provide a more comprehensive analysis of the monitored pavement sections and generate enough model training data, initial data are aggregated with 10m per unit for indexes calculations.

5.5.1.1 Raw data analysis

After initial observation of the dataset, it is found that the database retains a large amount of auxiliary information other than road damage condition data from the automatic data

collection equipment. The auxiliary information mainly includes the vehicle type, the data collection date, GPS location information, and so on. Those pieces of information are necessary for offline pavement management and network-level decision making, but they should be considered as noise data when modeling defects related to road performance predictions, so they must be filtered out. Moreover, the distribution ranges of different road damage indicators vary significantly between each other, which can influence the robustness of the model. Therefore, data normalization operations should be conducted before the model training.

5.5.1.2 Feature Selection

The feature filtering operation aims to remove the auxiliary information from the collected raw dataset. This procedure is conducted manually by professional persons. To ensure the quality of the feature selection, a double check is conducted with the following important aspects being met:

1) Make sure that all the auxiliary information records are completely removed;

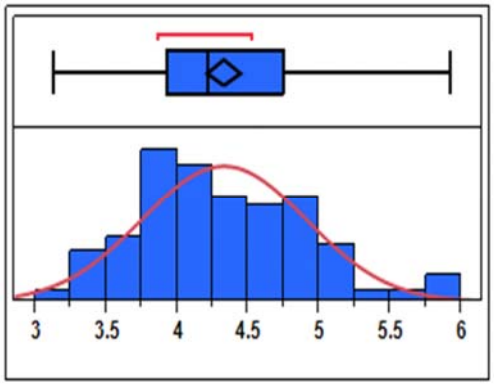
2) Make sure that all the pavement surface distress and performance indices data are fully retained. Since the initial data are aggregated with units of 10 metres, the assumption is that zero values indicate that no defects are observed within that 10-metre road section.

It is noteworthy that the feature selection operations are conducted according to the MTO PM2 specifications.

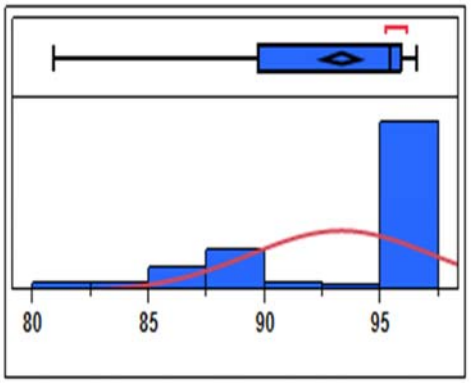
5.5.1.3 Data normalization

Although the pavement surface distresses and the performance index data are obtained after the feature selection procedure, significant imbalance distributions between each feature are observed. This phenomenon can be observed from the data distribution graphs shown in Figure 5.3. Since the whole database includes several records, this may cause a mess if

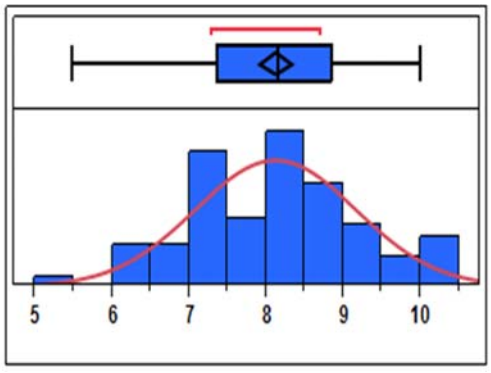
displaying the distribution for every record. Therefore, only pavement deformation condition indices and the overall performance indices are selected for demonstrations. It can be observed that the distribution ranges of different indices vary significantly between each other. This phenomenon is reasonable because each index evaluates the pavement performance from a certain perspective. However, this unbalanced distribution phenomenon may cause the model to suffer from overfitting. Therefore, data normalization operation is one of the methods to avoid the significant diverse distribution caused by model overfitting. The quantities distribution and the statistics summary of the different these indices can be seen in Table 5.3.



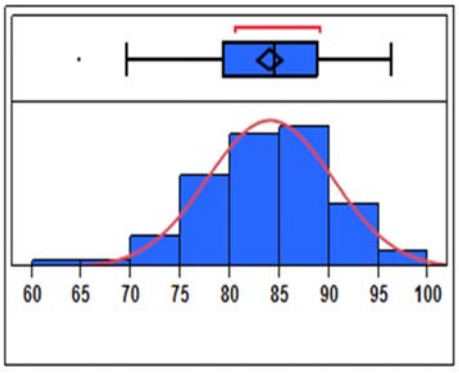
(a) Original rutting distribution



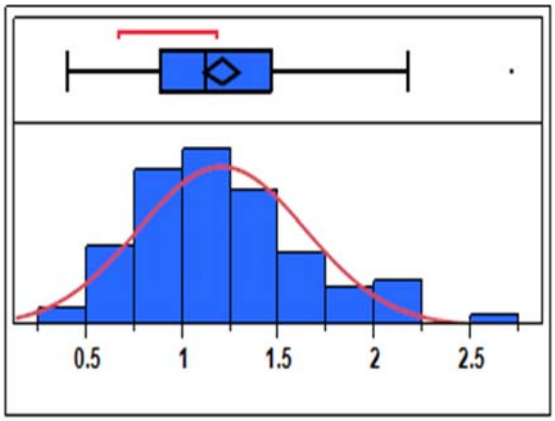
(b)Original DMI distribution



(c)Original RCI distribution



(d) Original PCI distribution



(e) Original IRI distribution

Figure 5.3: Examples of raw data distributions

Table 5.3: Distribution and summary of original data

Index	Quantiles					Summary Statistics				
	Maximum	75%	50%	25%	Minimum	Mean	Std Dev	Std Err Mean	Upper 95% Mean	Lower 95% Mean
Rut	5.93	4.75	4.23	3.93	3.13	4.23	0.58	0.058	4.44	4.21
DMI	96.54	96.54	95.41	89.69	80.86	93.26	3.93	0.39	94.03	92.48
RCI	10	8.85	8.15	7.36	5.49	8.11	1.05	0.10	8.32	7.91
PCI	96.25	88.83	84.52	79.36	64.62	83.98	6.31	0.63	85.22	82.73
IRI	2.71	1.46	1.12	0.89	0.41	1.20	0.43	0.04	1.29	1.11

Date normalization is also called data standardization in some situations. The results are that the distributions of all records are projected between 0 and 1. Meanwhile, no correlations should exist in the standardized dataset. Therefore, this operation can bring two aspects of benefit: (1) The distributions of different indicators can be unified, namely, all the features are converted within the range of $[0, 1]$; (2) Facilitate the efficiency of model training process since the data range disturbance is minimized.

In literature, there are two kinds of data normalization methods being widely used, which are the *MIM-MAX* normalization (Equation 5.5) and the Z-score normalization (Equation5.6).

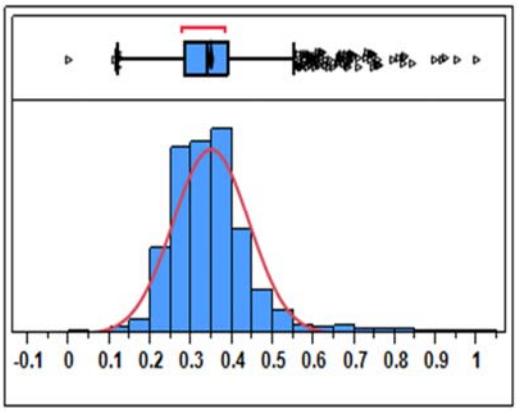
$$x_i^{'} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{5.5}$$

where $x_{\max}$ and $x_{\min}$ denote the maximum and minimum values of the inputs, respectively.

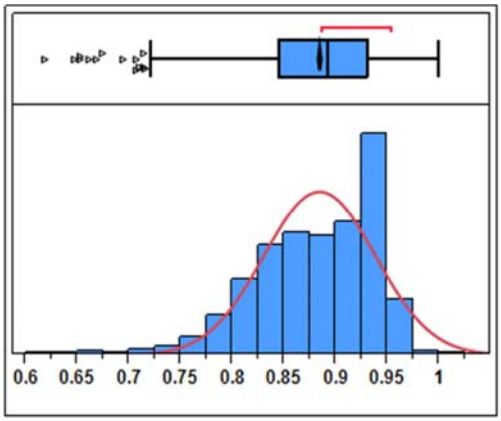
$$Z = \frac{x_i - \mu}{\delta} \tag{5.6}$$

where μ and δ denote the average and the standard deviation (SD) of the dataset.

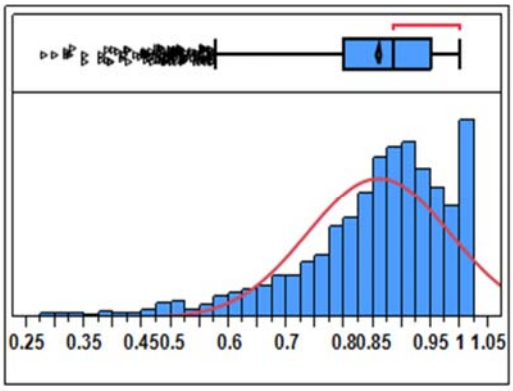
Considering the neurons of the BPNN adopted Sigmoid activation function, which can prevent the output from being saturated due to excessive absolute calculations. Therefore, the *MIN-MAX* normalization (Equation 5.5) method has been chosen for data preprocessing. The distributions of normalized indices are shown in Figure 5.4. It can be seen that after the data normalization operations, those distributions appear more uniform compared to the original ones. Thus, this avoids overfitting avoiding capability and ensures the high generalization capability of the trained model. Detailed distribution and the summary of the different normalized indices are shown in Table 5.4.



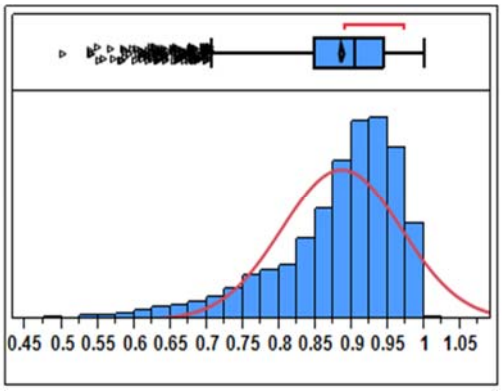
(a) Normalized Rut distribution;



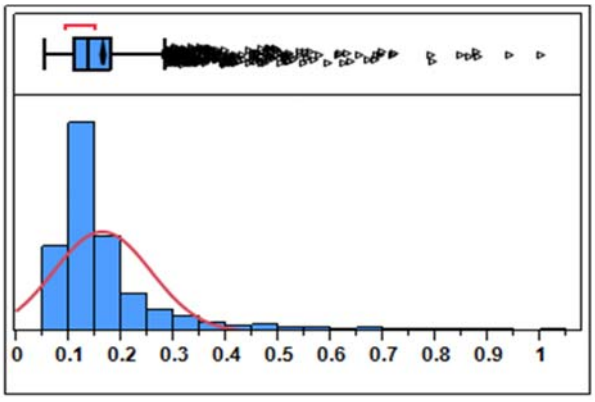
(b) Normalized DMI distribution



(c) Normalized RCI distribution



(d) Normalized PCI distribution



(e) Normalized IRI distributions

Figure 5.4: Examples of normalized data distribution

Table 5.4: Distribution and summary of normalized data

Index	Quantiles					Summary Statistics				
	Maximum	75%	50%	25%	Minimum	Mean	Std Dev	Std Err Mean	Upper 95% Mean	Lower 95% Mean
Rut	1	0.39	0.34	0.28	0	0.35	0.09	0.001	0.35	0.34
DMI	1	0.93	0.89	0.85	0.61	0.88	0.05	0.001	0.86	0.88
RCI	1	0.95	0.88	0.80	0.27	0.86	0.12	0.002	0.86	0.85
PCI	1	0.94	0.91	0.85	0.50	0.89	0.08	0.001	0.89	0.88
IRI	1	0.19	0.14	0.11	0.05	0.16	0.09	0.002	0.165	0.158

5.5.2 Ensemble Learning

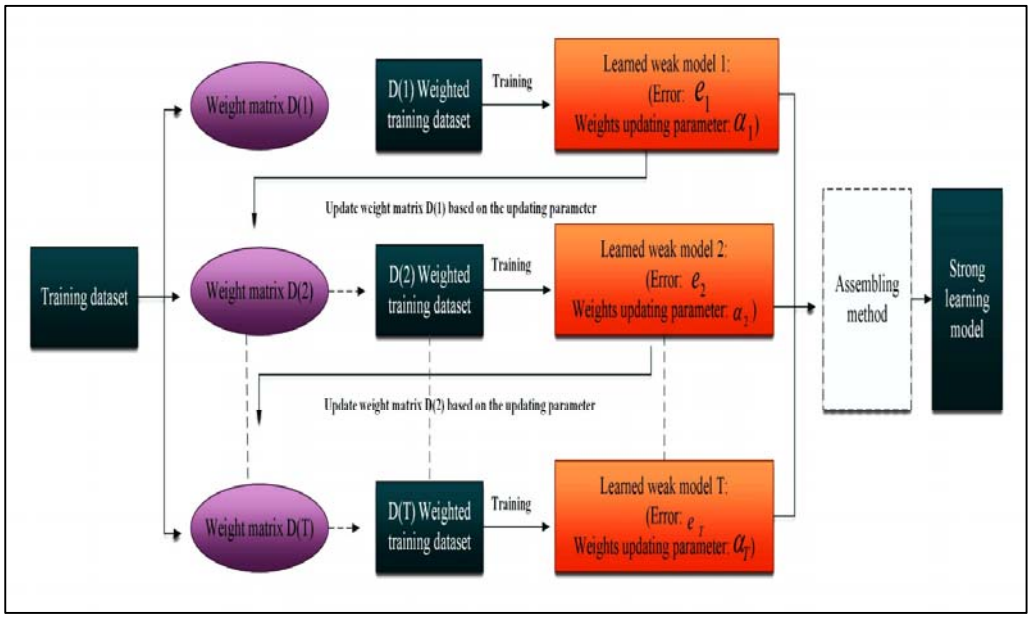
Ensemble Learning, which is also called a boosting method, is characterized by its function mechanism of integrating the advantages of several individual sub-models to generate the final model. Through this intelligent assembling integration, the final learned model tends to be more efficient and robust. The theoretical basis of ensemble learning is the probably approximately correct (PAC) based learnability theory, which is proposed by Kearns and Valiant. This theory is usually being utilized to deal with various classification and regression tasks.

The selection of integration methodology is critical to the overall performance of the final model. That includes how to utilize the advantages of the several sub-models, and

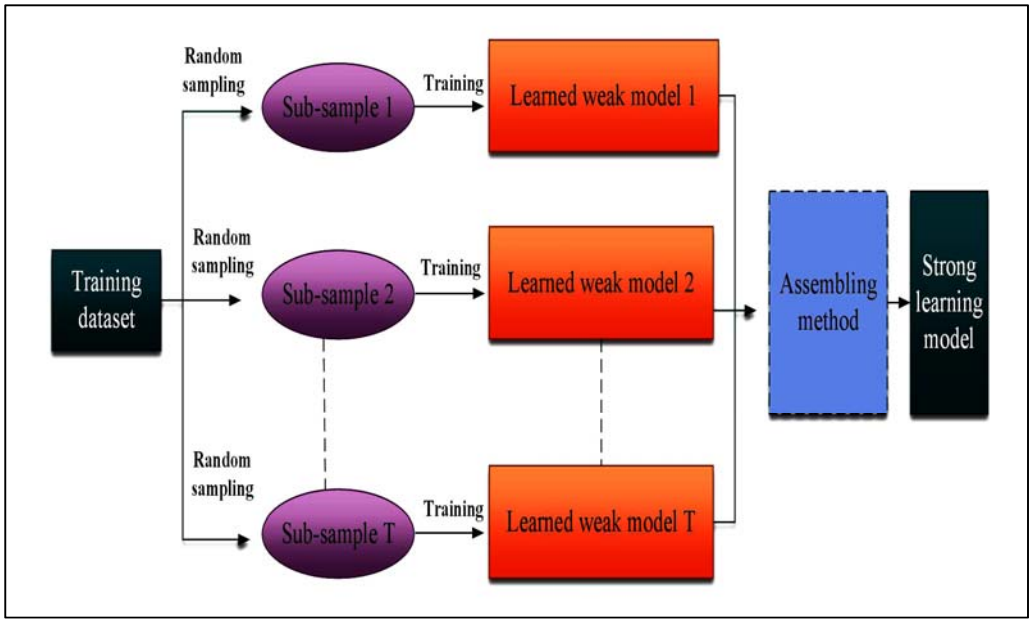
subsequently forming the integrated model with the best performance. In literature, two methods have been developed by researchers for individual model integration, which are:

1）Boosting method (Figure 5.5a). As can be seen from Figure 5.5a, the mechanism of the boosting method is a progressive learning procedure. The individual sub-models are dependent, where the later ones are influenced by the previous sub-models. Firstly, an initial weak sub-model is learned from the original training database. After that, the error rate is used to update the weights of this sub-model. The method used in the step is to increase the weights of samples with higher error rate while decreasing weights of those that have a lower error rate. Then, the updated weights are used to train the second sub-model. The iteration cycle is repeated until the number of required sub-models are obtained. Finally, those sub-models are assembled using the assembling strategy, resulting in the final strong model. The representatives of this kind of models are XGBoost, Adaboost and GBDT methods.

2) Bagging method (Figure 5.5b). As shown in Figure 5.5b, the mechanism of the bagging method is simpler than the boosting approach. Bagging method is based on the bootstrap sampling approach. In the beginning, the initial training dataset is randomly sampled to N times. Thereafter, use these N sub-samples to train the model, obtaining N week sub-models. Finally, the strong model can be obtained by assembling these N weak sub-models. It should be noted that the training data of those sub-samples might be of high similarity since all the random samplings are conducted on the original database. The representative bagging method is the Random Forest (RF) method.



(a) Boosting method



(b) Bagging method

Figure 5.5: Assembling methods

In summary, the boosting assembling learning method can be regarded as a series of re-weighting procedures while the bagging approach includes a series of re-sampling procedures. Since the data re-sampling process may result in high-similarity training dataset, especially when the categorical data are highly imbalanced, and the data size is too small. This fact causes the final trained model to suffer from overfitting. In comparison, the boosting method has less of this aspect of concern since all the weak sub-models are learned from the whole dataset. What's more, the boosting methods focus is on decreasing the error in the model training process, while the final strong model is assembled from the weak sub-models which have weak generalization ability. The assembled model tends to be more accurate, efficient and more robust.

5.5.3 XGBoost Methodology

As has been discussed previously, XGBoost method is a kind of boosting ensemble learning method. The unit weak sub-models used in the XGBoost method are a series of classification and regression trees (CARTs). By assembling the advantages of the individual CART, the final model can achieve the best performance. The basic framework of XGBoost method is shown in Figure 5.6.

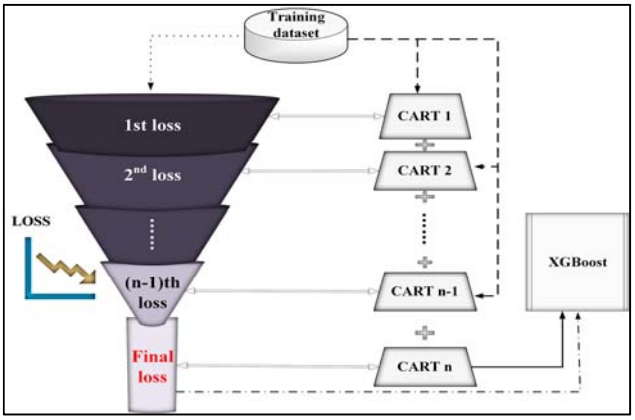


Figure 5.6: Basic framework of XGBoost method

According to the framework of Figure 5.6, the mathematical explanation of the key procedures in XGBoost method can be introduced as follows:

Generally, a regression task can be considered with the purpose of obtaining the model expressed in Equation (5.7) ：

$$\overset{\Lambda}{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{5.7}$$

where K denotes the number of individual CARTs. f_k denotes the kth CART. $\overset{\Lambda}{y}_i$ denotes the predicted value for factor x_i .

The cost function of the model can be calculated by:

$$\text{Obj}(\theta) = \sum_{i=1}^{n} l(y_i, \overset{\Lambda}{y}) + \sum_{k=1}^{K} \Omega(f_k) \tag{5.8}$$

The first part of the cost function: $\sum_{i=1}^{n} l(y_i, \overset{\Lambda}{y})$ denotes the total error of the training data samples, which can be calculated by summing the prediction difference of each input factor. The second part is the normalization term: $\Omega(f_k)$.

Let's consider a single CART, which can be expressed as：

$$f_t(x) = w_{q(x)}, \quad w \in R^T, \ \text{q: } R^d \to \{1, 2, \cdots T\} \tag{5.9}$$

where $w_{q(x)}$ denotes the score of the leaf node $q(x)$. T denotes the total number of leaf nodes in that CART. Then, the computational complexity of this tree can be calculated by：

$$\Omega\ (f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{5.10}$$

where γ and λ are normalization parameters.

Therefore, the final objective function can be obtained as：

$$Obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}^{(t-1)} + f_t(x_i)) + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 + C \tag{5.11}$$

Based on the above calculations, the XGBoost model can be obtained by the integration of individual CARTs, which can be realized by the following steps:

Step 1: Initialize the model (no tree in the model, the prediction result is set to 0): $\hat{y}_i^{(0)} = 0$

Step 2: Add the first CART and train the first model: $\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$

Step 3: Add the next CART to the trained model, obtaining the newly trained model: $\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)$

…

Step *t*: Add the t^{th} CART to the trained model, obtaining the final trained model: $\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$

where f_k denotes the k^{th} tree，and $\hat{y}_i^{(t)}$ denotes the final prediction result. The total loss of the model can be calculated as:

$$\begin{aligned} \text{Obj}(\theta) &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \\ &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + C \end{aligned} \tag{5.12}$$

If using the mean square error (MSE) to calculate the loss, the error of the model can be calculated by Equation (5.13):

$$\mathrm{Obj}(\theta) = \sum_{i=1}^{n} [2(y_i - \hat{y}_i^{(t-1)}) f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + C1 \tag{5.13}$$

5.5.4 Architecture Design and Model Solving

Figure 5.7 shows the framework of XGBoost based pavement performance prediction methodology. Firstly, data normalization is conducted to all the data, resulting in the normalized training dataset. Then, each individual CART is trained by the training data. After that, the weights of the CARTs are updated according to the predetermined threshold. Finally, the XGBoost models are obtained by integrating those individual CARTs. The detailed experiments and the selections of the key parameters are discussed as follows.

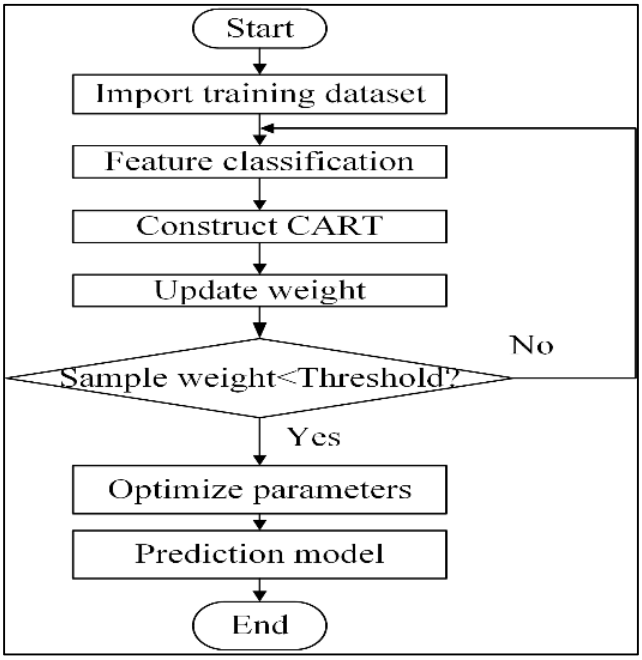


Figure 5.7: Flow chart of XGBoost method

According to Taylor expansion method, the objective function of XGBoost model can be expressed as:

$$\begin{aligned} Obj^{t}(\theta) &= \sum_{i=1}^{n} l(y_i, \overset{\Lambda}{y}^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \\ &\approx \sum_{i=1}^{n} [l(y_i, \overset{\Lambda}{y}_i^{(t-1)}) \partial_{y^{t-1}} l(y_i, \overset{\Lambda}{y}^{(t-1)}) f_t(x_i) + \frac{1}{2} \partial^2{}_{y^{t-1}} l(y_i, \overset{\Lambda}{y}^{(t-1)}) f_t(x_i)^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 + C \end{aligned} \tag{5.14}$$

For simplicity, the constant value $l(y_i, \overset{\Lambda}{y}^{(t-1)})$ is removed, while conducting the replacement of $g_i = \partial_{\overset{\Lambda}{y}^{(t-1)}} l(y_i, \overset{\Lambda}{y}^{(t-1)})$, $h_i = \partial_{\overset{\Lambda}{y}}{}^2 l(y_i, \overset{\Lambda}{y}^{(t-1)})$. Therefore, the objective function becomes:

$$\begin{aligned} Obj^{t}(\theta) &\approx \sum_{i=1}^{n} [l(y_i, \overset{\Lambda}{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 + C \\ &\approx \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \end{aligned} \tag{5.15}$$

In the previous section, it has been introduced that $f_t(x) = w_{q(x)}, w \in R^T, q: R^d \to \{1, 2, \cdots T\}$. At the same time, the item $\sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2]$ can be considered as the sum of all the correlations of the leaf nodes of the t^{th} tree. Therefore, the objective function can be expressed by the leaf nodes of the t^{th} tree, which is:

$$\begin{aligned} Obj^{t}(\theta) &\approx z \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{i=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \end{aligned} \tag{5.16}$$

where T denotes the total number of leaf nodes of the t^{th} tree. $I_j = \{i \mid q(x_i) = j\}$ denotes the sample at the j^{th} leaf node, while w_j denotes the score of the j^{th} node.

Assuming $G_j = \sum_{i \in I} g_i$, $H_j = \sum_{i \in I} h_i$, then the objective function can be revised as:

$$Obj^t(\theta) = \sum_{j=1}^{T} [G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2] + \gamma T \tag{5.17}$$

Calculate the partial derivative with respect to w_j. Then, set the value 0, which means Equation (5.18) can be obtained:

$$G_j + (H_j + \lambda) w_j = 0 \tag{5.18}$$

Thus, the scores can be calculated as：

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{5.19}$$

The objective function becomes：

$$Obj^* = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{5.20}$$

5.5.5 Modeling Details

5.5.5.1 XGBoost parameter setting

Several parameters should be set when training the model. Those parameters with their corresponding descriptions are listed in Table 5.5. The selection of some important parameters is discussed in the following part.

Table 5.5: Parameter description

Parameter	***Description***
learning_rate(lr)	The decay value for updating the weights
n_estimators(I)	The best iterations
max_depth(D)	The maximum depth, which is higher than 1. The higher this value, the better the model for the training dataset.
min_child_weight(W)	The sum of the weights of leaf nodes which have the minimum sample size.
subsample(S)	The proportion of the training set to the total dataset.
colsample_bytree (CS)	Column sampling parameter, which should be set to the range of 0.5-1.0.
Gamma (g)	Parameter used to control whether to pruning
reg_alpha (α)	L1 normalization parameter. The higher this value, the better the model for avoiding overfitting.
reg_lambda (λ)	L2 normalization parameter. The higher this value, the better the model for avoiding overfitting.
scores (s)	The accuracy of the model

5.5.5.2 XGBoost model training

R^2 (in the following part of this chapter, the term "scores" refer to the R^2 values) is used as the indicator to evaluate the performance changing of the model with the increasing number of iterations. Herewith, the prediction of DMI is selected as an example to introduce the model training operation. The best suitable iterations should be decided in the first place. Two steps of best iteration selections are conducted to ensure the performance of the model, which are rough selection and the detailed (best) selection procedures. Figure 5.8 shows the rough selection process, in which the scores are observed with an observation interval of 100 iterations. It can be found that the highest score can be obtained between 1000 and 1200 iteration of weights updating.

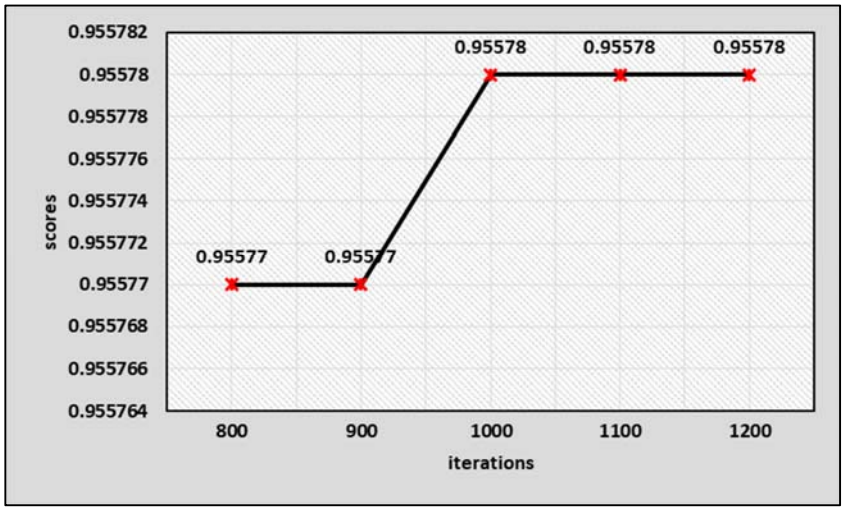


Figure 5.8: Rough iteration range selection

From the curve shown in Figure 5.8, it can be found that the accuracy of the model can be achieved around 0.95578 between 1000 and 1200 iterations. However, the best number of iterations is still not clear, which needs to be further determined. Therefore, based on the previous conclusion the observation interval is changed to 50 iterations. The scores are shown in Figure 5.9, from which can be found that the best score is obtained when the model is trained at 1050 iterations. Hence, 1050 is used as the final iteration. Meanwhile, the best predicting accuracy can be obtained as 0.95579.

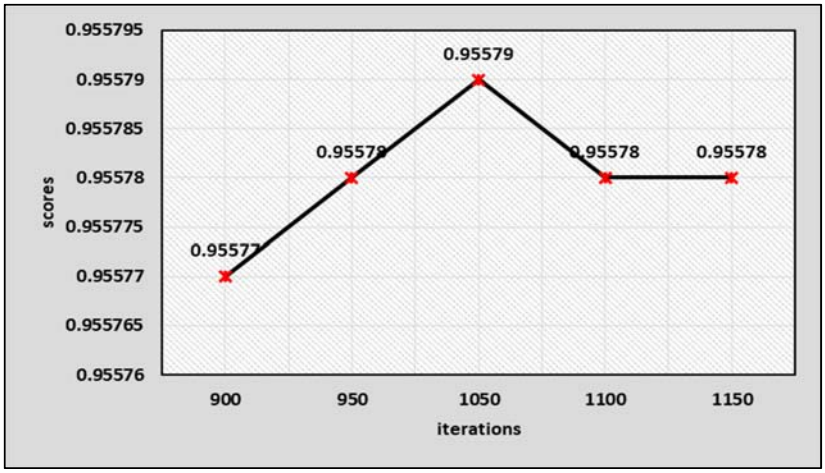


Figure 5.9: Best iteration selection

Similarly, the selection of other parameters: *D*, *W*, and lr, is conducted in the same method. Since *D* and *W* are closely related to each other, these two parameters are decided simultaneously. Figure 5.10 shows the accuracy changing with different pairs of depth (D) and weight values (W). It can be found that the highest score is achieved when the depth of the tree equals to 3 and the minimum child weight equals to 1.

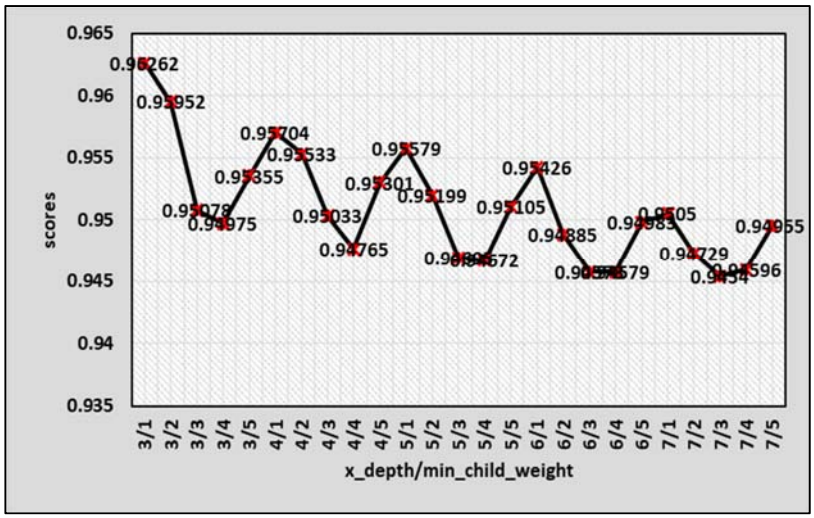


Figure 5.10: Selection of parameter D and W

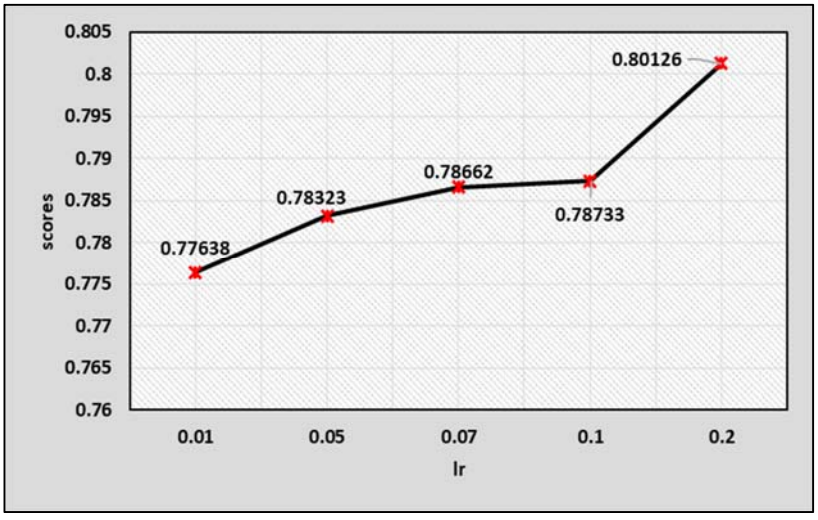


Figure 5.11: Scores changing with different lr

Figure 5.11 shows the scores of the model with different learning rates. It can be observed that at the beginning of model training, the scores increase with the increase of learning rate. With an lr close to 0.2, the scores tend to decrease. Therefore, the optimum learning rate has been set to 0.2.

Other parameters are determined in similar ways. Finally, the optimum parameters are shown in Table 5.6:

Table 5.6: XGBoost parameters

Parameter	**Value**	**Parameter**	**Value**
learning_rate	0.2	seed	0
n_estimators	1050	subsample	0.8
max_depth	3	colsample_bytree	0.8
min_child_weight	1	gamma	0.1
reg_alpha	0.05	reg_lambda	3

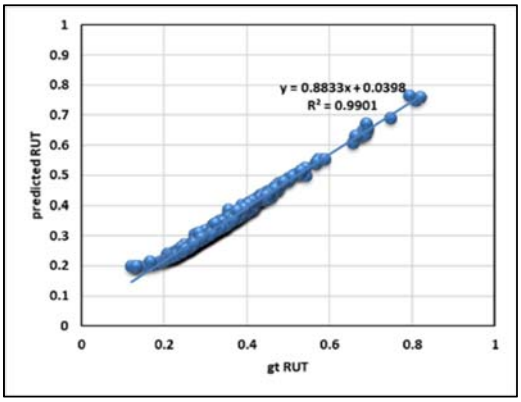
5.5.5.3 Results and Analysis

Using the method discussed in the previous part, the XGBoost models for the predictions of RUT, IRI, DMI, RCI, and PCI are trained. Finally, the selected parameters with the corresponding prediction accuracies are shown in Table 5.7. It can be seen that the prediction accuracies for all these indices are higher than 0.8, which proves the effectiveness of XGBoost method.

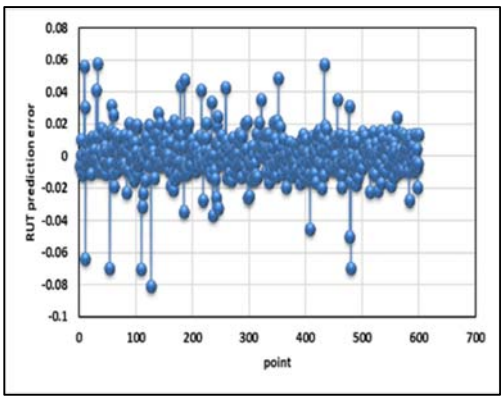
Table 5.7: XGBoost parameters selection and performance

Method	lr	Depth	Weight	Alpha	Lambda	Iteration	Accuracy
RUT	0.05	3	1	0.05	0.05	600	0.99
IRI	0.01	3	3	0.05	0.05	1000	0.84
DMI	0.2	3	1	0.05	3	1050	0.85
RCI	0.07	4	3	0.05	0.05	900	0.99
PCI	0.2	3	1	0.05	0.05	1000	0.97

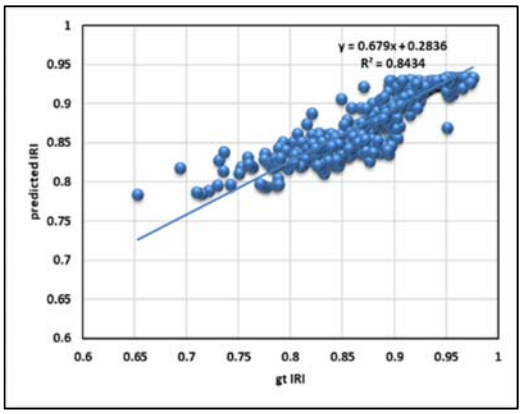
In Figure 5.12, the model prediction accuracy and error curves are plotted. The fitted line between the predicted values and the corresponding ground truth (a1, b1, c1, d1, e1) are plotted. It can be found that the average R^2 of these five indices are 0.9568. Moreover, the residuals (a2, b2, c2, d2, e2) indicate the prediction errors for these indices are all within ± 0.15. These results proved the satisfactory performance of XGBoost models developed in this part.

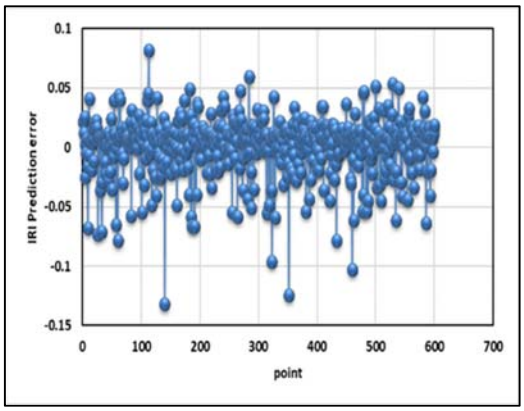


(a1) Fitting results of RUT

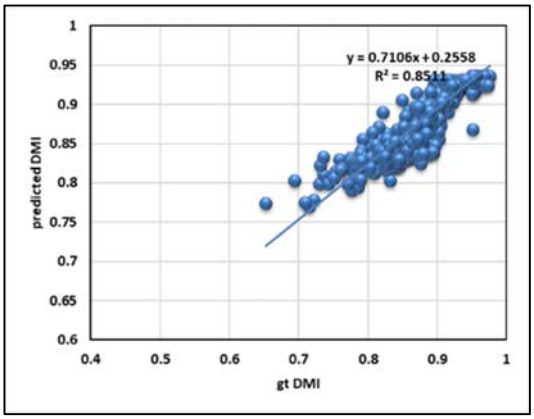


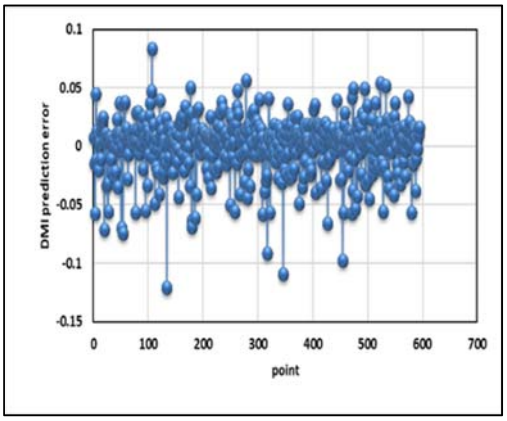
(a2) Errors of RUT prediction



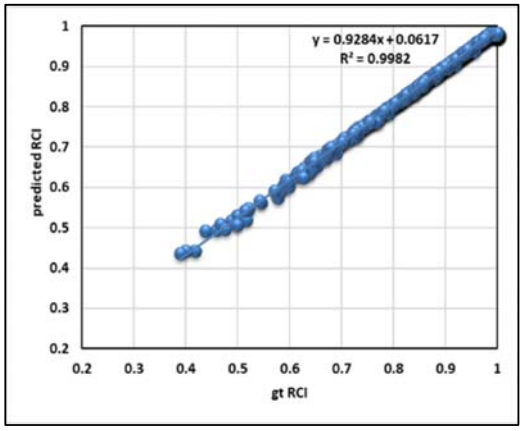


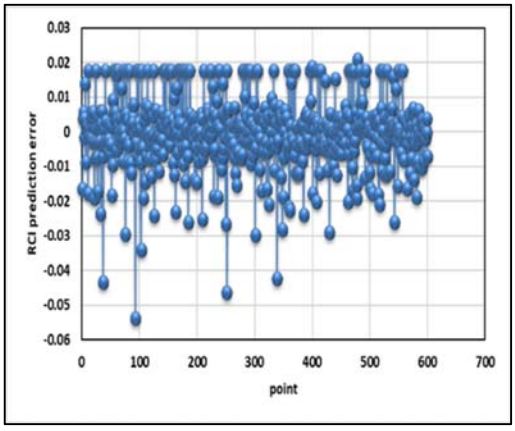
(b1) Fitting results of IRI **(b2) Errors of IRI prediction**



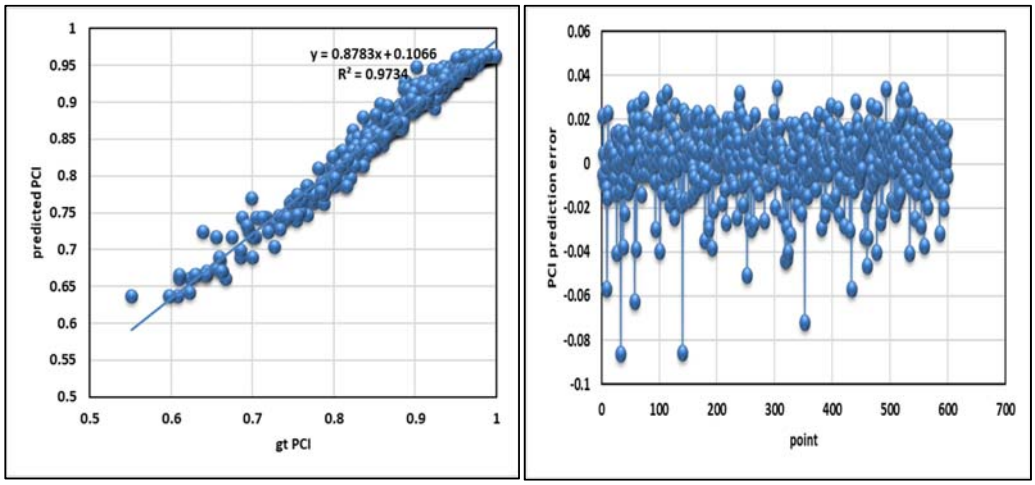


(c1) Fitting results of DMI **(c2) Errors of DMI prediction**





(d1) Fitting results of RCI **(d2) Errors of RCI prediction**



(e1) Fitting results of PCI **(e2) Errors of PCI prediction**

Figure 5.12: XGBoost method results

The error calculations are conducted for each prediction result. Mean absolute error (MAE) is selected as the measurement. The calculated MAEs are:

$$MAE_{PCI} = 0.011608$$

$$MAE_{RCI} = 0.007064$$

$$MAE_{DMI} = 0.015782$$

$$MAE_{\mathrm{RUT}} = 0.009070$$

$$MAE_{\mathrm{IRI}} = 0.016611$$

It can be observed from the above results that the MAEs are all around 0.01. Hence, those XGBoost models can be regarded as suitable for the performance indices prediction.

5.5.5.4 Model comparative analysis

To verify the effectiveness of the proposed method, performance of the proposed XGBoost model is compared with the widely used BPNN methods. The results are shown in Table 5.6. Based on the R^2 and MAE values shown in the table, it can be concluded that XGBoost method outperforms GBDT in most conditions. The reason is that XGBoost ensembles several CARTs by integrating the advantages of each individual regression tree to generate the final model. In comparison, BPNN uses a single multi-layer perception method to achieve the final prediction purpose. The most commonly seen problem of BPNN is overfitting, which is observed from the less satisfactory performance of the model in the test dataset. However, both methods can achieve satisfying results for the prediction of RUT (R^2=0.987, 0.978) and DMI (R^2 =0.827). Specifically, characterized by the simplicity and convenience, BPNN can be considered as slightly outperforming XGBoost for the prediction of RUT, whereas XGBoost has the best performance for the predictions of all the other network level indices.

Table 5.8: Performance comparison

Index	R^2		MAE	
	GBDT	**XGBoost**	**GBDT**	**XGBoost**
IRI	0.753	**0.811**	0.031	**0.017**
PCI	0.949	**0.963**	0.014	**0.011**
RUT	**0.987**	**0.978**	**0.003**	**0.009**
RCI	0.959	**0.993**	0.017	**0.007**
DMI	**0.827**	**0.827**	0.016	**0.016**

5.6. Chapter Summary

This chapter used the pavement performance data collected by ARAN 9000. The raw data include 2D image-based pavement defects indicators, pavement roughness condition indices, the overall road performance indices, and other auxiliary data.

Based on the above dataset, this chapter developed image indicator-based roughness condition prediction and the overall performance prediction models. The most state-of-the-art machine learning based prediction method, XGBoost model, has been used for modeling. Meanwhile, the performances of these models are compared with GBDT methods, which indicate that XGBoost method performs better then GBDT in most conditions.

From the result of this chapter, it can be concluded that 2D image analysis-based roughness predictions and the overall performance predictions are highly promising. This is a direct benefit from recent advances in machine learning technology. Therefore, pavement management departments are encouraged to take advantage of this finding, i.e. integrating those methodologies into current 2D pavement management systems, thereby improving PMS's performance as well as reducing the overall pavement management budget allocation.

CHAPTER 6. Examining Dominant Pavement Surface Distresses Indicators Influencing Overall KPIs

6.1. Introduction

The overall pavement Key Performance Indicators (KPIs) are important references being used in pavement management systems for maintenance decision making. As important components of the functions for calculating the most KPIs, pavement surface distresses exist in diverse types with various severity levels. Pavement surface distresses are the visible manifestations resulting from effects of the traffic loads, pavement environmental influences and pavement aging. The impacts of different kinds of pavement surface distresses on pavement overall performance usually vary significantly. That is one of the reasons why the coefficients of the functions for calculating different KPIs should be calibrated when being used in specific regions. Moreover, how to select the best suitable pavement maintenance operations has always been a challenge of effective pavement management.

The automated and semi-automated pavement surface distress detection approaches have undergone extensive research in recent years. The most straightforward ways are to detect pavement surface distresses from pavement images collected by imaginary systems. Digital image processing technology has been widely utilized in the image-based pavement surface distress detection and characterization after several years of development. However, no guidelines can be found in current literature providing straightforward instruction on which kind of surface distress has the most significant influence on the overall pavement performance and which comes the next. Therefore, it is challenging for effective pavement maintenance decision making.

Current pavement maintenance decision making is based on deterioration of performance and cost analysis. Life Cycle Cost Analysis (LCCA) is the most widely used maintenance scheduling mathematical approach. A performance prediction curve should be developed initially, which is usually a deterioration curve showing the changing of performance indicators with pavement aging. The performance indicators are selected as PCI, IRI, etc., for most systems. Then, the cost-effective M&R treatments are scheduled according to budget allocation. During this procedure, different alternatives should be assessed to choose the best cost-effective one. The selection of the best suitable M&R alternative is always a challenging task. There are two important reasons: (1) Historical data should be used to develop the performance deterioration curve. However, it is not easy to obtain reliable and complete pavement historical data for most agencies; (2) The M&R treatments should be planned in a cost-effective way, which means that different cost should be considered. However, not all of the costs can be quantified, such as user delay cost, vehicle damage cost and the environment impact cost. Those costs can be considered as hidden costs, but they are important part of the transportation cost.

Considering the above problems, this research aims to examine the dominant pavement surface distresses. Firstly, neural network (NN)-based pavement performance prediction models are developed. Various kinds of pavement distress parameters are used as the inputs, while the overall performance indices are used as outputs for each individual model. Then, the mean impact values (MIVs) for different distress parameters are calculated, which are considered as indicators for evaluating the influence condition of each distress types. On this basis, high contributed distresses are regarded as the dominant distresses.

6.2. Introduction of Neural Network

NN is proposed as an algorithmic mathematical model which can conduct distributed parallel computing. A series of interconnecting internal nodes of a NN act as functional

units to achieve multi-level information extraction. Each neural node is connected with a weight value, which is changeable in the model training process. In the beginning, the weights can be initialized by a random number or zero value. Then, the model is trained by updating the values of the weights with the purpose of minimizing the difference between the model outputs and the ground truth. A suitable loss function that represents those differences should be defined to facilitate the model training procedure. The calculation principle of a basic neuron is shown in Figure 6.1. The actual architecture of a complete NN model is usually formed by a non-linear combination of a large amount of basic calculation units. Observing from the demonstration of a unit neuron, it can be found that five components are included, which are the input part, the weight values, the bias value, the summation, and the activation function. Through this end-to-end processing, non-linear model for predicting the output values based on the inputs can be obtained.

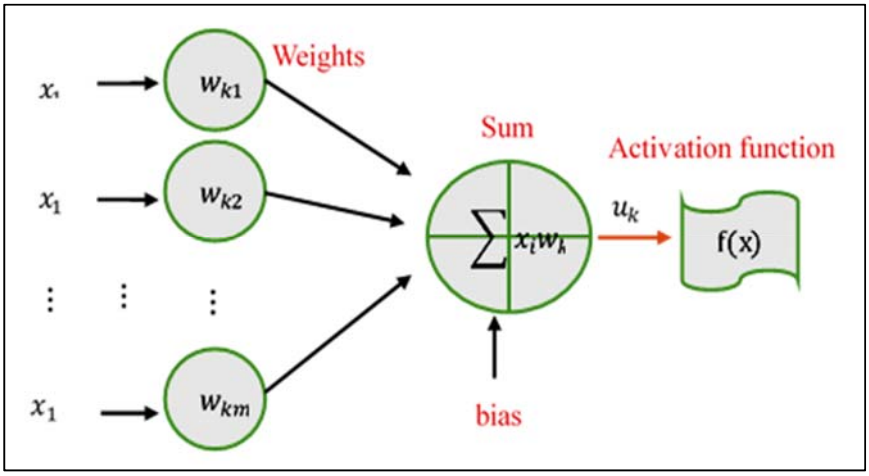


Figure 6.1：Demonstration of a neural model

Mathematically, the model of Figure 6.1 can be defined by simple function expressed in Equation (6.1)

$$y_k = f(\sum_{j=1}^{n} W_{ij}\chi_j - b_k) \tag{6.1}$$

where $\chi_1, \chi_2, \cdots, \chi_n$ is the input data vector. W_{k1}, W_{k2}, W_{kn} denote the values of the weights between the input values and the k^{th} calculating neural node. b_k denotes the corresponding bias. f is the activation function. y_k denotes the predicted output value.

6.2.1 Back Propagation Neural Network (BPNN)

6.2.1.1 The architecture of BPNN

The name of BPNN comes from the model training mechanism used in this kind of neural network model. Literally, the difference between the output value and the ground truth, which is called error, should be highest at the beginning of model training. Then, the error is to be minimized from the back-propagation process. The basic structure of a BPNN includes the input layer, the hidden layers, and the output layer. The simplest three-layer BPNN with only one hidden layer can be seen in Figure 6.2.

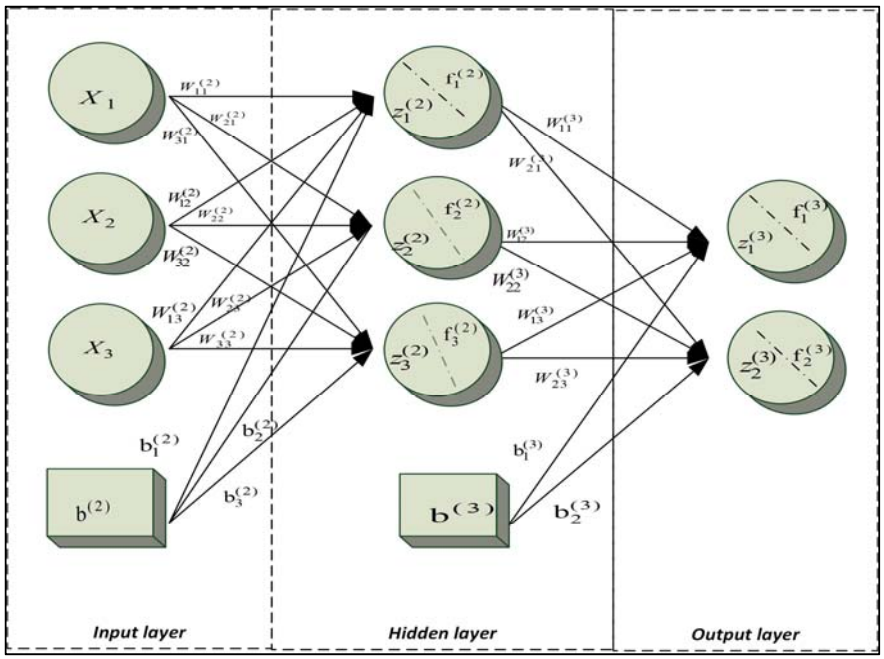


Figure 6.2: Three-layer BPNN

6.2.1.2 Model solution procedure of BP method

The concept of the BP method is simple as that mentioned previously. Two basic procedures are included in the process of BP method, which are (1) Conduct dot product between each weight value and the corresponding input values of each node and calculate the activated result for every neural node from the input layer to the output. (2) Calculate the errors between the predicted value and the ground truth. Then, update the values of the weights while back propagating the error. As an example, the updating of the weights based on error back propagating mechanism of the three-layer network, which is shown in Figure 6.2 can be introduced as follows:

Let the input data vector be: $x=(x_1,x_2,x_3)^T$. The nodes of the hidden layer and the output layer are supposed to be 3 and 2, respectively.

Stage I: Forward propagation

The first stage of BP method is the forward propagation, which calculates the output values by two steps: (1) Conduct dot product between the input values and the weights assigned to then, then summate all the weighted input values for the corresponding node. (2) Use the summation results as the parameter of the activation function to conduct the activation operation, obtaining in the output results. In this procedure, two kinds of conditions should be considered according to the place of the neuron nodes, which are:

- If the nodes are in the hidden layer, the calculations should be conducted by Equations (6.2) and (6.3)

$$\begin{aligned} z_1^{(2)} &= w_{11}^{(2)}x_1 + w_{12}^{(2)}x_2 + w_{13}^{(2)}x_3 + b_1^{(2)} \\ z_2^{(2)} &= w_{21}^{(2)}x_1 + w_{22}^{(2)}x_2 + w_{23}^{(2)}x_3 + b_2^{(2)} \\ z_3^{(2)} &= w_{31}^{(2)}x_1 + w_{32}^{(2)}x_2 + w_{33}^{(2)}x_3 + b_2^{(2)} \end{aligned} \tag{6.2}$$

$$\begin{aligned} a_1^{(2)} &= f(z_1^{(2)}) \\ a_2^{(2)} &= f(z_2^{(2)}) \\ a_3^{(2)} &= f(z_3^{(2)}) \end{aligned} \tag{6.3}$$

- If the nodes are in the output layer, the calculations should be based on Equations (6.4) and (6.5)

$$\begin{aligned} z_1^{(3)} &= w_{11}^{(3)} a_1^{(2)} + w_{12}^{(3)} a_2^{(2)} + w_{13}^{(3)} a_3^{(2)} + b_1^{(3)} \\ z_2^{(3)} &= w_{21}^{(3)} a_1^{(2)} + w_{22}^{(3)} a_2^{(2)} + w_{23}^{(3)} a_3^{(2)} + b_2^{(3)} \end{aligned} \tag{6.4}$$

$$\begin{aligned} a_1^{(3)} &= f(z_1^{(3)}) \\ a_2^{(3)} &= f(z_2^{(3)}) \end{aligned} \tag{6.5}$$

In general, the output of the n^{th}layer in the forward propagation process can be expressed by Equation (6.6):

$$\begin{aligned} z^n &= W^{(n)} a^{(n-1)} + b^n \\ a^{(n)} &= f(z^n) \end{aligned} \tag{6.6}$$

where $a_i^{(j)}$ denotes the output of a specific neural node.

Stage II: Back propagation

The total error of the neural network should be calculated at first. Then, based on the calculated error value, the weights can be updated with the purpose of minimize the error gradually. The error calculation can be varied based on the specific models defined by the developers. The most straightforward innovation is to calculate the difference between the model output and the ground truth. For this consideration, Equation (6.7) is the most widely used individual neuron error function in literature, which is also utilized by this research.

$$E_{(i)} = E(w_i, b_i) = \frac{1}{2}\left\| y^{(i)} - o^{(i)} \right\| = \frac{1}{2}\sum_{j=1}^{n} (y_j - o_j)^2 \tag{6.7}$$

where o_j denotes the prediction. y_j denotes the corresponding ground truth; n is the total number of inputs for that neuron.

Based on the calculated individual error of each node, the total error of all the neurons can be obtained by Equation (6.8)

$$E_T = \frac{1}{N}\sum_{i=1}^{N} E(w_i, b_i) \tag{6.8}$$

As has been mentioned, the weights of the neurons should be updated with the purpose of minimizing the total error, thereby guiding the predictions close to the expected results. Therefore, a specific method should be developed to fulfil this purpose. The gradient descent (GD) method has been proved to be extremely effective. Firstly, the gradient value that represents the most drastic difference should be calculated. Then, the corresponding weight can be updated by subtracting the difference from the original assigned weight value. Iterating this procedure until the error arrives at the acceptable level. Then, the final weight matrix is used as the optimal model, which can be used for testing of the new dataset.

The values of weights and biases of the network model can be updated based on the GD method using Equations (6.9) and (6.10):

$$W^{(l)} = W^{(l)} - \mu\frac{\partial E_T}{\partial W^{(l)}} = W^{(l)} - \frac{\mu}{N}\sum_{i=1}^{N}\frac{\partial E_{(i)}}{\partial W^{(l)}} \tag{6.9}$$

$$b^{(l)} = b^{(l)} - \mu\frac{\partial E_T}{\partial b^{(l)}} = b^{(l)} - \frac{\mu}{N}\sum_{i=1}^{N}\frac{\partial E_{(i)}}{\partial b^{(l)}} \tag{6.10}$$

Observing the calculations in the above functions, the key task is to calculate two values, which are $\frac{\partial E_{(i)}}{\partial W^{(l)}}$ and $\frac{\partial E_{(i)}}{\partial b^{(l)}}$. The calculation method of these two factors for the output layer and the hidden layer are different from each other. Therefore, they are

explained separately.

(1) Output layer weights update

Following the calculations in the previous sections, the error between the output layer and the hidden layer of the architecture shown in Figure 6.2 can be revised as:

$$\begin{aligned} E&=\frac{1}{2}\|\mathrm{y}-o\|=\frac{1}{2}\left\|\mathrm{y}-a^{(3)}\right\|=\frac{1}{2}((y_1-a_1^{(3)})^2+(y_2-a_2^{(3)})^2)=\frac{1}{2}((y_1-f(z_1^{(3)}))^2+(y_2-f(z_2^{(3)}))^2) \\ &=\frac{1}{2}((y_1-f(w_{11}^{(3)}a_1^{(2)}+w_{12}^{(3)}a_2^{(2)}+w_{13}^{(3)}a_3^{(2)}+b_1^{(3)}))^2+(y_2-f(w_{21}^{(3)}a_1^{(2)}+w_{22}^{(3)}a_2^{(2)}+w_{23}^{(3)}a_3^{(2)}+b_2^{(3)}))^2) \end{aligned} \tag{6.11}$$

Considering a single neural node in the output layer, the derivative of the error function with respect to its weight value can be obtained by:

$$\frac{\partial E}{\partial w_{11}^{(3)}}=\frac{1}{2}\bullet 2(\mathrm{y}_1-\mathrm{a}_1^{(3)})\ \ (-\frac{\partial a_1^{(3)}}{\partial w_{11}^{(3)}})=-(\mathrm{y}_1-\mathrm{a}_1^{(3)})\,\mathrm{f}'(z_1^{(3)})\frac{\partial z_1^{(3)}}{\partial w_{11}^{(3)}}=-(y_1-a_1^{(3)})f'(z_1^{(3)})a_1^{(2)} \tag{6.12}$$

Denote $\frac{\partial E}{\partial z_i^{(l)}}=\delta_i^{(l)}$, then Equation (6.12) can be simplified as:

$$\frac{\partial E}{\partial w_{11}^{(3)}}=\frac{\partial E}{\partial z_1^{(3)}}\frac{\partial z_1^{(3)}}{\partial w_{11}^{(3)}}=\delta_1^{(3)}a_1^{(2)} \tag{6.13}$$

where $\delta_1^{(3)}=\frac{\partial E}{\partial z_1^{(3)}}=\frac{\partial E}{\partial a_1^{(3)}}\frac{\partial a_1^{(3)}}{\partial z_1^{(3)}}=-(y_1-a_1^{(3)})f'(z_1^{(3)})$

Similarly, the values of the weights of the other neural nodes in the output layer can be calculated as:

$$\begin{aligned} &\frac{\partial E}{\partial w_{12}^{(3)}}=\delta_1^{(3)}a_2^{(2)} \quad \frac{\partial E}{\partial w_{13}^{(3)}}=\delta_1^{(3)}a_3^{(2)} \\ &\frac{\partial E}{\partial w_{22}^{(3)}}=\delta_2^{(3)}a_2^{(2)} \quad \frac{\partial E}{\partial w_{23}^{(3)}}=\delta_2^{(3)}a_3^{(2)} \end{aligned} \tag{6.14}$$

where $\delta_2^{(3)} = -(y_2 - a_2^{(3)}) f'(z_2^{(3)})$.

Generally, if the neural network has L layers in total, the corresponding values should be calculated in a more general way as expressed in Equation (6.15):

$$\begin{aligned} &\delta_i^{(L)} = -(y_i - a_i^{(L)}) f'(z_i^{(L)}); (1 \le i \le n_L) \\ &\frac{\partial E}{\partial w_{ij}^{(L)}} = \delta_i^{(L)} a_j^{(L-1)}; (1 \le i \le n_L, 1 \le j \le n_{L-1}) \end{aligned} \tag{6.15}$$

The matrix expression Equation (6.15) is:

$$\begin{aligned} &\delta^{(L)} = -(y - a^{(L)}) \odot f'(z^{(L)}) \\ &\nabla_{W^{(L)}} E = \delta^{(L)} (a^{(L-1)})^T \end{aligned} \tag{6.16}$$

where $\odot$ denotes the Hadamard product calculation.

(2) Hidden layer weights update

The weights updating of the hidden layer is slightly different from the output layer because there usually have more than one hidden layer in the network. For simplicity, the updating methodology for the single hidden layer model is explained first. In this circumstance, Equation (6.17) is the updating function.

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \frac{\partial E}{\partial z_1^{(l)}} \frac{\partial z_i^{(l)}}{\partial w_{ij}^{(l)}} = \delta_i^{(l)} a_j^{(l-1)} \tag{6.17}$$

where $\delta_i^{(l)}, 2 \le l \le L-1$, which can be calculated as:

$$\delta_i^{(l)} = \frac{\partial E}{\partial z_i^{(l)}} = \sum_{j=1}^{n_l+1} \frac{\partial E}{\partial z_j^{(l+1)}} \frac{\partial z_j^{(l+1)}}{\partial z_i^{(l)}} = \sum_{j=1}^{n_l+1} \delta_i^{(l+1)} \frac{\partial z_j^{(l+1)}}{\partial z_i^{(l)}} \tag{6.18}$$

If there is a $l-layer$ neural network model including the fully connected layer. The

total error E is expended in the back-propagation process.

Assume that the error in the $(l+1)-th$ layer is denoted as $z^{(l+1)}$. All the neurons of the $(l+1)-th$ layer is connected with those of the $l-th$ fully connected layer. Therefore, the term $z_i^{(l)}$ appears n_{l+1} times, and every appearance of the term $z_i^{(l)}$ corresponds to a certain value $z_i^{(l+1)}, 1 \le j \le n_{l+1}$. Therefore, the derivative value can be calculated as:

$$\frac{\partial E}{\partial z_i^{(l)}} = \frac{\partial E}{\partial z_1^{(l+1)}} \frac{\partial z_1^{(l+1)}}{\partial z_i^{(l)}} + \frac{\partial E}{\partial z_2^{(l+1)}} \frac{\partial z_2^{(l+1)}}{\partial z_i^{(l)}} + \ldots + \frac{\partial E}{\partial z_{n_{l+1}}^{(l+1)}} \frac{\partial z_{n_{l+1}}^{(l+1)}}{\partial z_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \frac{\partial E}{\partial z_j^{(l+1)}} \frac{\partial z_j^{(l+1)}}{\partial z_j^{(l)}} \tag{6.19}$$

Based on the above synthesis function, the $\delta^{(l+1)}$ of the $(l+1)^{th}$ layer can be calculated based on the result of the previous layer, recursively conducting this operation from the output layer to the first layer. Finally, the weights of every neuron can be updated.

(3) Biases update

The updating of biases is simpler than the updating of the weights. The biases of all the layers can be updated using the same method, which can be expressed by Equation (6.20)

$$\frac{\partial E}{\partial b_i^{(l)}} = \frac{\partial E}{\partial z_1^{(l)}} \frac{\partial z_i^{(l)}}{b_i^{(l)}} = \delta_i^{(l)} \tag{6.20}$$

Figure 6.3 shows the flowchart of the BP model training method.

6.2.1.3 Mean Impact Value (MIV) factor importance evaluation

The MIV is proposed to measure the contribution of each input factor with respect to the output. The detailed calculation of MIV is explained in this section:

Assume the developed NN model has P inputs, which represent P factors. The input variables are all observed m times. Then, the input space can be obtained, which is denoted as $X = [x_1, x_2, \ldots, x_m]$. Meanwhile, the corresponding results with respect to

each sample point can be denoted as $Y = [y_1, y_2, \ldots, y_m]$. Therefore, the training dataset can be obtained as $\{X, Y\}$, which is constructed by the input matrix and the output vector. On this basis, the NN model can be trained to obtain the final model.

Then, the original independent input parameters should be revised in two aspects, which are: (1) add 10% to each value based on their original values; (2) minus 10% to each value based on their original values. As a result, two new variable spaces can be obtained as:

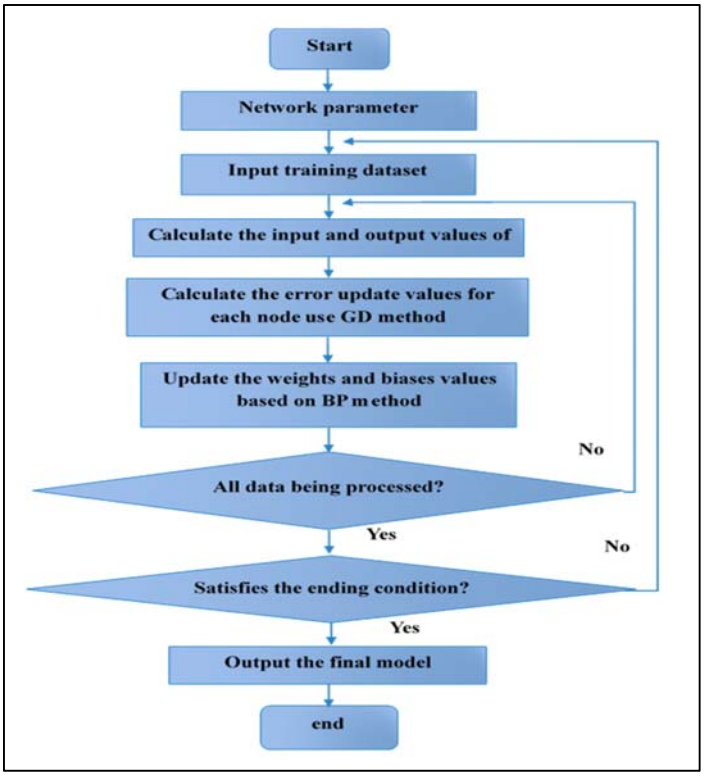


Figure 6.3: Flow chart of BP algorithm

$$X_i^{(1)} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i1}(1+10\%) & x_{i2}(1+10\%) & \cdots & x_{im}(1+10\%) \\ \vdots & \vdots & \vdots & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pm} \end{bmatrix}$$

$$X_i^{(2)} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i1}(1\text{-}10\%) & x_{i2}(1\text{-}10\%) & \cdots & x_{im}(1\text{-}10\%) \\ \vdots & \vdots & \vdots & \vdots \\ x_{p1} & x_{p2} & \cdots & x_{pm} \end{bmatrix} \quad (6.21)$$

Then, the newly constructed datasets are used as the input of the NN model separately. At the output side, two vectors can be obtained as:

$$\begin{aligned} Y_i^{(1)} &= [\, y_{i_1}^{(1)}, y_{i_2}^{(1)}, y_{i_3}^{(1)}, \ldots, y_{i_m}^{(1)} \,] \\ Y_i^{(2)} &= [\, y_{i_1}^{(2)}, y_{i_2}^{(2)}, y_{i_3}^{(2)}, \ldots, y_{i_m}^{(2)} \,] \end{aligned} \quad (6.22)$$

By calculating the difference between the above two vectors, the impact value vector can be obtained as

$$I_{MIV,i} = \sum_{j=}^{m} \frac{I_{V,i}(j)}{m}, i = 1, 2, 3, \ldots, p \quad (6.23)$$

where $I_{MIV,i}$ denotes the MIV of the i^{th} input factor on the output. The absolute value represents the relative impact condition, and the positive/negative of the result shows the impact direction of the i^{th} factor.

For each input factor, MIV can be calculated by the above method. Then, those MIVs are employed to assess the importance condition of the input factors.

6.3. BPNN based Pavement Performance Prediction and MIV Analysis

The experiments are conducted using pavement performance data collected by ARAN 9000 automatic pavement inspection vehicle. The raw data obtained from the storage of the data

collection device mainly include two categories: (1) image indicators of pavement surface distress, and (2) pavement depth measurements and the overall performance indices.

In the beginning, all the data should be normalized to ensure the robustness of the trained model. On this basis, the image indicators of pavement surface distresses, such as the crack length, width, area, and percent, etc., are used as inputs. All those indicators can be measured using data image processing technologies from the 2D pavement images. The pavement depth measurements, such as IRI and RUT, and the overall performance indices, such as PCI, DMI, and RCI are used as the output of individual models. Then the models are trained separately using gradient descent back propagation algorithm. Figure 6.4 shows the procedures, which include:

Step1: Data preparation. This procedure is a preparation stage that aims to facilitate the following model training procedure. The machine learning technologies intelligently learn meaningful information from the training dataset, which means that the quality of the data significantly influence the overall performance of the final model. Therefore, this step directly affects the effectiveness of the trained model. The main operations of this step are the feature indicator selection and data normalization.

Step 2: Neural Network architecture design. The numbers of layers, the number of nodes for each layer and other parameters of the structure should be decided in this step.

Step 3: Model training. The initially constructed architecture should be trained using the prepared training dataset. Gradient descent back propagation algorithm is selected for the model training in this step.

Step 4: Factor importance evaluation. The MIVs for the input factors are calculated in this step using the method discussed in the previous section.

Step 5: Analysis and discussion.

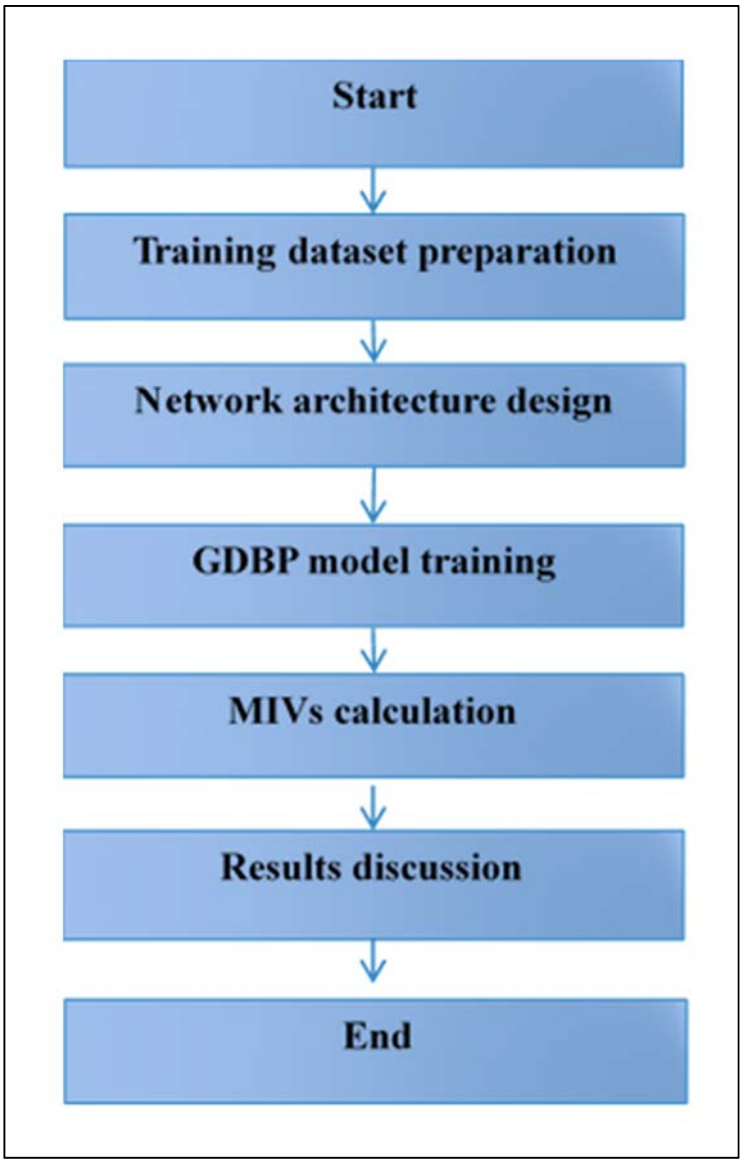


Figure 6.4: Methodology procedure

6.3.1 BPNN Architecture Parameters Selection Principle

Once the input and the output are decided, two other important parameters are determined in order to setup the initial NN architecture. These parameters are the number of hidden layers and the number of neurons for each hidden layer.

As has been introduced previously, the inputs of the architecture are the indicators of pavement surface distresses that can be detected by 2D image processing algorithms. The outputs are the pavement depth indices and the overall performance indices, respectively.

The nodes in the hidden layer are selected by two procedures:

Step 1: Set only one hidden layer, and decide three potential values for the number of nodes in this hidden layer based on the following principles:

1) $n_1 = \sqrt{nm}$, where n_1 is the number of nodes in the hidden layer. m is the number of nodes in the output layer. n is the nodes of the input layer.

2) $n_1 = \sqrt{n+m} + a$, where m is the number of nodes in the output layer. n is the number of nodes in the input layer. a is a constant value within the range of [1,10].

3) $n_1 = \log_2 n$, where n is the number of nodes in the input layer.

Step 2: Train the three models decided by the above parameters and select the one with the best performance;

Step 3: Judge whether the selected model meets the requirement or not by the following methods:

a. If the model has satisfied performance, then it can be regarded as the final model; Then continue with Step 4.

b. If not, then both add and subtract 1 from the initially decided nodes. Meanwhile, the number of nodes should be judged for whether it is within the range of [1, 10]. If yes, turn to Step 2; If no, then increase the number of hidden layers by 1 and turn to Step 1.

Step 4: End of processing.

The Sigmoid function is used as the activation function for all the NN models. Equation (6.24) is the mathematical definition of sigmoid function and Figure 6.5 shows its activation mechanism. Observing the Sigmoid curve shown in Figure 6.5, it can be seen that the outputs are projected to the range of [0,1] and have an 'S' shape.

$$Y = \frac{1}{1 + e^{-x}} \tag{6.24}$$

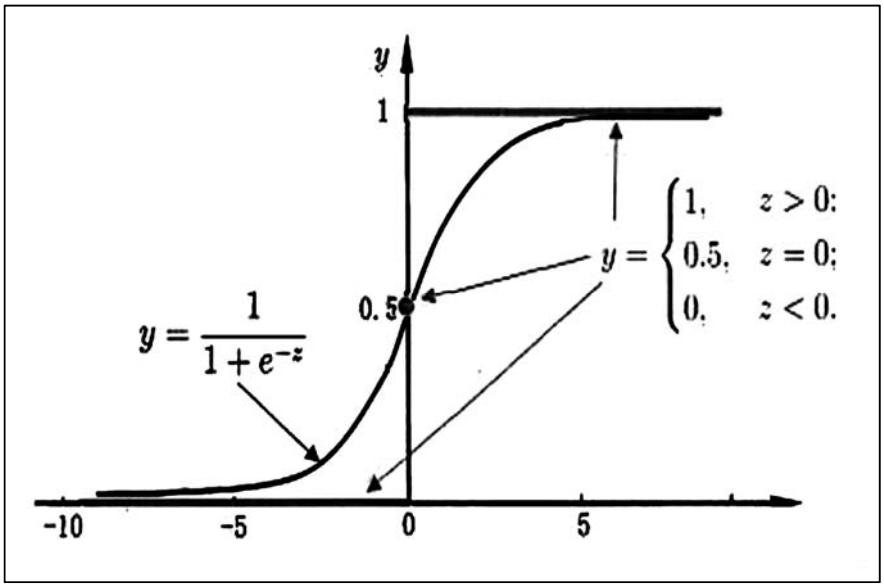


Figure 6.5: Sigmoid curve

The model training procedure usually includes thousands of iterations to update the parameters and minimize the total loss. Therefore, the criteria being used to stop the model training procedure are critical. Two criteria being used in this research on model training are:

(1) When the iteration reaches to 1000;

(2) When the average total error is lower than 0.03.

The training process cannot be stopped unless both above criteria are satisfied.

6.3.2 Model Performance Assessment

Two measurement indices are used to assess the performance of the models developed in this research. The first measurement is the coefficient of determination (R^2), which evaluates the fitness condition of the model.

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = 1 - \frac{\sum_i^n (x_i - x_p)^2}{\sum_i^n (x_i - \bar{x}_i)^2} \tag{6.25}$$

where $\sum_i^n (x_i - \bar{x}_i)^2$ is the total sum of squares. $\sum_i^n (x_i - x_p)^2$ is the sum of residuals. x_p denotes the predicted value.

Another measurement is the mean absolute error (MAE), which reflects the error of the model.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} | x_o - x_p | \tag{6.26}$$

where x_o denotes the expected value. x_p denotes the model output.

6.4. Data Analysis and Discussion

6.4.1 IRI Prediction and Influence Analysis

A comprehensive description of the image distress indicators is shown in Figure 6.6. Diverse types and parameters of the surface cracks have been used as the main indicators.

Those parameters include the length, width, severity, counting, percent of the transverse and the longitudinal cracks, as well as the severity, density, width and the percent of pattern/fatigue cracks. Besides cracks, other surface defects are considered such as raveling, bleeding, etc.

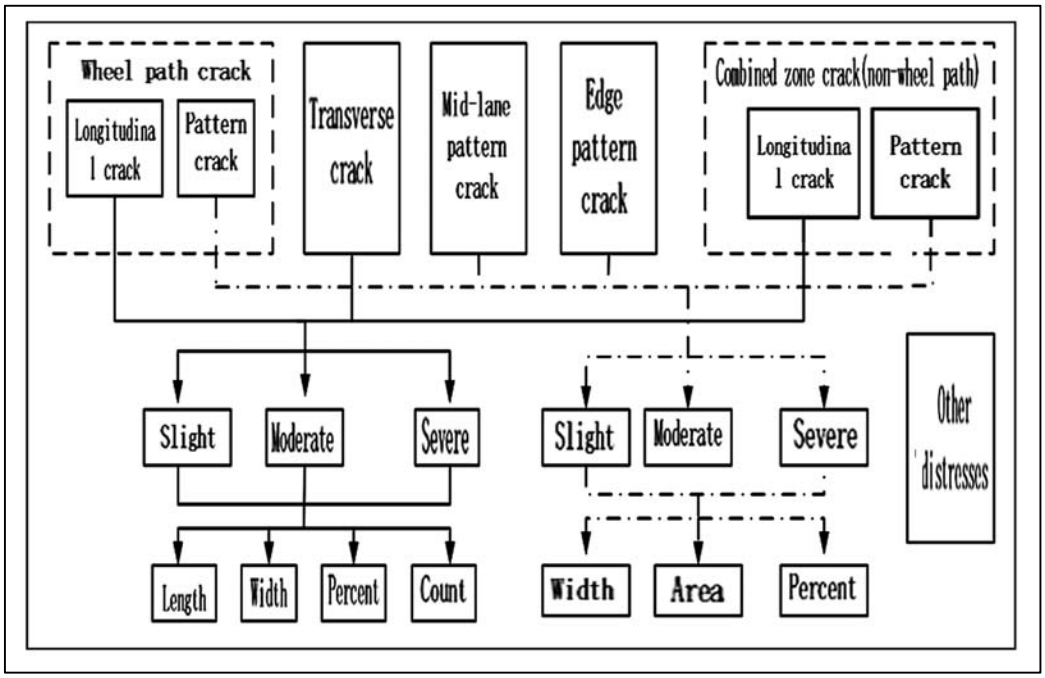


Figure 6.6: Data structure for model training

To be specific, 80% of the original normalized data are used for model training and cross validation. By architecture setup and model training, a four-layer BPNN is obtained to predict the value of IRI. As Figure 6.7 shows, the input layer includes the 59 kinds of road surface distress indicators which provide a comprehensive description of pavement surface condition. Given those image-based pavement distress indicators, the predicted IRI value can be obtained from the output layer. The basic setup parameters of the final model are: nodes in hidden layer1=19, nodes in hidden layer2=15 and learning rate=0.01.

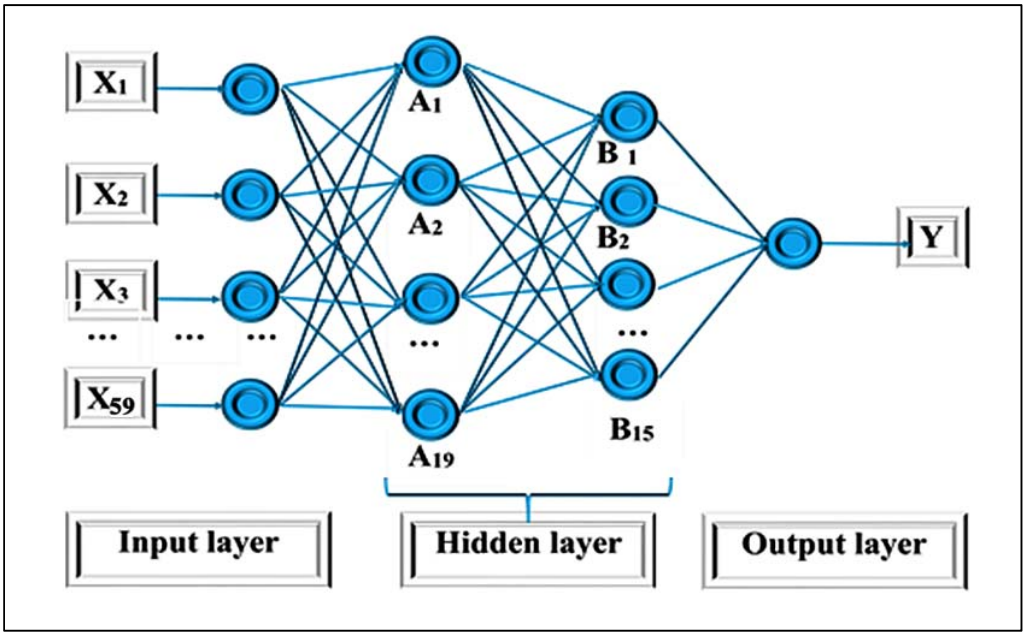
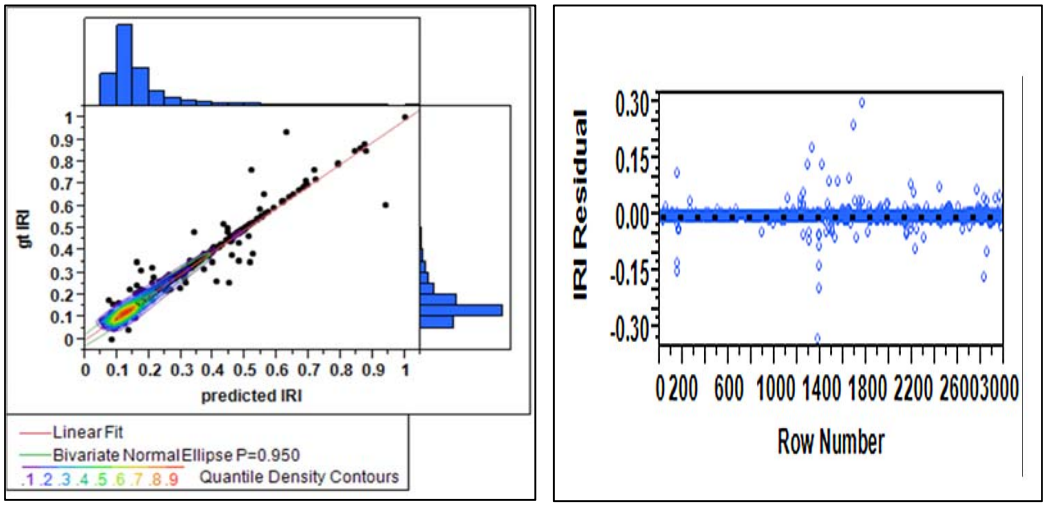


Figure 6.7: BPNN architecture for IRI prediction

When the optimal trained model is obtained, the rest of the data are used to test and assess the model performance. Figure 6.8 (a) shows a fitting line between the predicted IRI and the corresponding expected values. It can be observed that the $R^2 = 0.925$, which can be considered as satisfactory in terms of accuracy requirement. Meanwhile, the residual distribution in Figure 6.8(b) indicates that the residuals are randomly distributed and within ± 0.3.The model is easy to setup and train. Therefore, it is reasonable to conclude that this model has a satisfactory performance.



(a) Fitting results **(b) Residuals**

Figure 6.8: IRI prediction analysis:

Furthermore, the contributions of different factors to the IRI are analyzed using the optimist network model. As a result, the MIV values of different image-based indicators are obtained as shown in Appendix II (a). The top 10 (10 highest) significant contributions are selected and shown in Figure 6.9. It can be found that: (1) different surface distress indicators have diverse impacts on the IRI value. Some indicators only weight for 0.0001 or less, while some others have higher than 0.05 contribution; (2) in general, most of the crack indicators have more significant influence than the other surface distresses, such as raveling; (3) pavement wheel path cracks (longitudinal, transverse, alligator) have the most significant impacts on the IRI changing than the other distress indicators (especially the top 10 indicators). Based on those results, wheel path cracks are suggested to be the highest priority for repairing work schedule. For crack quantification, the percent of the cracked regions and the width of cracks can be considered as the effective measurements

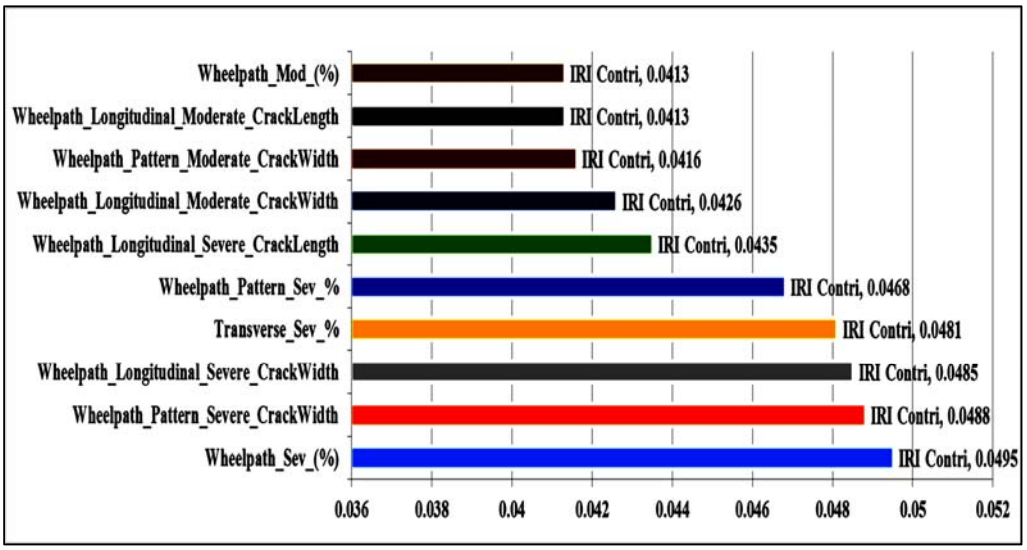


Figure 6.9: High contributions MIVs for IRI

6.4.2 RUT Prediction and Analysis

The mechanism of RUT prediction is similar to that of IRI prediction. After architecture adjustment and model training, the optimum RUT prediction model structure can be obtained as Figure 6.10 shows. The final model has only one hidden layer with 13 neurons. The best learning rate is 0.03.

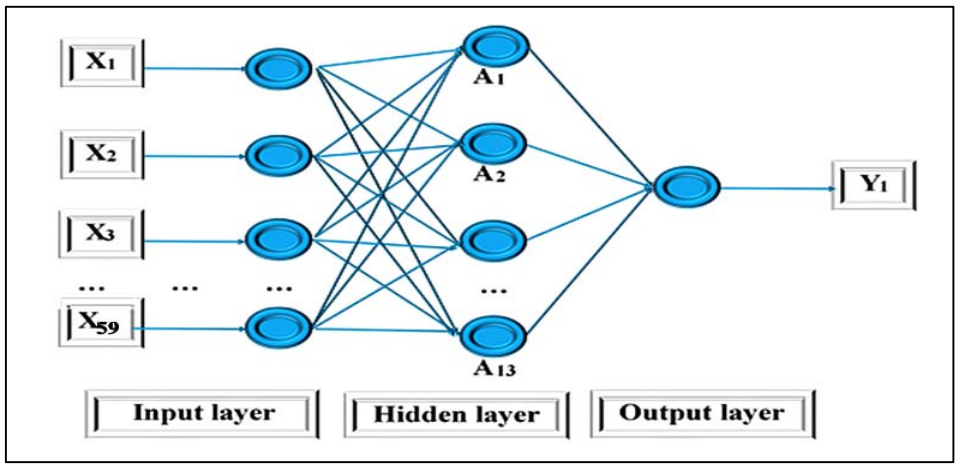
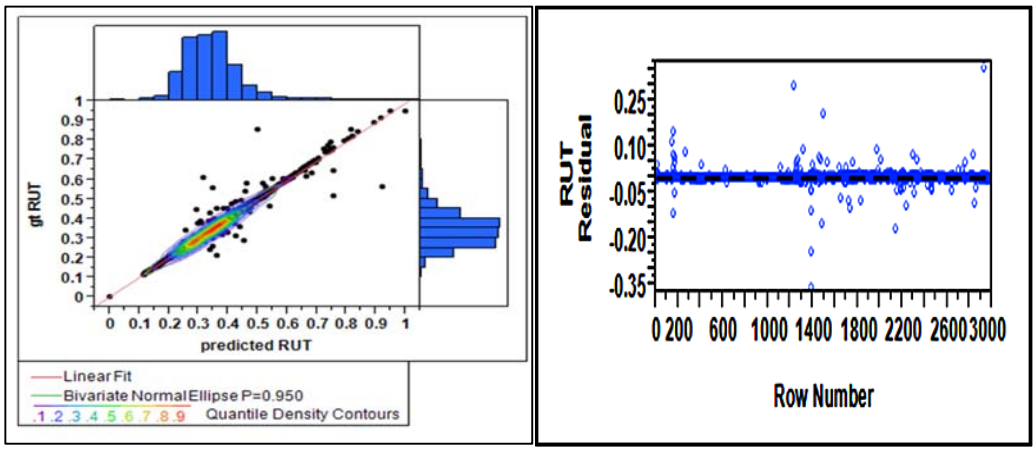


Figure 6.10: BPNN architecture for RUT prediction

The R^2 of the RUT prediction model is 0.975, represented by the fitting line shown in Figure 6.11(a). The distribution of residuals is shown in Figure 6.11(b), within ± 0.35. It can be seen that even though the absolute value of residuals is higher than that of the IRI prediction model, the $R^2$of RUT prediction is higher as well. All these results show the satisfactory performance of the models.



(a) Fitting results **(b) Residuals**

Figure 6.11: RUT prediction analysis

The MIVs of input indicators on the influences of RUT are shown in Appendix II (b). The top 10 indicators that have the highest contributions are shown in Figure 6.12. The findings from the RUT prediction are quite similar to those obtained in IRI prediction. (1) The influences of different indicators on the values of RUT vary significantly. The lowest MIV can be around 0.0001, while the highest MIV tends to exceed 0.05; (2) Pavement surface cracks exceed other types of surface distresses, therefore they can be regarded as the most important surface distress type; (3) More specifically, attention should be given to the transverse cracks and the wheel path cracks (longitudinal, transverse, alligator) with high priorities, because their MIVs are the highest among all the other distresses. Moreover, the crack width, length and area are all important indicators for the rutting analysis.

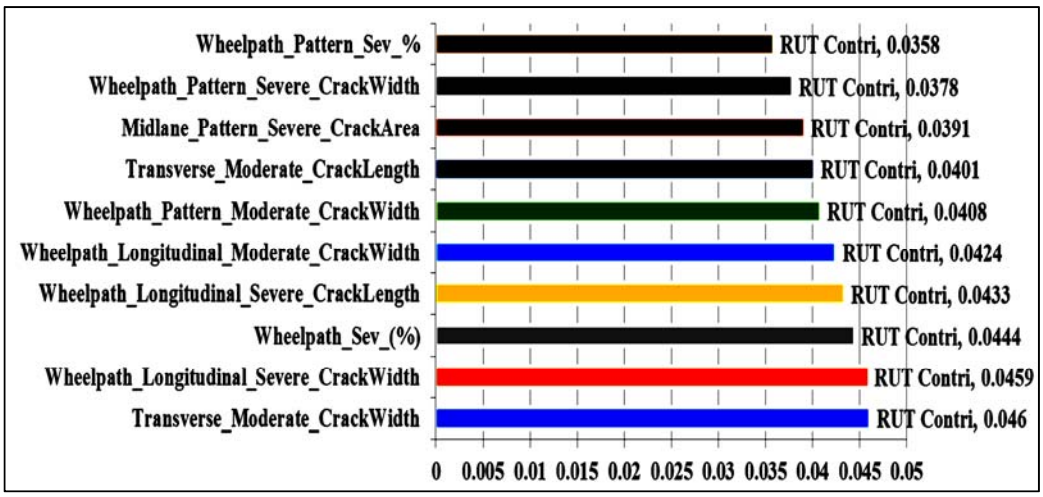


Figure 6.12: High contributions MIVs for RUT

6.4.3 DMI, PCI, and RCI Prediction Analysis

In similar ways, the BPNN models for DMI, PCI and RCI predictions are also obtained using the training datasets. The network architectures, results analysis, and the top 10 MIVs for different performance indices are shown through Figures 6.13 to 6.21. Detailed MIVs of different indicators with the corresponding indices are listed in Appendix II(c) (d) and (e).

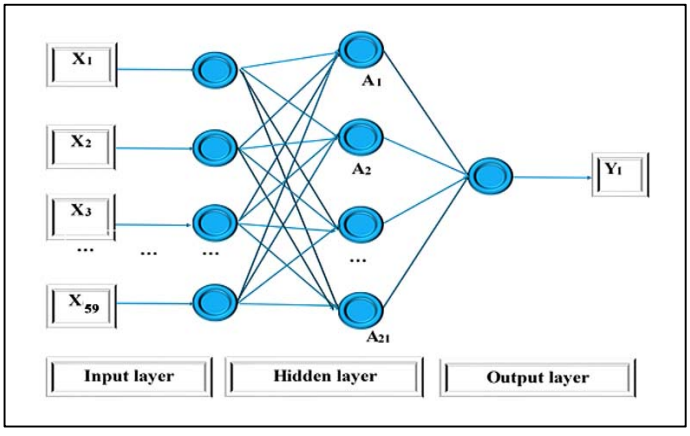
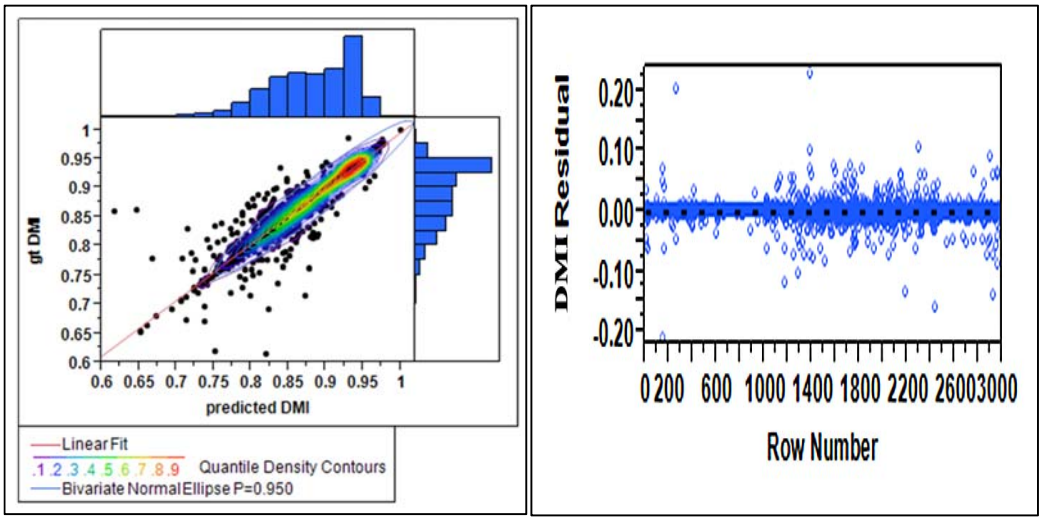


Figure 6.13: BPNN architecture for DMI prediction



(a) Fitting results **(b) Residuals**

Figure 6.14: DMI prediction analysis

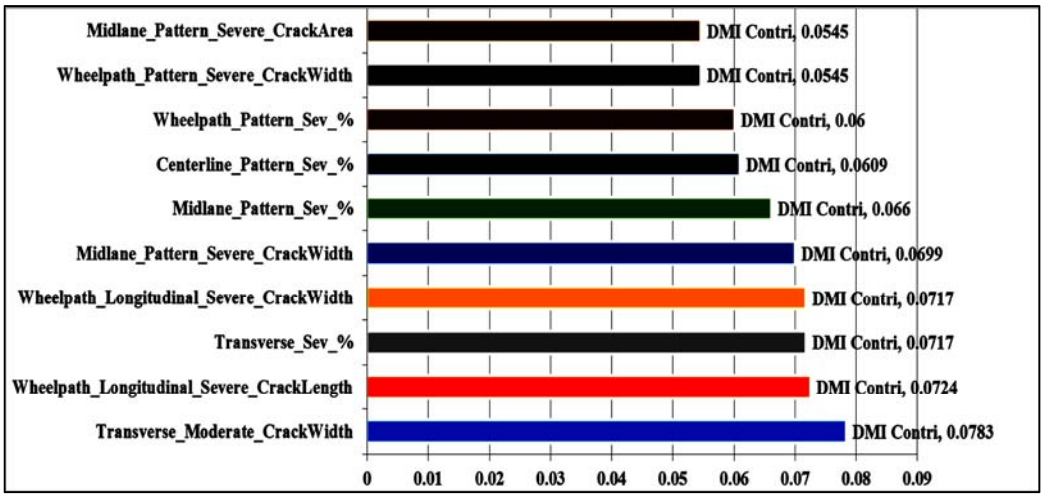


Figure 6.15: High contributions MIVs for DMI

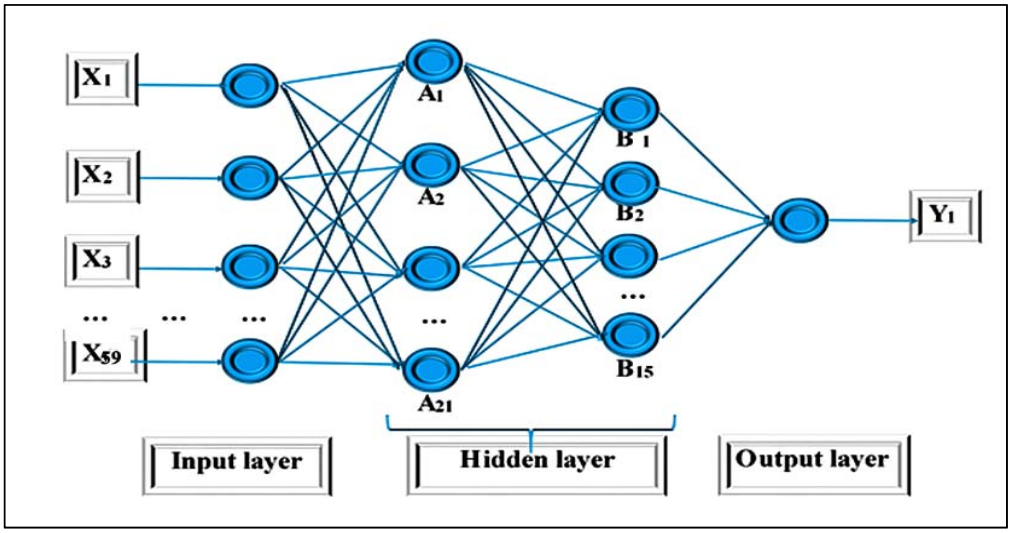
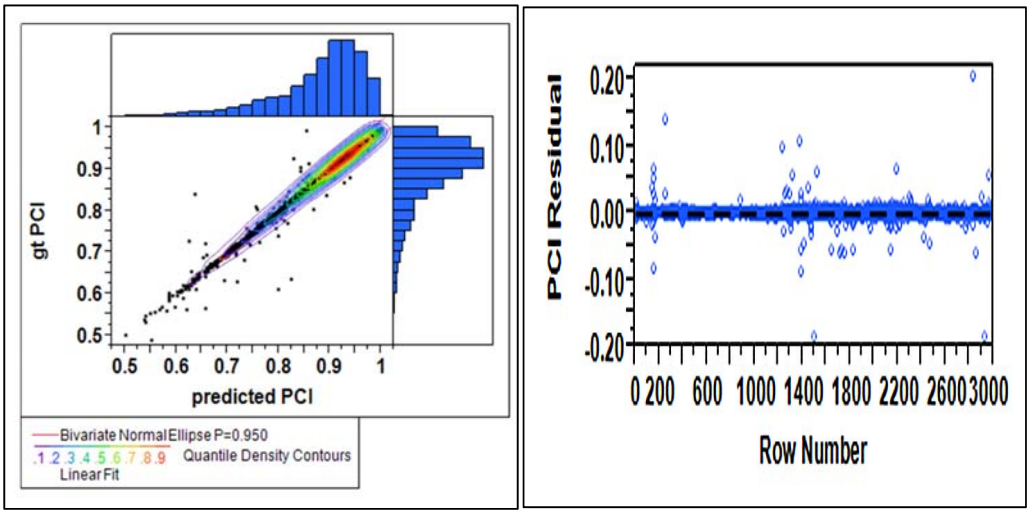


Figure 6.16: BPNN architecture for PCI prediction



(a) Fitting results **(b) Residuals**

Figure 6.17: PCI prediction analysis

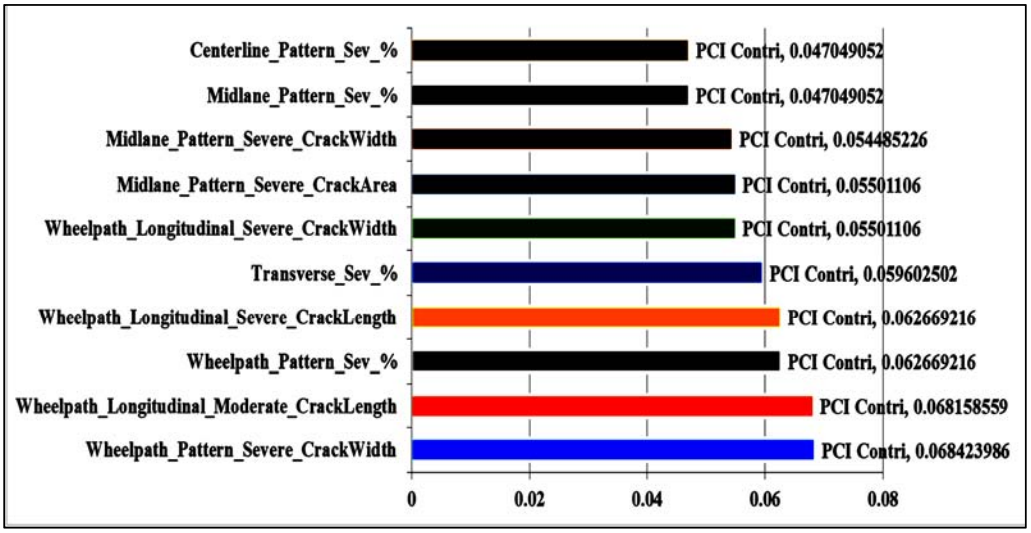


Figure 6.18: High contributions MIVs for PCI

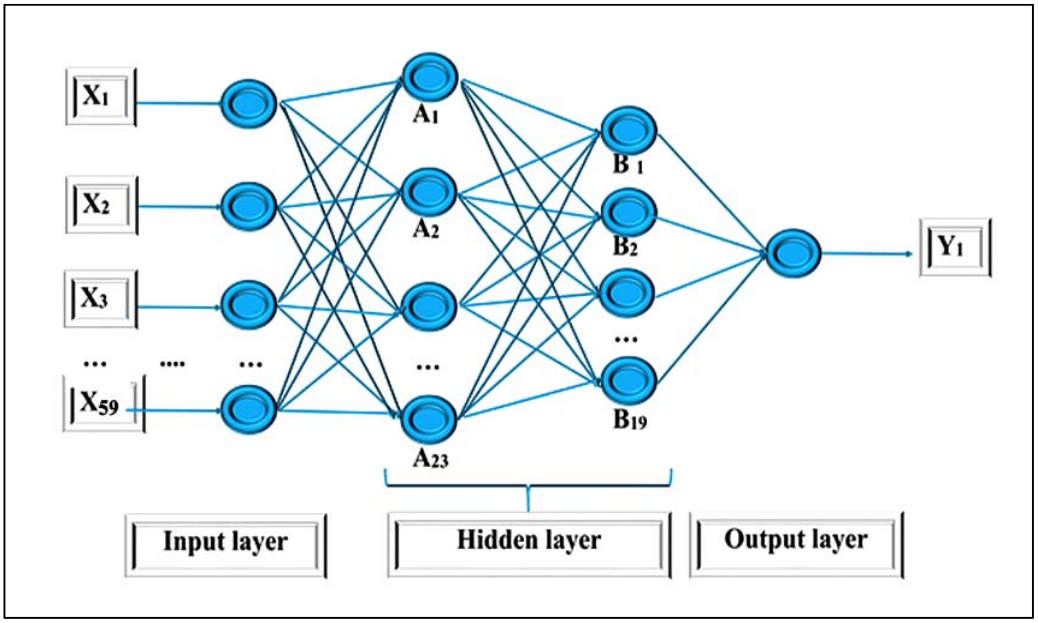
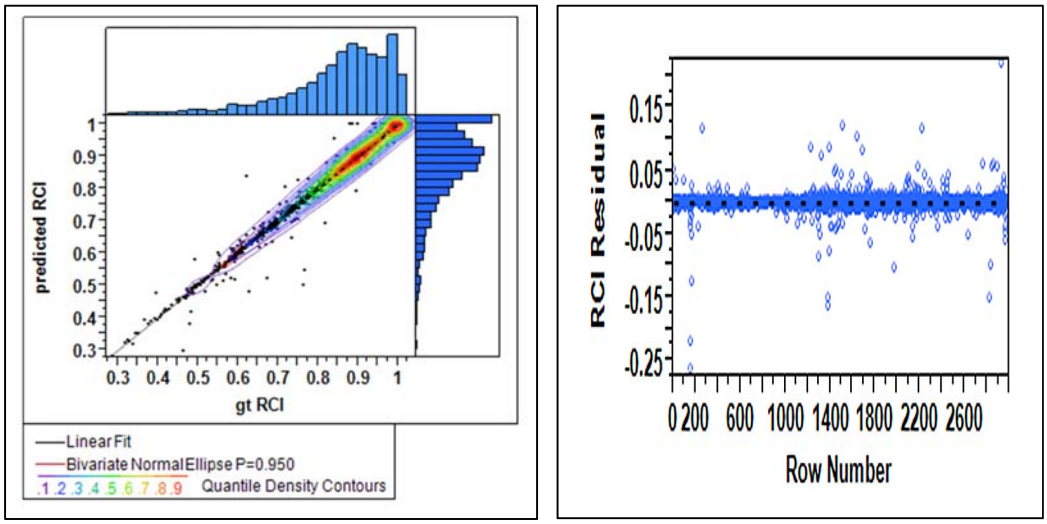


Figure 6.19: BPNN architecture for RCI prediction



(a) Fitting results **(b) Residuals**

Figure 6.20: RCI prediction analysis

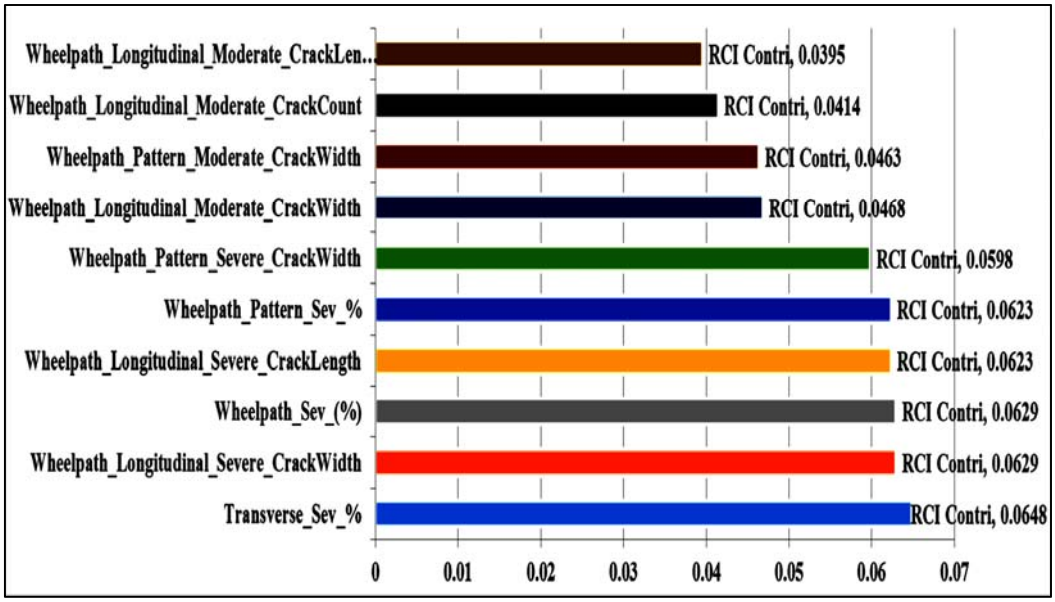


Figure 6. 21: High contribution MIVs for RCI

6.5. Chapter Summary

This chapter used the same dataset as in chapter 5. However, this chapter aims to figure out the dominant pavement surface distresses that influence the pavement roughness condition and the overall performance indices.

The methods being used for the model training are BPNNs, followed by MIVs analysis. Specifically, in the beginning, the BPNN based roughness condition indices prediction and the overall performance indicators prediction models are developed. On this basis, the MIVs for different input image indicators are calculated. Those MIVs represent the contribution condition of each input indicator to the corresponding output indices. Therefore, they are used to determine the dominant pavement surface distresses.

The data analysis shows that the wheel path cracks are recognized as the most important road surface defects influencing the roughness condition and the overall performance**.** Hence it is recommended that wheel path cracks should be the top priorities for consideration when planning the M&R treatments.

CHAPTER 7. Pavement Structural Condition based Performance Prediction Using Optimized Support Vector Machines

7.1. Introduction

Pavement Management System (PMS) (Haas, Hudson, & Kennedy, 1975) can be considered as a group of tools or methods that assist better pavement treatments decision making. Thereby, pavement can be maintained under satisfying condition over a long period of time. The effectiveness of pavement treatments (Uddin, Meyer, Hudson, & Stokoe, 1985) is greatly related to the reliability of pavement condition assessment. Therefore, comprehensive pavement condition evaluation (Wang, Hou, & Williams, 2010; Sun, Zhao, Li, Hao, & Huyan, 2015; Li, Huyan, Tighe, Ren, & Sun, 2017) benefits not only pavement condition evaluation departments, but also the treatments planning organizations.

In a general perspective, a comprehensive pavement condition database can be regarded as a combination of different types of surface distresses data, pavement longitudinal/transverse profile data, surface texture data and structural condition data. Specifically, pavement surface distresses are caused by complex effects, which include environmental influences (Robinson, Beg, Dossey, & Hudson, 1996), pavement service aging (Wu, Flintsch, Ferreira, & Picado-Santos, 2012) and extensive traffic. Manual pavement condition inspection is the simplest way, but it can be potentially inaccurate due to subjectivity. At the same time, extensive human labour and large quantities of time commitment are required. Exposing the inspectors to dangerous working situations are a common phenomenon, especially in the highway condition surveys. Therefore, using well programmed either Destructive Testing (DT) or Non-Destructive Testing (NDT) devices as

a replacement of human operations is highly required.

Researchers have realized the necessity of automated pavement condition evaluation in early 1990 when highly advanced computing technologies began to show promising future in facilitating engineering work. For example, automatic pavement crack sealing route can be planned based on the acquired crack map through digital image processing based automated pavement crack detection. Over years of development, automated pavement condition systems have upgraded into high integrated, high efficiency, high accuracy, highly automated, 3D equipment facilitated systems (Qiu, Wang, Zhang, Li, & Moravec, 2018) from the initial low performance systems. Current high-performance automated pavement condition survey systems have been used across the world for analysis of diverse functional classes of pavements. The most integrated automatic pavement management systems can provide comprehensive pavement condition calculations including surface distress parameter calculations, rutting calculations, surface texture analysis, and even 3D road surface reconstruction. The algorithms being used in those systems are traditional digital image processing approaches, digital signal processing methods, statistics analysis methods, and the recent highly advanced machine learning methodologies. Based on those calculations, the overall performance indices are automatically calculated as parts of the system outputs for pavement treatment decision making.

Therefore, periodic pavement condition survey and evaluation are necessary for more effective pavement management. With the advanced application of advanced technologies in the engineering side, conventional manual pavement condition assessment is gradually replaced by automated devices. Well programmed and calibrated devices can obtain precise network level pavement condition indices, such as PCI, DMI, etc. However, by observing the algorithms of these indices, it has been found that conventional evaluation indices only focused on pavement surface distresses without considering the structural condition, resulting in less accurate condition evaluation. Hence this chapter sought to develop

pavement structural condition evaluation approaches through optimized Support Vector Machines (SVMs). The data distributions are divided into four levels indicating different performance conditions. On this basis, pairwise SVMs prediction models are trained by pavement surface modulus data. During the model training procedure, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) methods are used to improve overall performance. Finally, PSO-SVM with 200 iterations is obtained as the best model.

7.2. Data Collection and Analysis

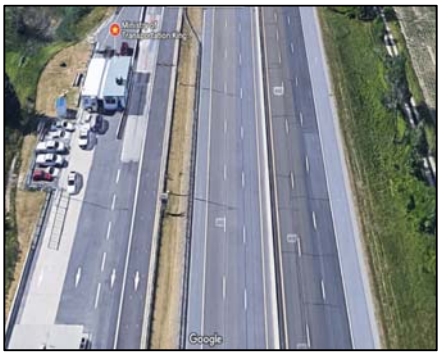
7.2.1 Highway Data Collection

The widely used pavement dynamic strength test devices include the Heavy Weight Deflectometer (HWD), Falling Weight Deflectometer (FWD), and Light Weight Deflectometer (LWD). HWDs are mostly in airport pavement testing, which requires for expanded load range. FWDs are the most widely used devices for both pavement design and in-suit construction quality assurance testing. The FWD is a computerized device which exerts a certain load by the falling mass to the pavement surface. The falling mass is known and fixed according to a programmed height. The stiffness modulus data of different pavement layers can be obtained by internal sensors of FWD. The LWD is portable and light weight and can measure the in-place pavement strength. The applied load of LWD is also changeable according to different pavement surface types. By connecting to the GPS location component, the LWD is easy to use especially for construction quality assurance measurement. Therefore, for flexibility, this research used LWD to pavement surface deflection modulus data for the structural condition integrated performance modeling.

SurPro, a road surface profile measurement device, is used for pavement roughness measurement and to collect the wheel paths IRI data. The pavement performance

methodology: RUBIX, is used to measure the pavement roughness condition and obtain the RCI measurement evaluation.

Pavement performance data collection in supporting this chapter has been conducted in June 2017, which is conducted with the help of my colleagues in Centre for Pavement And Transportation Technology (CPATT). Figure 7.1 shows a basic overview of the data collection road section and the on-site test photos. SBR represents South Bend Right and SBL represents South Bend Left, which are used as denotations in the dataset.



(a) Test section **(b) A bird eye-view of the test section**

(c) SurPro roughness data collection (d) LWD surface modulus data collection

Figure 7.1: Data collection (Picture taken by Ju Huyan in on June 14, 2017, at Highway 400, Ontario, Canada, in the data collection with the help of colleagues from CPATT)

In order to provide the detailed dynamic changing of pavement surface deflection within the time of load exerted, the surface modulus changing has been recorded as Figure 7.2 shows. A time interval of 5-m has been used to record the surface deflection with the reference to a complete even pavement surface. It can be observed that the deflection keeps almost a constant phase delay with respect to the load applied to it. However, there always exists a long tail of variance in deflection when the load has disappeared. This phenomenon exists in almost every test point, because the condition of the pavement is influenced by the complex roadway environment. Therefore, to ensure the reliability of further analysis, the raw data points which have significant variance in the tail are removed.

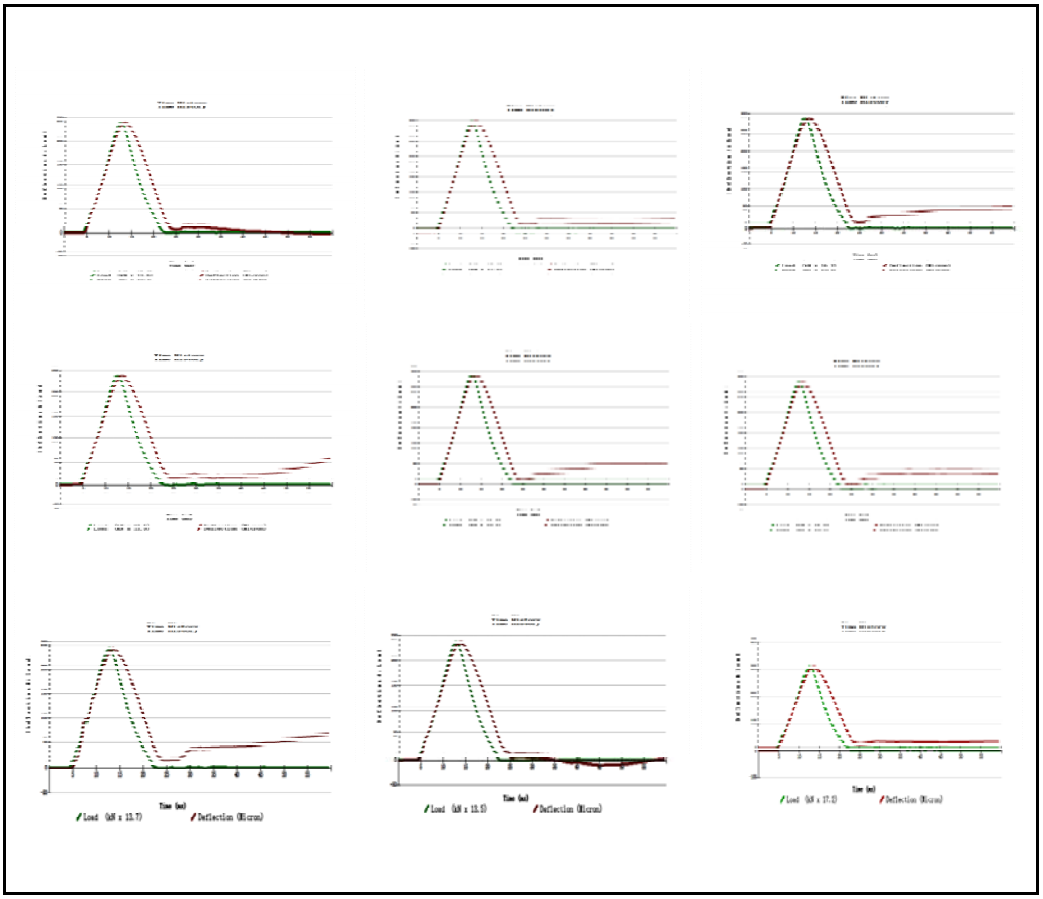


Figure 7.2: Pavement surface deflection changing in LWD test

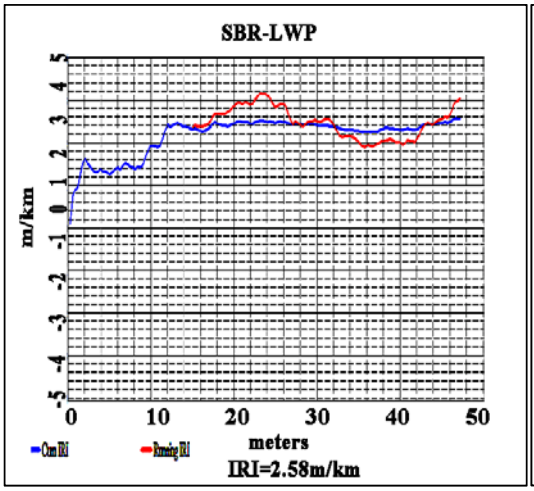
7.2.2 Roughness Analysis

Figure 7.3 and 7.4 are the analysis of the roughness conditions of two lanes: SBR and SBL, respectively. The tested average IRI values are: IRI_{SBR_LWP}=2.58, IRI_{SBR_RWP}=3.77, IRI_{SBL_LWP}=3.62, IRI_{SBL_RWP}=4.84. This means that the SBL is rougher than SBR. Meanwhile, for each individual lane, the right wheel path (RWP) is rougher than the left wheel path (LWP).

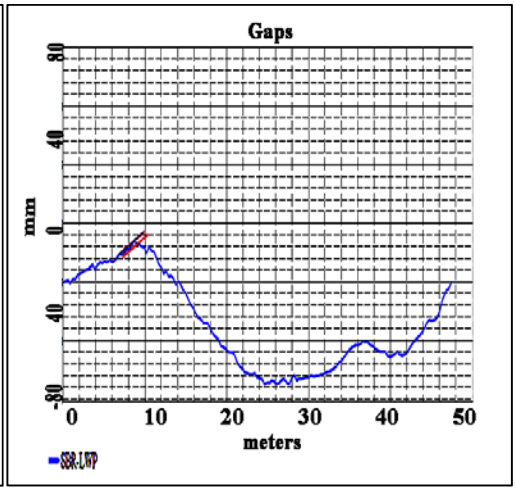
In the IRI curves shown in Figure 7.3 a, c and Figure 7.4 a, c, two other kinds of detailed IRIs are also calculated beside the average IRI, which are the cumulative IRI and the 15m IRI. (1) The cumulative IRI (the blue curves): this kind of measurement type shows the general roughness condition of the tested section. (2) The 15m IRI (the red curves): this kind of measurement shows the local roughness condition of the tested road section, because this value is calculated only based on the last 50 steps during Dipstick data collection.

To measure the local failure of the tested road, the gap analysis has been used in the collected profile data. The method is called: gaps under straightedge. The demonstrations are shown in Figure 7.3b, d, and Figure 7.4b, d. In this method, a specified line is defined to represent the straightedge, which are the pink lines in these figures. The other line is defined below the straight edge (red line in these figures) with the predefined maximum allowable gap between these two lines. Then, these two lines are moved simultaneously from one side of the curve to the other side. The failed points (might be caused by the error in the data collection or the worse condition of the corresponding road section) are those points which stay at the outside of the region between these two lines. By this way, the local performance condition of the pavement can be determined. The failed points should be deleted from the database in the following model development procedures as they cannot reflect the general condition of the pavement and may influence the performance of the

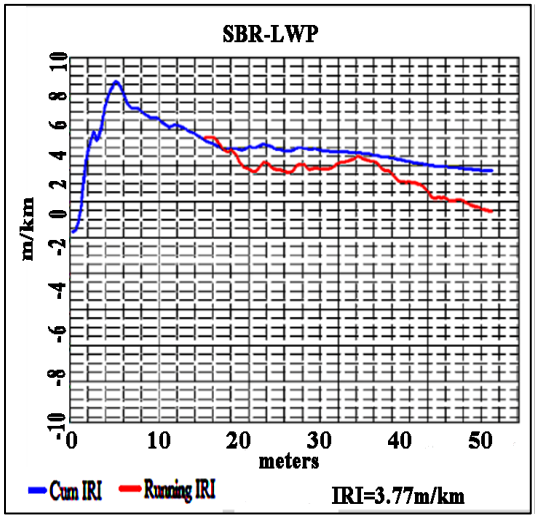
developed model. In Figure 7.3 and Figure 7.4, the maximum allowable gap is set to 3mm and totally 5 points have been deleted from the raw dataset.



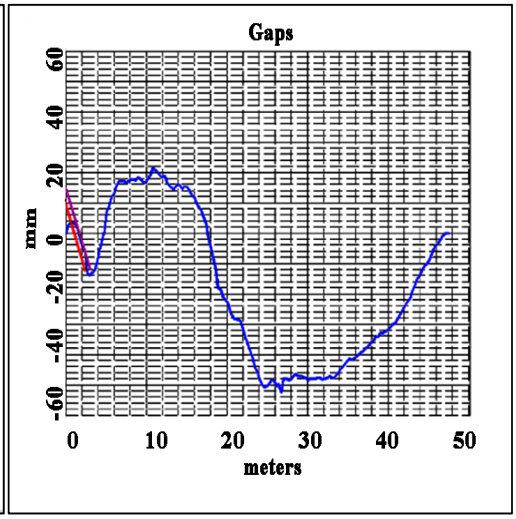
(a) SBR_LWP IRI



(b) SBR_LWP profile gaps

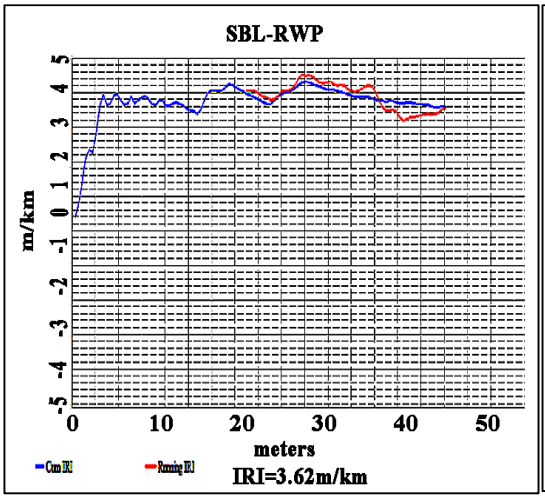


(c) SBR_RWP IRI

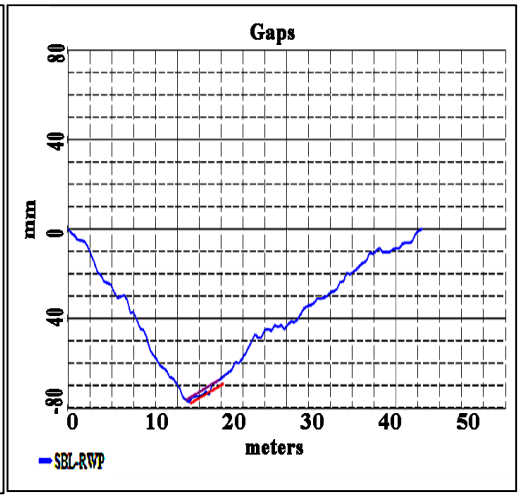


(d) SBR_RWP profile gaps

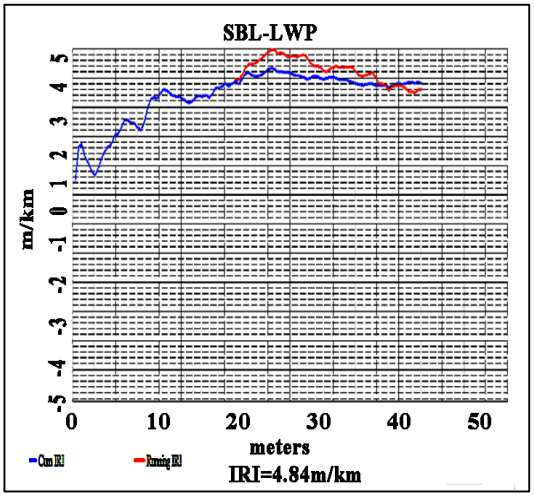
Figure 7.3: SBR IRI test results



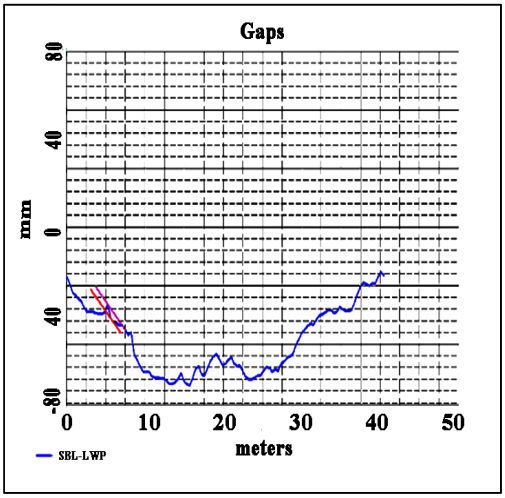
(a) SBL_LWP IRI



(b) SBL_LWP profile gaps



(c) SBL_LWP IRI profile gaps



(d) SBL_LWP profile gaps

Figure 7. 4: SBL IRI test results

7.3. Support Vector Machine

As one of the most widely used machine learning techniques, SVMs is popular in diverse fields for solving the classification, regression and recognition problems. The advantages

of SVM methods are (1) SVM can provide global and unique solutions; (2) SVM method has less chance of being overfitted to the training data; (3) SVM can project the complex problem from the original dimension to a hyperplane of another dimension, in which the calculations are conducted. As a result, only the results are returned to the original problem dimension. This purpose is realized by the definition of the kernel function. By this way, the computational complexity is reduced significantly. Meanwhile, it also provides the model designer with the opportunity of considering the problem from an engineering perspective.

To be straight forward, the primary purpose of SVM methodology is to find the optimal hyperplane which has two significant properties: (1) can separate two different categories with the best performance; (2) can maximize the margin between the nearest points and the hyperplane in different categories. Figure 7.5 illustrations the basic concept of the SVM method. The kernel function (which also called 'kernel trick') acts as the projector in mapping the input feature to its corresponding higher dimension. Then, by finding the best suitable hyperplane in the projected dimension, the final classification objective can be achieved. Finally, the results can be returned to the original lower dimension for engineering use.

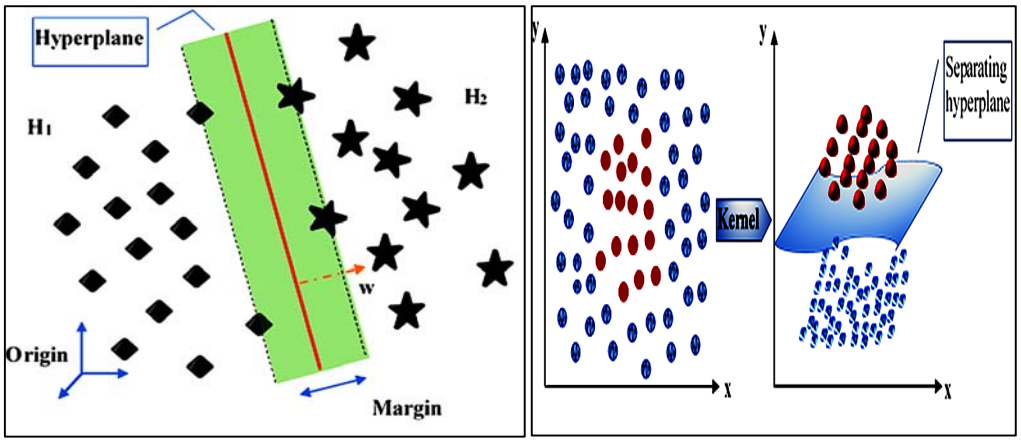


Figure 7.5: Illustration of margin and separating hyperplane and 'Kernel trick'

The mathematical solutions of the SVM method (Cortes & Vapnik, 1995) are explained as follows:

The simplest two categories classification problem can be considered as the development of a 1/-1 classification model with respect to R^2. The training set can be defined as $T = \{(x_i, y_i)\}_{i=1,\ldots,N}$, where x_i denotes the featured item of the $i-th$ sample; $y_i \in \{1, -1\}$ denotes the $i-th$ label. The linear classifier $f(x)$ can be expressed by Equation (7.1):

$$\begin{aligned} & f(x) = w^T x + b, \\ & s.t. \begin{cases} f(x) > 0, x \in class1 \\ f(x) < 0, x \in class-1 \end{cases} \end{aligned} \tag{7.1}$$

As discussed previously, a maximum distance γ should be defined, which is expressed by Equation (7.2)

$$\begin{aligned} & \max_{w \in R^m b \in R} \gamma \\ & s.t. y_i \frac{w^t x_i + b}{\|w\|} \geq \gamma, i = 1, 2, \ldots, N \end{aligned} \tag{7.2}$$

To simplify the above model with constraint, the model can be transformed into Equation (7.3):

$$\begin{aligned} & \max_{w \in R^m b \in R} \frac{\hat{\gamma}}{\|w\|} \\ & s.t. y_i (w^t x_i + b) \geq \hat{\gamma}, i = 1, 2, \ldots, N \end{aligned} \tag{7.3}$$

It can be observed from the above equation that the model is homogenous with respect to parameters w γ and b. Therefore, multiplying w γ and b by a constant number, say λ, does not affect the final solution. Then, two replacement operations can be conducted: (1) replace $\hat{\gamma}$ with a constant value of 1; (2) replace the minimization of $||w||^2$with the maximization of term $\frac{1}{||w||}$. According to these principles, the dual function of the problem

expressed in Equation (7.3) can be obtained as expressed in Equation (7.4).

$$\begin{aligned} &\min_{w\in R^m b\in R} \|w\|^2 \\ &s.t. y_i(w^t x_i + b) \geq 1, i = 1, 2, ..., N \end{aligned} \tag{7.4}$$

Then, consider another mostly commonly condition: the original data are not linearly distributed. In this circumstance, the slack variable: ξ_i, should be introduced. The slack variables aim to allow some instances to fall off the determined maximum margin. In this time, the previous objective function should be revised as:

$$\begin{aligned} &\min_{w\in R^m b\in R} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ &s.t. y_i(w^t x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \end{aligned} \tag{7.5}$$

where C denotes a trade-off, the parameter used to avoid overfitting.

The Lagrange multiplier method can be used to solve the above constrained problem. Then, the final purpose becomes finding the best solution for Equation (7.6):

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n} \alpha_i \\ &-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\Phi(x_i) \bullet \Phi(x_j)) \\ &s.t. \begin{cases} \sum_{i=1}^{n} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases} \end{aligned} \tag{7.6}$$

where (x_i, y_j) denotes the model training dataset. n denotes the number of training samples. α is the parameter vector. Each individual item of the vector represents the contribution level of the corresponding training data. $\Phi(x)$ denotes the kernel function which maps the input data into a higher dimension. 's.t.' stands for 'subject to', representing the constraint of the objective function.

For the most important kernel function selection, this research selected the most widely used Gaussian function which is expressed in Equation (7.7):

$$K(x_1, x_2) = \Phi(x) = \exp(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}) \\ = \exp(-g \bullet \|x_1 - x_2\|^2) \tag{7.7}$$

$$g = -\frac{1}{2\sigma^2} \tag{7.8}$$

where the Gaussian curve parameter g controls the performance of the final model in the given training dataset.

It should be noted that the traditional SVM approach is only suitable for two-category classification problems. In this research, a four-category classification problem is faced. Therefore, the pairwise SVM, which can be considered as a simple integration of several original SVMs, is employed for the prediction model development. The pairwise SVM can be realized in the following steps:

Step 1: One–verse-one SVM classification model development. Design SVM classifiers between each two categories for the randomly paired group of the training dataset. Generally, $k \times (k-1)/2$ classifiers are required for a specific k-category classification problem. In this research $k = 4$, so there are totally 6 SVM initial classifiers to be developed.

Step 2: Vote for each category. In the beginning, the scores for every category are assigned to zero. Next, judge the category assigning of each individual data point with regard to its classification results of Step1 and vote for the corresponding category increasing by 1.

Step 3: Selection and deciding: Select and assign the highest score to each item as the final classified category. Considering the i^{th} point as an example. If there are 4 out of 6 SVM classifiers voted at this point to category 1, then for category 1, its voted value is 4, which is the highest vote. Therefore, the i^{th} point should be classified into category 1 as the final result.

To summarize, two important parameters, which are the constraint parameter C and the Gaussian curve shape control parameter g, should be carefully selected. In the following part, the basic Grid Search (GS) method, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) method are used to find the best values for these two parameters. Meanwhile, the performance of different methods is evaluated for the prediction of pavement riding comfort index values.

7.4. Initial SVM Model for Performance Assessment

7.4.1 SVM Model Establishment

Riding Comfort Index (RCI) is an objective index which is calculated based on the measured IRI data as defined by MTO PMS-2. IRI only reflects the surface roughness condition, without considering the structural performance of the pavement. Using IRI is simple and straightforward, but it may not reflect the actual performance of the pavement, because sometimes the minor structural failures of pavement are not reflected on the roughness condition changing. Considering this, pavement surface modulus together with the IRI data are used as the input. The labeled RCI values are used as the output. Figure 7.6 shows the basic framework of the SVM method for structural condition facilitated performance prediction.

To provide a more clarified evaluation for the riding quality of pavement, the Riding Performance Level (RPL) is proposed based on RCI. The proposed RPL classified the riding quality of the pavement into 4 levels, which are: Excellent ($\mathrm{RCI} \geq 7$). Good ($5 \leq \mathrm{RCI} < 7$), Moderate ($3 \leq \mathrm{RCI} < 5$)and Poor ($\mathrm{RCI} < 3$). On this basis, the RCI values are labelled to different levels which represent output categories with respect to the input features. 80% of the collected data are used as a training set to obtain the best model. Then,

the best model is tested using the left 20% of the data.

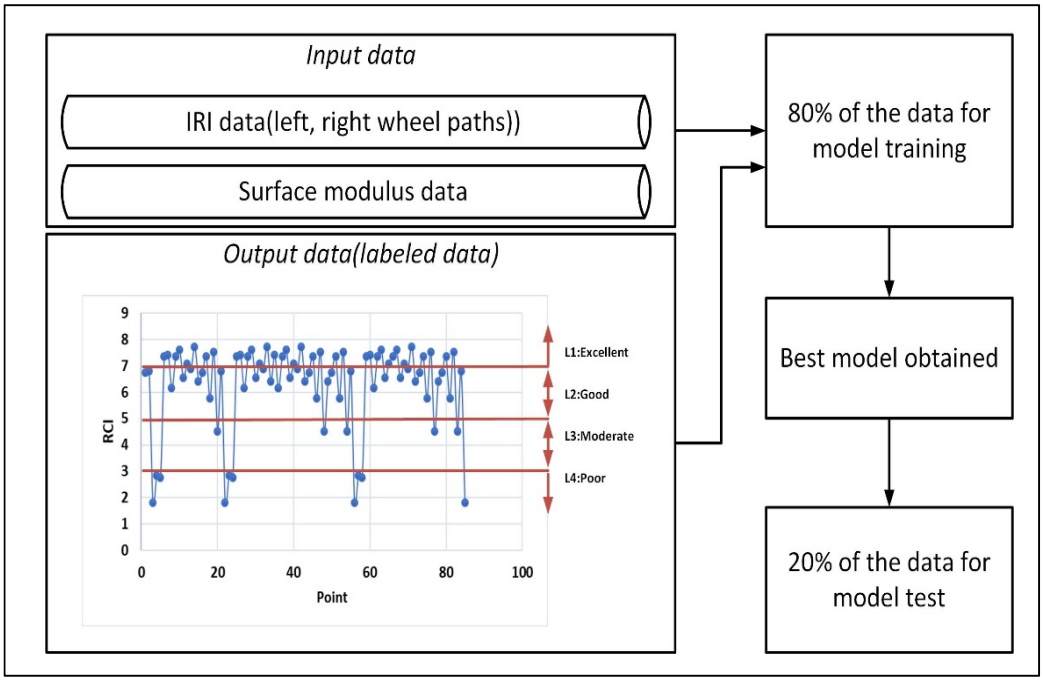


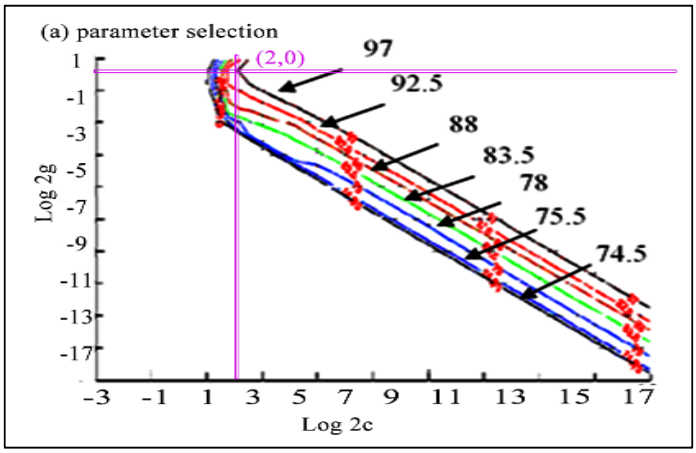
Figure 7.6: Framework of SVM pavement performance model

7.4.2 Grid Search Method for SVM Parameter Selection

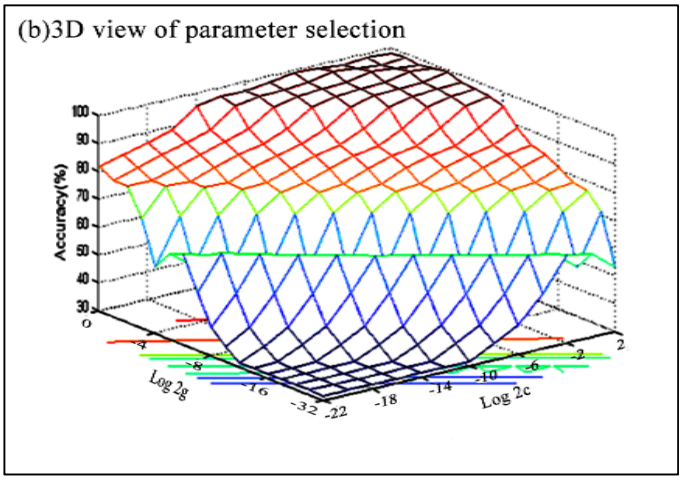
As discussed in section 7.3, the accuracy of SVM models is highly dependent on the selected parameters, which are penalty parameter (c) and kernel function parameter (g). Therefore, in this part, the simplest parameter method called Grid Search (GS) is be used for the best parameter selection. The grid search method is realized by two simple operations. Firstly, the solution domain is divided into unit grid cells with each cell representing a potential solution. Then, one evaluates the performance of each potential solution until finding the best one.

According to the method explained previously, the solution domain is set as: $2^{-10} < c, g < 2^{10}$. The search step between each potential solution is 2. The reason is that the best

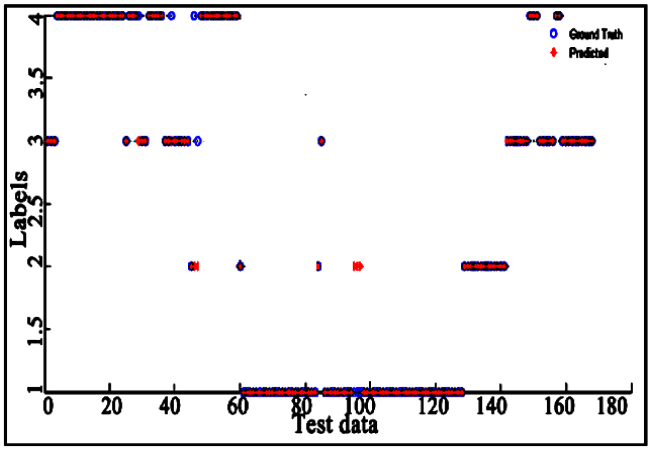
parameters are always found to be a power of 2, which is influenced by the mechanism of computer processing. Meanwhile, in order to obtain a global solution, a wide searching range (2^{-10}~2^{10}) is used. The demonstration of the best parameter selection using the GS method and the corresponding test performance of the final model are shown in Figure 7.7. It can be observed that when c=4 and g=1, the model can get the best classification accuracy, which can be up to 97.2182%.



(a) Contour chart



(b) 3D view of parameter selection



(c) Test accuracy

Figure 7.7: Grid Search optimized SVM test results

It should be noted that the GS method can provide the best parameters for the model, but it is based on the sacrifice of a large amount of computation resources. That means the GS method is not time efficient, especially when dealing with a big dataset, which may cost more than half an hour to obtain the optimal result. However, the high efficiency of the methodology is strongly required in engineering fields. Considering this, heuristic optimization methods, which are the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) method, are used to optimize the SVM models. The detailed optimization methodologies of these two approaches are explained in the following sections.

7.5. GA Optimization of SVM Method

Genetic Algorithm (GA) simulates biological systems using selection, crossing, and mutation operations to optimize the model (Goldberg & Holland, 1988; Gao, Li, Woo, & yun Tian, 2018; Piersanti & Orlandi, 2018). Equation (7.9) can be used to define the general function of the GA approach**:**

$$GA = (N, F, s, c, m, p_c, p_m) \tag{7.9}$$

where N denotes the population size. F denotes the fitness function. $s, c, and\ m$ denote three operators: the selection operator, cross operator, and mutation operator. p_c and p_m denote the probabilities of the cross and mutation operation, respectively. The detailed process of the GA method is shown in Figure 7.8. The key operations in Figure 7.8 are as follows:

Data coding and population initialization. In the GA method, the chromosomes refer to the potential solutions that are binary coded to mimic the human genes. Meanwhile, the population is initialized using random data.

1) Fitness function definition. The fitness function is used to evaluate each individual chromosome's adaptivity. The chromosomes with the best adaptivity are selected as the best solution to the corresponding problem. Based on this principle, the fitness function is defined by the accuracy of the model under cross validation, which is expressed in Equation (7.10)

$$F = CV\left(\frac{\text{correctly classified}}{gt}\right) \tag{7.10}$$

   where gt denotes the ground truth value. CV represents "cross validation".

2) Selection (or Copy): Roulette method (Figure 7.9) is used to decide which one should be considered as the potential selection item. In the roulette method shown in Figure 7.9, the probability of a specific individual being selected is calculated by Equation (7.11):

$$p_i = \frac{f_i}{\sum f_i} (i - 1, 2, 3, ..., N) \tag{7.11}$$

where p_i denotes the selection probability of the i^{th} individual. f_i denotes the fitness of the i^{th} individual. $\sum f_i$ denotes the accumulated fitness condition of the whole population.

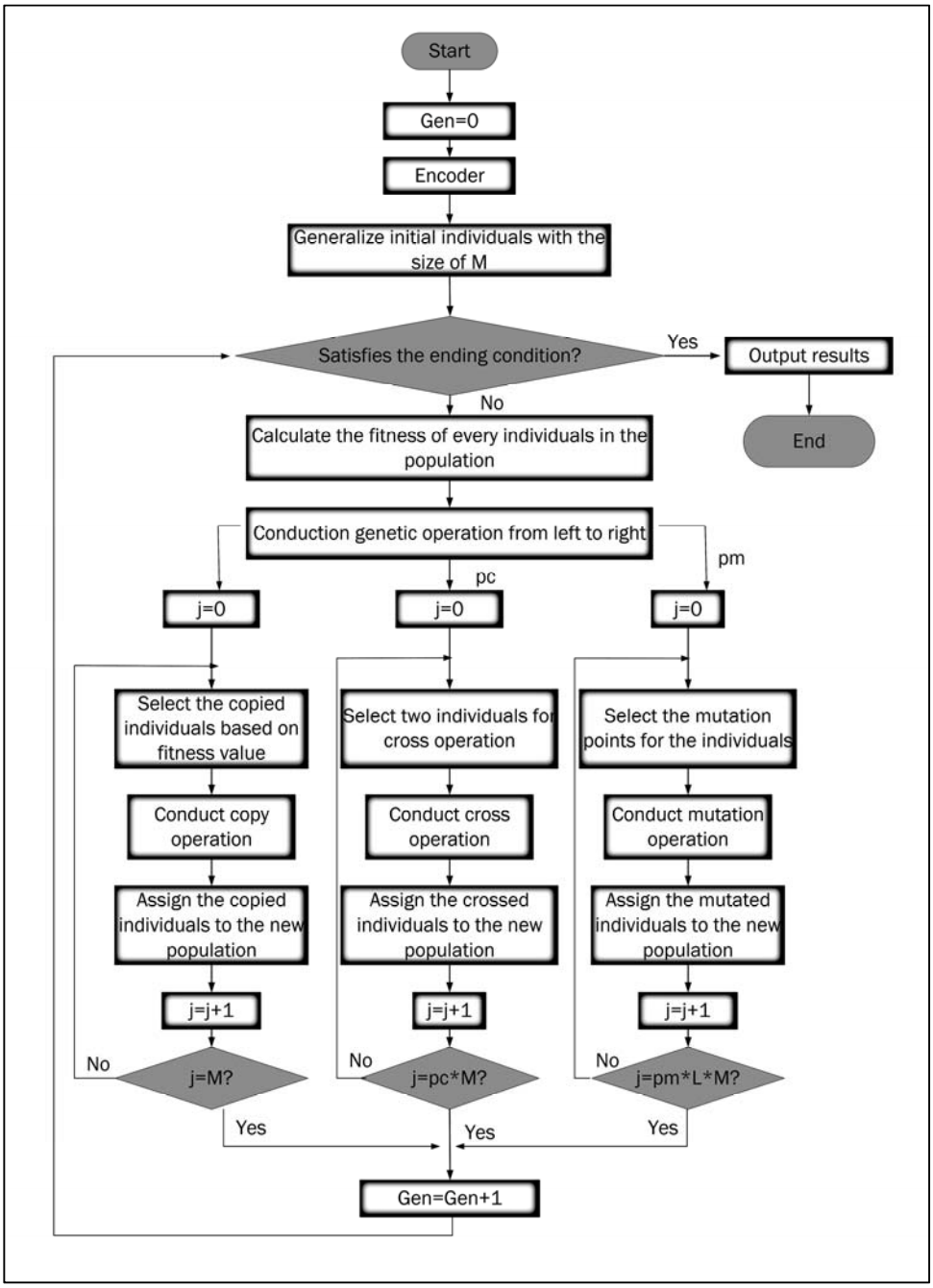


Figure 7.8: Flow chart of GA optimization process

The functional mechanism of Figure 7.9 is: The scale values on the ring represent the calculated fitness condition of each individual, which should be calculated using Equation

(7.11). The arrow is fixed to its initial position, and the outer ring rotates freely. Then, the individual to be selected is decided by the position where the arrow stops. Detailed processes include the following steps:

Step1: Sequentially accumulate the fitness of each individual in the population. The accumulated fitness values are denoted as $S_i(i = 1,2,3 \dots \dots n)$;

Step2: In the range of $[0, P_n]$, randomly generate numbers with even distribution, which can be represented as r;

Step3: Find the first value that satisfies the condition of $P_j \geq r$, which means the j^{th} individual should be selected;

Step4: Repeat step 3 successively until the size of the new generation is equal to the current generation.

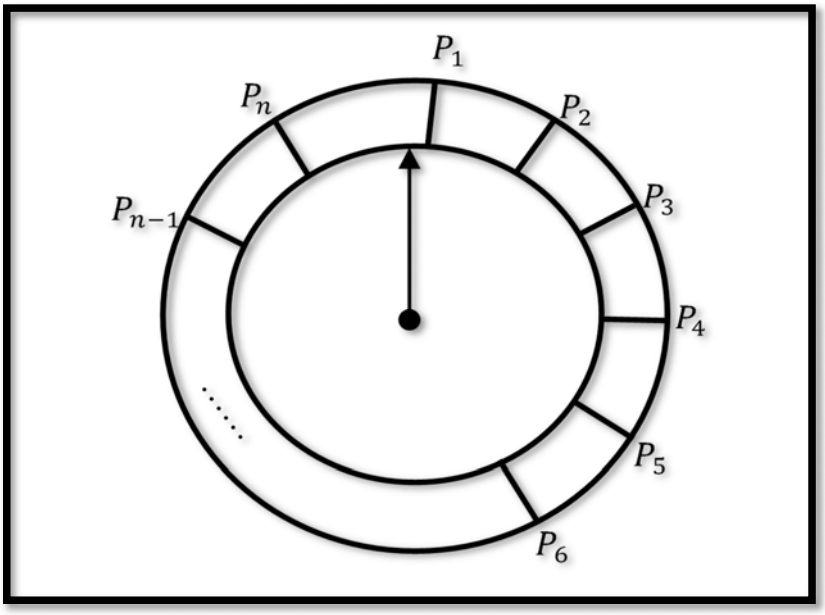


Figure 7.9: Roulette method

3) Crossover. The crossover means that the new generation is formed by exchanging some genes in the current generation. Paired individuals are the ones being considered as exchangeable in the current generation.

Step 1: Randomly make pairs for every two individuals of the current generation. As a result, a total of $\frac{N}{2}$ paired groups should be generated for a population size of N;

Step 2: Randomly set a point for each individual pair as the initial potential crossover position. It should be noted that for a chromosome with the length of λ, $(\lambda - 1)$ positions are generated as the potential crossover position;

Step 3: Use the probability function defined in Equation (7.12) to judge probability of the exchangeable positions. Then, new individuals can be generated.

$$p_i = \frac{N_C}{N} \tag{7.12}$$

where N denotes the size of the population, and $N_c$denotes the number of exchangeable individuals. p_i denotes the cross probability.

4) Mutate operation. The mutation operation is conducted as follows:

Step 1: Calculate the mutation position of the genes for each individual use Equation (7.13):

$$p_i = \frac{B}{N \bullet \lambda} \tag{7.13}$$

where B denotes the number of genes that should be processed with mutation operation. N denotes the population size. λ denotes the length of the individual. p_i denotes the mutation probability of the i^{th} individual. It should be noted that the mutation operation is conducted for each individual. Therefore, mutation probability p_i is defined based on a certain individual.

Step 2: Replace the original values of the decided gene points with their opposite value or

their allele value.

It should be noted that the crossover happens more frequent than mutation in the genetic fields. Therefore, the mutation probability is usually much lower than the crossover probability.

5) Genetic algebra. Usually, the genetic algebra is selected in the range of [50, 500] based on the characteristic of the specific problem.

7.6. PSO Optimization of SVM Method

In literature, Particle Swarm Optimization (PSO) has been used in many fields to solve the optimization problems, such as the electrical model parameters optimization (Calvini, Carpita, Formentini, & Marchesoni, 2015), solar cell model optimization (Khanna, Das, Bisht, & Singh, 2015), robot programming optimization (Tuvayanond & Parnichkun, 2017), etc. Based on searches of the current literature (Feng, Xiao-Ting, Qian, Wei-Xing, & Qi, 2013; Poli, 2007; Xinchao, 2010; Xu & Yu, 2018), PSO method is capable of obtaining the global optimal solution for the corresponding problem without overfitting. PSO method simulates the food searching behavior of birds, who search their nearest region at first and then update their positions with certain velocity based on the information they acquired. The velocity and position updates are the key operations of the PSO method. Therefore, PSO method is simpler than GA method.

For the velocity, this research proposed the contraction factor facilitated velocity update function, which is expressed in Equation (7.14)

$$\begin{aligned} & v_{i(t+1)} \\ & = K \bullet [v_{it} + \varphi_1 r_1 (p_i^{best} - x_{it}) \\ & + \varphi_2 r_2 (g_i^{best} - x_{it})] \end{aligned} \tag{7.14}$$

where v_{it} denotes the velocity of the i^{th} particle in the t^{th} iteration. P_i^{best} denotes local best solution of the i^{th} particle. P_i^{gbest} denotes the global best solution of the i^{th} particle. K denotes the contraction factor which can facilitate the convergence of the iteration. The contraction factor can be calculated by Equation (7.15) which is influenced by two parameters. The contraction factor can control the convergence of the iteration behavior and influence the overall process. The relationship between K and φ_1, φ_2 can be expressed by Equation (7.15):

$$K = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|}, \quad \varphi=\varphi_1+\varphi_2, \ \ \varphi>4 \tag{7.15}$$

where φ_1 and φ_2 are the two parameters which do not have any actual meaning. However, the selection of the parameters should be based on the constraint defined in Equation (7.15). Once the velocity update value is acquired, the position of the particle can be updated based on its current position. The position update function is expressed in Equation (7.16):

$$P_i(t+1) = P_i(t) + v_i(t+1) \tag{7.16}$$

Figure 7.10 shows the detailed process of the PSO method and the corresponding pseudocode of the algorithm.

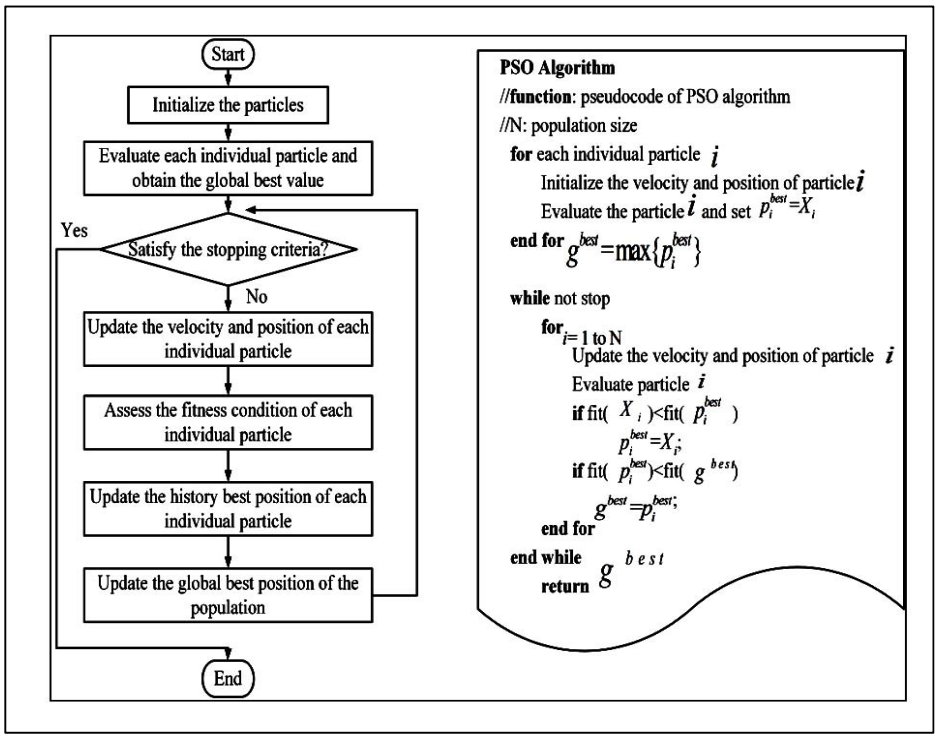
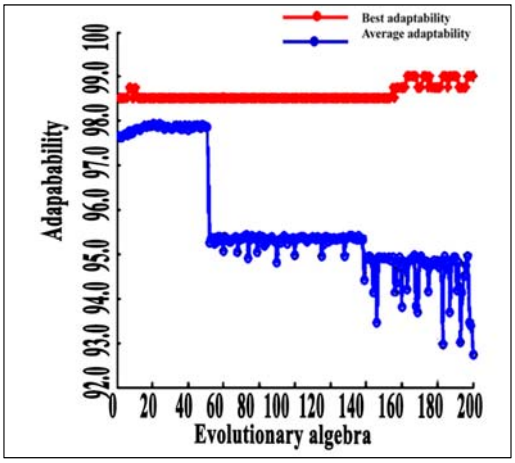
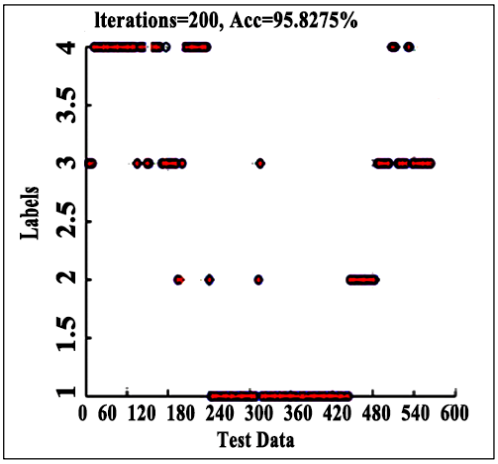


Figure 7.10: PSO optimization method

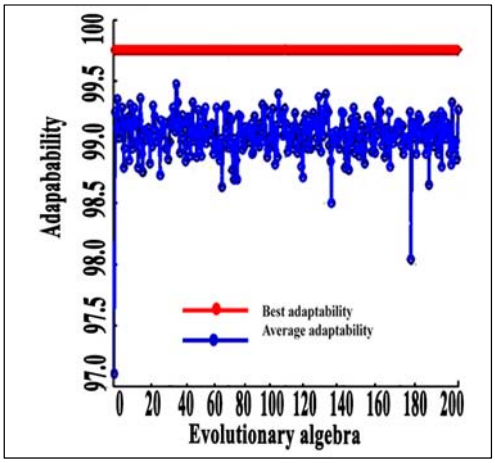
Using the above discussed GA and PSO facilitated SVM methods, the Riding Performance Level (RPL) prediction models are trained on the collected dataset. Figure 7.11 shows examples of the model training performances and the corresponding test performance of both GA and PSO methods. Comparing the adaptability changing curves of Figure 7.11a1 and a2, as well as the test accuracy curves of Figure 7.11b1 and b2. Fixed iterations of 200 has been used for both methods. It is clear that PSO optimized models outperform the GA optimized models.



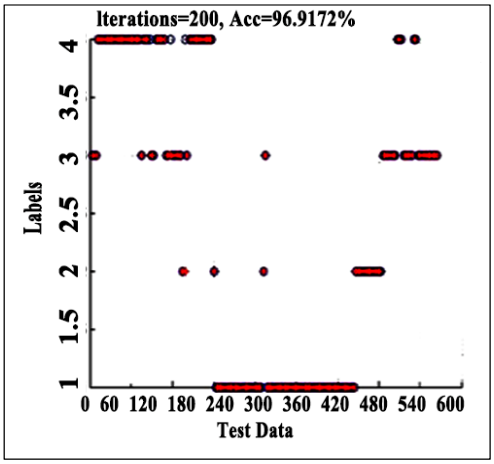
(a1) Fitness changing of GA _SVM model



(a2) Accuracy of GA _SVM model



(b1) Fitness changing of PSO _SVM model



(b2) Accuracy of PSO _SVM model

Figure 7.11: Heuristic method optimized SVM models testing

7.7. Analysis and Discussion

The most important advantage of heuristic methods compared with the computationally intensive grid search method is the high efficiency. However, the iterations used in the methods also significantly impact the efficiency of the methods. The lack of iterations may result in less accurate models but too many iterations require a longer time for model

training and can cause over fitting in the final model. Therefore, for both GA_SVM and PSO_SVM, the model training iterations are varied in the range of [100, 600] to find the best iterations. The performance under the iterations =100, 200, 300, 400, 500 and 600 is selected as the observation points. The best parameters under these conditions are:

$$GA\_SVM_{100}\{best_c = 34.5648\,, best_g = 5.8641\,\}$$

$$GA\_SVM_{200}\{best_c = 28.1456\,, best_g = 4.1084\,\}$$

$$GA\_SVM_{300}\{best_c = 25.4642\,, best_g = 3.8949\,\}$$

$$GA_{SVM\,400}\{best_c = 26.4259\,, best_g = 3.0148\,\}$$

$$GA_{SVM\,500}\{best_c = 28.4916\,, best_g = 3.1479\,\}$$

$$GA_{SVM\,600}\{best_c = 30.1102\,, best_g = 3.0469\,\}$$

$$PSO\_SVM_{100}\{best_c = 30.4419\,, best_g = 5.0647\,\}$$

$$PSO\_SVM_{200}\{best_c = 4.0112\,, best_g = 0.6847\,\}$$

$$PSO\_SVM_{300}\{best_c = 12.4957\,, best_g = 2.7216\,\}$$

$$PSO\_SVM_{400}\{best_c = 16.4816, best_g = 2.8457\,\}$$

$$PSO\_SVM_{500}\{best_c = 18.4752\,, best_g = 2.9115\,\}$$

$$PSO\_SVM_{600}\{best_c = 22.4816\,, best_g = 3.1105\,\}$$

The results can be seen in Figure 7.12 and 7.13. The findings are as follows:

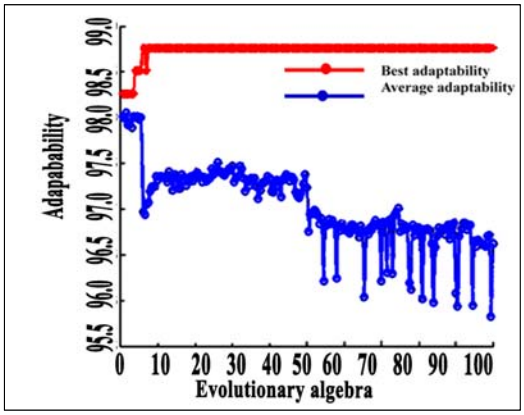
1) Comparing the adaptability changes of modeling training process under different iterations in Figure 7.12a, c, e, g and k, the optimistic aspect can be found that the best

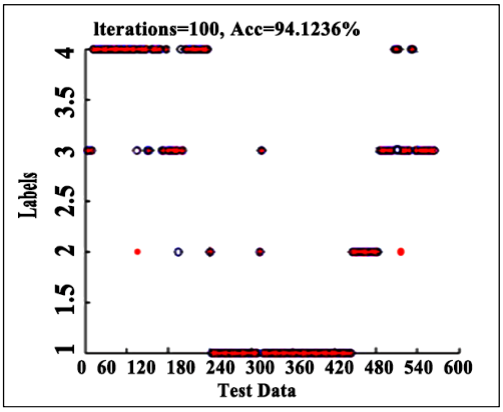
adaptability curves in each subfigure are all around 98%. Only minor fluctuations are observed with the increase of the training iteration. Moreover, considering the average adaptability curves in these subfigures, it can be seen that the curves with 200 or 300 iterations are more stable than the others, except for the beginning part of less than 20 iterations which show the start of model training and the existence of much fluctuations are reasonable. Through these analyses, it is reasonable to consider the 200 and 300 iterations as the potential best solutions.

2) Comparing the accuracy test results under different iterations (Figure 7.12b, d, f, h, j, and l), it can be found that the accuracy increases from 94.1236% (100 iterations) to 95.8275% (200 iterations). After 200 iterations, the accuracy gradually decreases to 92.4172% at 600 iterations. The highest accuracy can be observed at 200 iterations, though the accuracies of the other iterations are all acceptable.

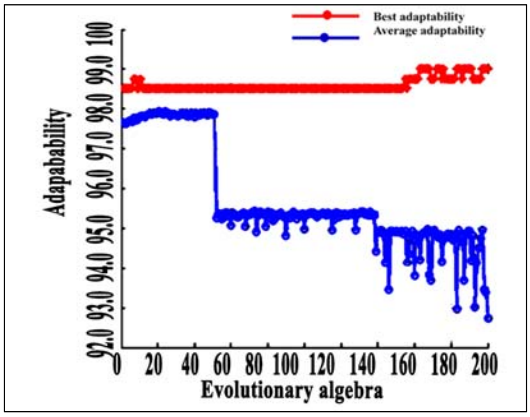
3) Considering the results of PSO_SVM methods shown in Figure 7.13 in similar ways. Similar conclusions can be arrived at those from GV_SVM in Figure 7.12, which means that 200 iterations of training can also be considered as the best for PSO_SVM method. Moreover, comparing the overall performance between GA_SVM and PSO_SVM at 200 iterations, it can be concluded that PSO_SVM outperforms GA_SVM under the same conditions. The best result is obtained in PSO_SVM method with 200 training iterations (96.9172%).

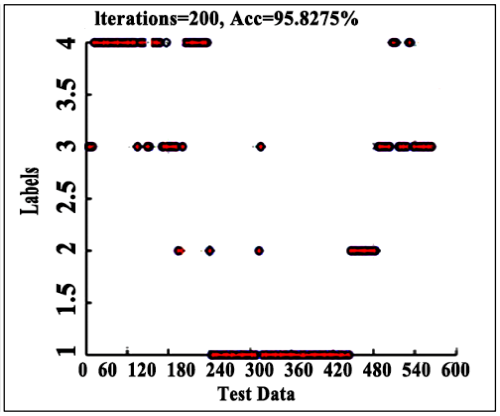
Moreover, a significant processing time difference has been found between these two methods, which are 48seconds for PSO_SVM and 126 seconds for GA_SVM. Therefore, both GA and PSO optimized SVM models can achieve acceptable performance for RPL prediction. However, PSO optimized SVM model under 200 training iterations is suggested.



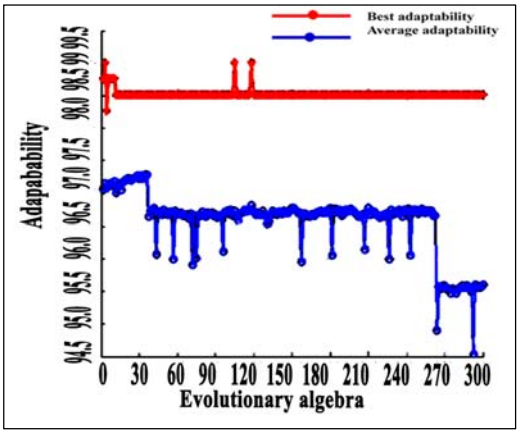


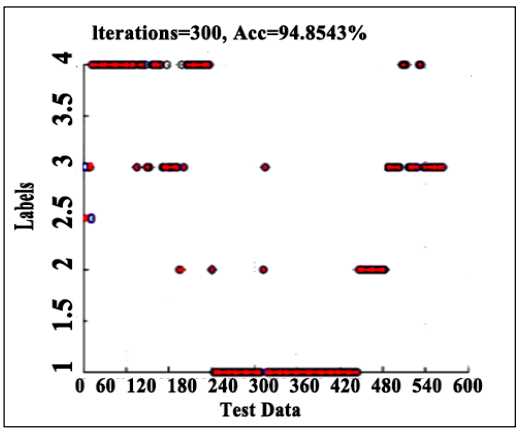
(a) Fitness changing for 100 iterations (b) Accuracy for 100 iterations (94.1236%)



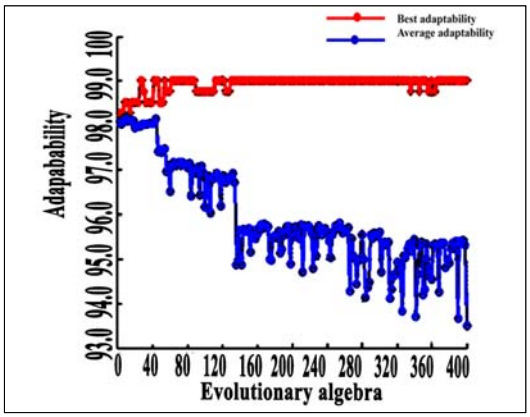


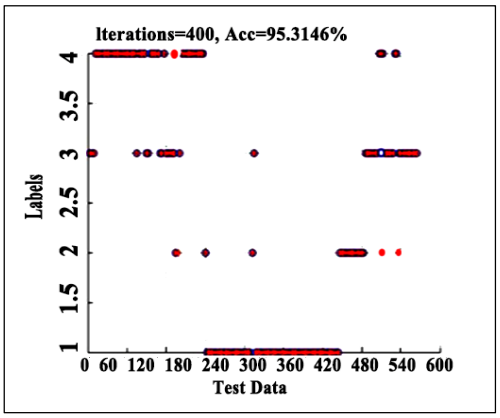
(c) Fitness changing for 200 iterations (d) Accuracy for 200 iterations (95.8275%)



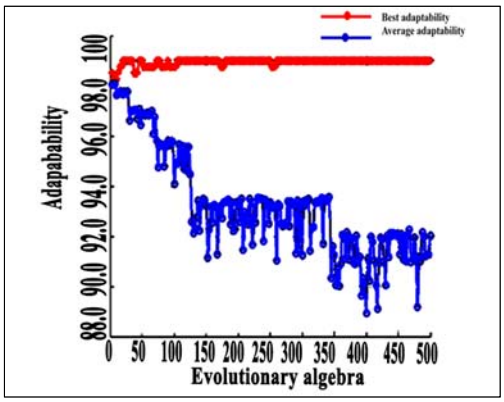


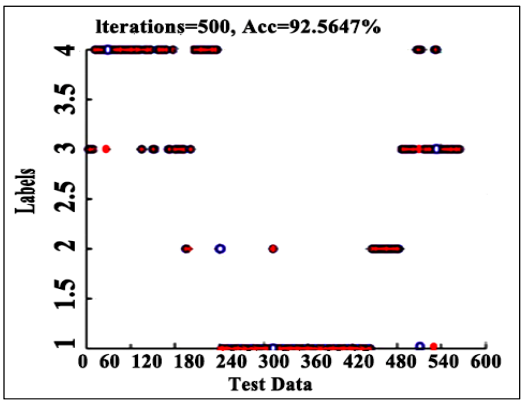
(e) Fitness changing for 300 iterations (f) Accuracy for 300 iterations (94.8543%)



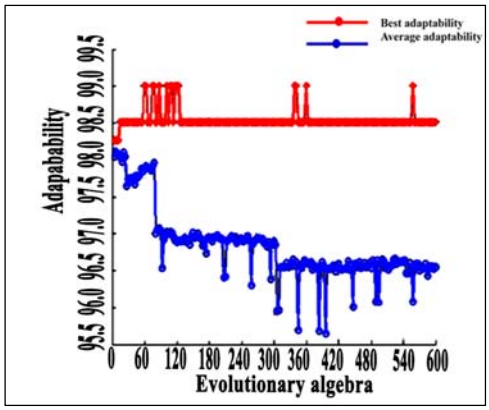


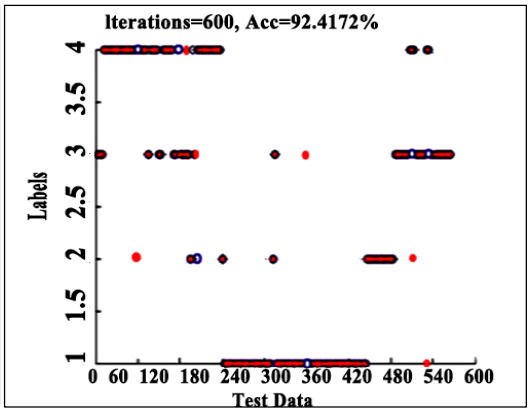
(g) Fitness changing for 400 iterations **(h) Accuracy for 400 iterations (95.3146%)**





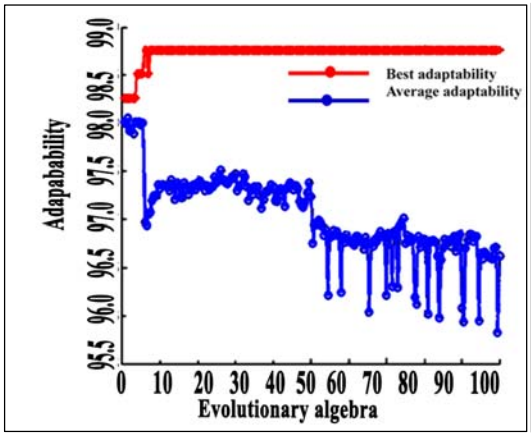
(i) Fitness changing for 500 iterations **(j) Accuracy for 500 iterations (92.5647%)**

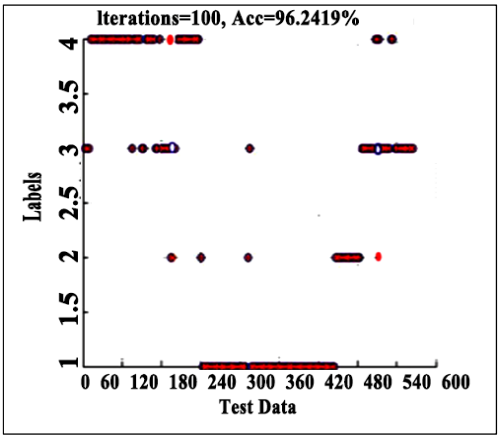




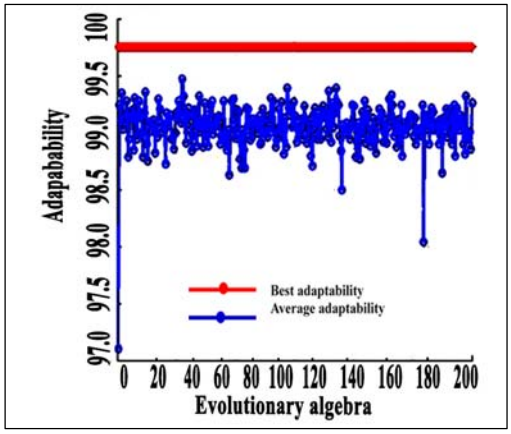
(k) Fitness changing for 600 iterations **(l) Accuracy for 600 iterations (92.4172%)**

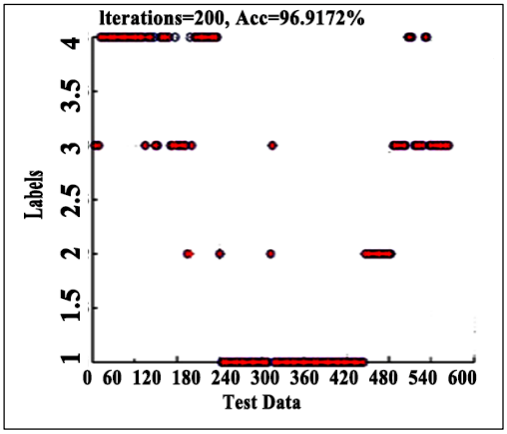
Figure 7.12: GA_SVM comparison experiments



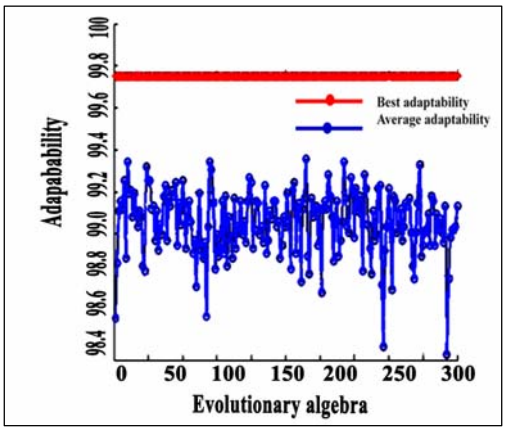


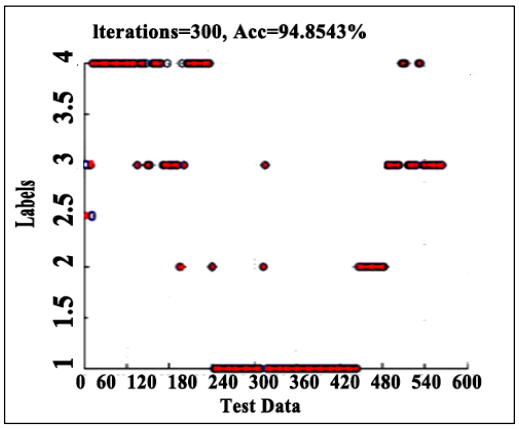
(a) Fitness changing for 100 iterations **(b) Accuracy for 100 iterations (96.2419%)**



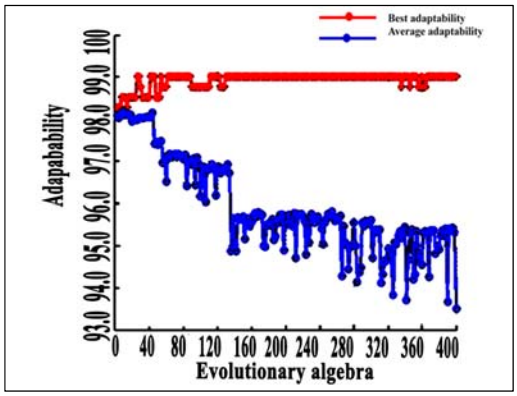


(c) Fitness changing for 200 iterations **(d) Accuracy for 200 iterations (96.9172%)**

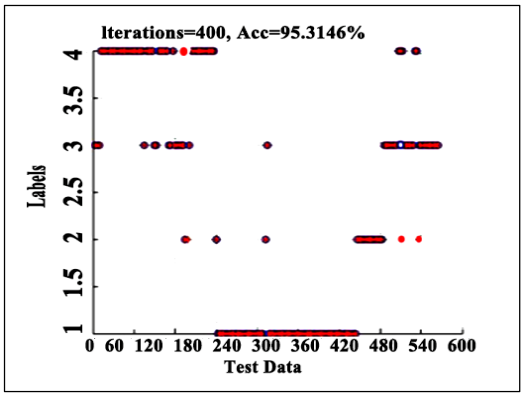




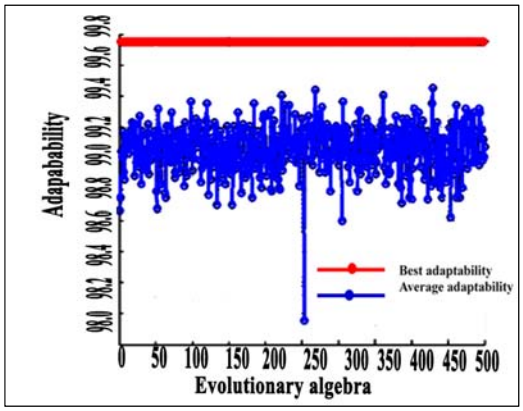
(e) Fitness changing for 300 iterations **(f) Accuracy for 300 iterations (94.8543%)**



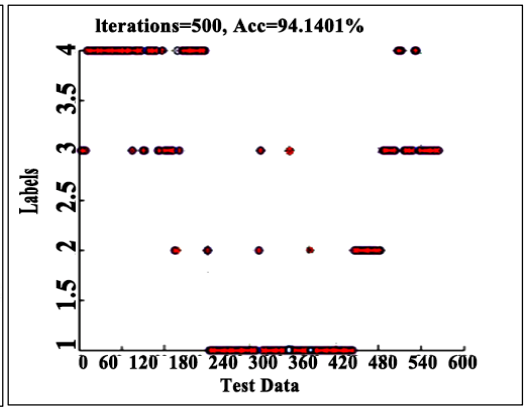
(g) Fitness changing for 400 iterations



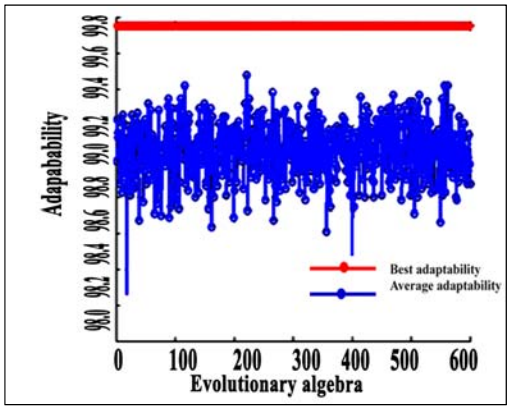
(h) Accuracy for 400 iterations (95.3146 %)



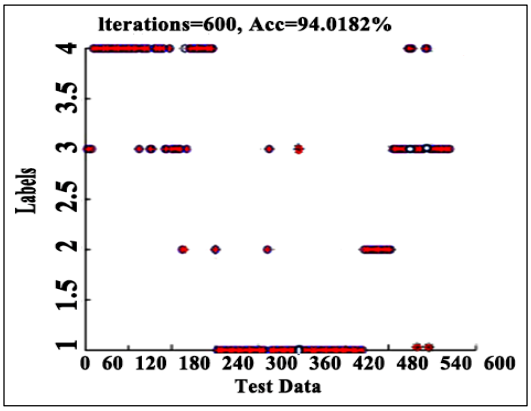
(i) Fitness changing for 500 iterations



(j) Accuracy for 500 iterations (94.1401%)



(k) Fitness changing for 600 iterations



(l) Accuracy for 600 iterations (94.0182%)

Figure 7.13: PSO_SVM comparison experiments

The performance of the methods proposed in this research is compared with other widely used machine learning techniques, which are K-means clustering algorithm, Boosting method and Decision Tree (DT). The performance measurements being used are prediction precision, accuracy, and F1_score which can be calculated through Equation (7.17) to Equation (7.21):

$$\mathrm{Pr}ecision(P) = \frac{TP}{TP + FP} \tag{7.17}$$

$$\mathrm{Re}call(R) = \frac{TP}{TP + FN} \tag{7.18}$$

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{7.19}$$

$$F1\_score = 2 \bullet \frac{\mathrm{Pr}ecision \cdot \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \tag{7.20}$$

$$FalsePositiveRate(FPR) = \frac{FP}{TN + FP} \tag{7.21}$$

The comparison results of different measurements for those methods are shown in Table 7.1. PSO_SVM method provides the highest values among all performance indicators. Then, in the second place is another heuristic algorithm optimized SVM method: GA_SVM, which is slightly lower than PSO_SVM approach. GS_SVM method can also obtain high accuracy which is close to the heuristic optimization methods. However, the high accuracy of this method is based on the sacrifices of the computation time. The rest three machine learning methods follow behind the GS_SVM method and they rank as Boosting method first, followed by K-means clustering algorithm, with the DT method being the last. These results are reasonable in considering the mechanisms of different methods. Heuristic algorithms employ more intelligent knowledge, they are therefore, capable of obtaining the global best solution with high accuracy. Boosting method assembled the advantages of all the unit regression and classification trees, therefore, it performs better than K-means clustering algorithm and DT method. Moreover, it can be seen from Table 7.1 that all those

machine learning methods can provide accuracies higher than 90%, which indicate the usability of machine learning techniques for riding quality prediction of the certain pavement based on structural condition data.

Table 7.1: Comparison between traditional methods and the proposed method in terms of precision, recall, accuracy, and F1_score

Method	**P**	**R**	**Acc**	**F1_score**
K-means clustering	86.58%	85.71%	86.215%	86.143
Boosting	87.21%	86.36%	86.78%	86.783
DT	85.97%	84.59%	84.96%	85.275
GS_SM	90.89%	90.92%	90.24%	90.402
GA_SVM	91.35%	91.21%	91.95%	91.778
PSO_SVM	***94.52%***	***95.78%***	***95.63%***	***95.155***

ROC curve is widely used to assess the overall performance of machine learning techniques in dealing with a specific problem. Therefore, the ROC curves of the compared methods are shown in Figure 7.14. It can be observed that the results agree with those obtained in Table 7.1.

To summarize, the heuristic methods optimized SVM models (GA_SVM, PSO_SVM) are more suitable than K-means clustering algorithm, Boosting algorithm and Decision Tree. Some reasons might be contributed by the intelligent properties of the heuristic algorithms that facilitated the modeling of practical engineering problem of this research.

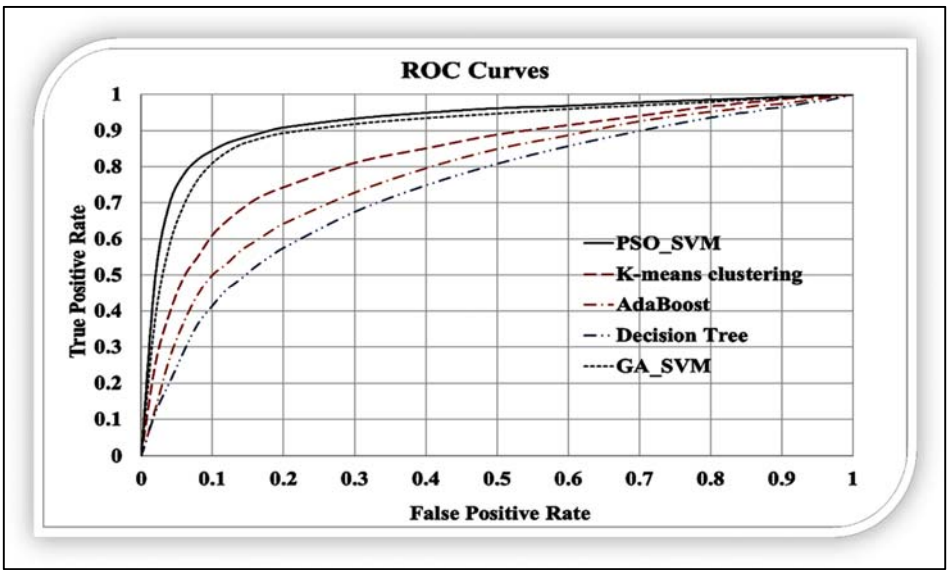


Figure 7.14: ROC curves

7.8. Chapter Summary

The research in this chapter aims to provide effective pavement structural condition assessment models using Support Vector Machines. To summarize, the surface modulus data collection, structure performance level assignment, SVM model training and optimization, and the comparative model evaluation are the key components of this chapter.

LWD pavement surface testing device is used to collect the pavement surface modulus data used in this part. The obtained pavement surface modulus data reflect the overall structural condition of the pavement layers. Therefore, four service levels are assigned based on the collected surface modulus data. The divided pavement service levels are Excellent (Level 1), Good (Level 2), Moderate (Level 3), and Poor (Level 4).

To achieve the purpose of multi-classification of pavement performance levels, pairwise SVM methods are developed thereby. Then, the two most widely used heuristic

optimization methods, which are the GA method and PSO algorithm, are adopted to select the best parameters for the SVM models. Finally, using the optimized SVM modes, several comparative evaluations are conducted. Results demonstrate that PSO optimized pairwise SVMs pavement structural condition prediction methods have the best performance.

According to the results of this chapter, it can be suggested that pavement structural condition prediction models to be integrated into the overall pavement performance assessment systems. By this way, the PMSs can provide more comprehensive, reliable, and economic suggestions for long time pavement management.

CHAPTER 8. Crack Deep Network for Box-level Pavement Crack Detection

8.1. Introduction

In recent years, pavement preventive preservation has been playing an increasingly important role in long-term pavement management. The main reason is that significant benefits have been observed in the cost-effective pavement management aspect by conducting routine preservation maintenance. Therefore, accurate, timely pavement condition investigation and performance evaluation are important procedures which can provide foundations for the maintenance decision-making. Through effective pavement surface distress detection and condition rating, any initial performance deteriorations can be observed and recorded. Therefore, both short-term and long-term maintenance operations can be planned with comprehensive considerations of different alternatives. By this way, the overall pavement life-cycle cost can be reduced, and the more pavement sections can be considered for maintenance with a limited budget.

Manual pavement condition surveys have long been criticized for their inconsistent, inaccurate, time consuming and labor-intensive characteristics. Therefore, researchers have reached out to develop effective automatic pavement condition assessment tools for several years. Meanwhile, remarkable achievements are made with the successful applications in practical pavement management work. As the name signifies, automatic pavement condition investigations are facilitated by the advancements of high-speed sensors and computerized technologies, which allow for accurate data collection and comprehensive data analysis. In the automatic road condition data collection process, high speed vehicles (Flintsch & McGhee, 2009; Gopalakrishnan & Peeta, 2010) are mounted with cameras or laser components for data collection. Those kinds of cameras or laser systems usually have

the capabilities of capturing continuous images with pavement inspection vehicles driving through the corresponding pavement. Current days, several highway management departments are turning to the utilizations of automatic pavement condition inspection, especially the more precise 3D monitoring systems (Tsai & Wang, 2015) and artificial intelligent techniques based pavement condition evaluation systems,

In traditional ways, the collected high-resolution pavement surface images should go through a series of processing procedures to obtain the pre-processed images, which should have specific size, resolution and other properties to be suitable for final feature extraction and information interpretations. The main reasons are that traditional systems do not have strong storage systems under the limitation of less advanced technologies, and the algorithms are not intelligent enough for dealing with random quality images. The algorithms embedded with those Pavement Management Systems (PMSs) are usually statistics-based methods for network-level pavement condition evaluation and decision-making. The statistical methods are simple to use and easy for an explanation, but the results are not so reliable in terms of meeting the requirement of a pavement maintenance decision making. Therefore, human post-processing is usually involved with regards to the incorrect analysis results of semi-automatic or automatic approaches. Considering this, a body of researchers have developed state-of-the-art technologies for more intelligent pavement management.

In the way of developing more intelligent pavement condition survey and performance evaluation algorithms, the state-of-the-art computer vision and machine learning based approaches have attracted the researcher's attention for quite a long time. Though some attempts have been made by scholars and industry technicians, significant limitations still exist in the accurate pavement crack detection practice. The inherent challenges result from the properties of pavement crack images, which usually contain inhomogeneous background and road way environment conditions. The intensity threshold-based

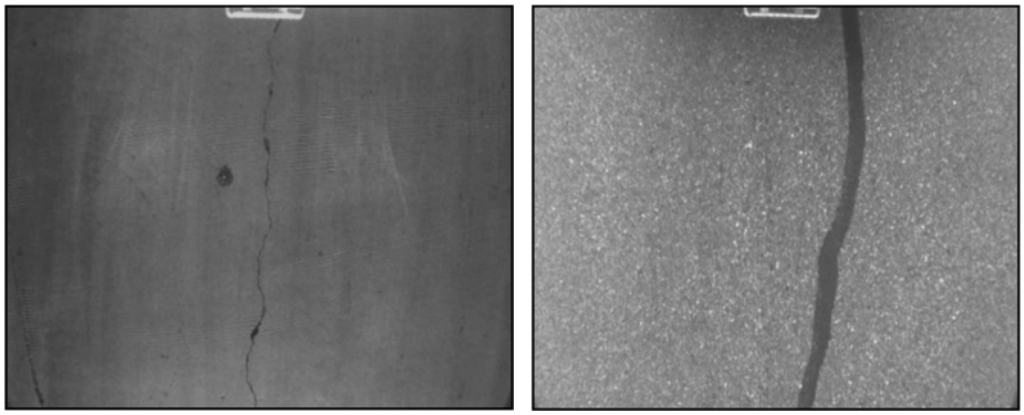
approaches, edge detector-based methods, space transformation-based algorithms, texture analysis-based methods, and machine learning based techniques are the main categories of automated pavement crack analysis methods. In the very beginning of digital image processing (DIP)-based crack analysis, threshold-based methods are the main approaches. The threshold-based methods utilize a predetermined threshold value as the dividing value between the background and the cracked regions. As a result, the regions in the image with intensity values lower than the threshold are considered as the background. Otherwise, they are assigned as cracked regions. The drawback in these kinds of methods is that they are just suitable for dealing with images with high quality, but less effective for high noise containing crack images. Considering this problem, scholars have considered two aspects of solutions. The first aspect focuses on developing optimization methods such as dynamic optimization methods (Tsai et al., 2009), or heuristic optimization methods and entropy methods (Oliveira & Correia, 2009). The other aspect of the solution is the development of new models, resulting in the continuous emergence of diverse methods. Representative work such as Zhang et al. (Zhang & Wang, 2017) proposed the crack analysis method with the integration of the region of aggregation (ROA) and the region of belief (ROB), and Chambon et al. (Chambon & Moliard, 2011) proposed the multi-scale feature extraction facilitated Markovian crack image segmentation method. The edge detector-based crack detection methods extract crack information by different filter templates, such as the most popular Canny filter, Sobel filter, and morphological filter. The transformation based approaches are represented by the discrete wavelet transformation (DWT) and discrete Haar transformation (DHT) algorithms, which can provide multi-level features (Subirats, Dumoulin, Legeay, & Barba, 2006; Wang, Li, & Gong, 2007) of the original image. By this way, the comprehensive information can be obtained for further analysis. Meanwhile, Ying et al (Ying & Salari, 2010) proposed a beamlet transformation method which aims to deal with low quality pavement crack images, which has a robust performance on linear cracks as shown by the authors.

Recently, machine learning based feature extraction and analysis approaches have gradually replaced the traditional DIP methods along with the advancement of artificial intelligence technologies. The fuzzy logic Neural Network based pavement crack detection method, which utilized the intensity difference between the cracked points and the non-crack points, has achieve significant success. This kind of fuzzy logic Neural Network crack analysis approach outperforms the traditional Bayes principle and K-nearest neighbor (K-NN) based methods (Kaseko, Lo, & Ritchie, 1994). In 2013, Oliveira and Correia (Oliveira & Correia, 2013) developed a machine learning based pavement crack analysis system called CrackIT, which is featured by its capability of multi-level crack feature extraction. However, cracks less than 2mm in width can not be detected by CrackIT. Koch's team also develop the Support Vector Machine (SVM) based road crack detection method (Koch, Georgieva, Kasireddy, Akinci, & Fieguth, 2015), which achieved satisfactory crack detection results, but the method is computational expensive. Hence, the practical applications are limited.

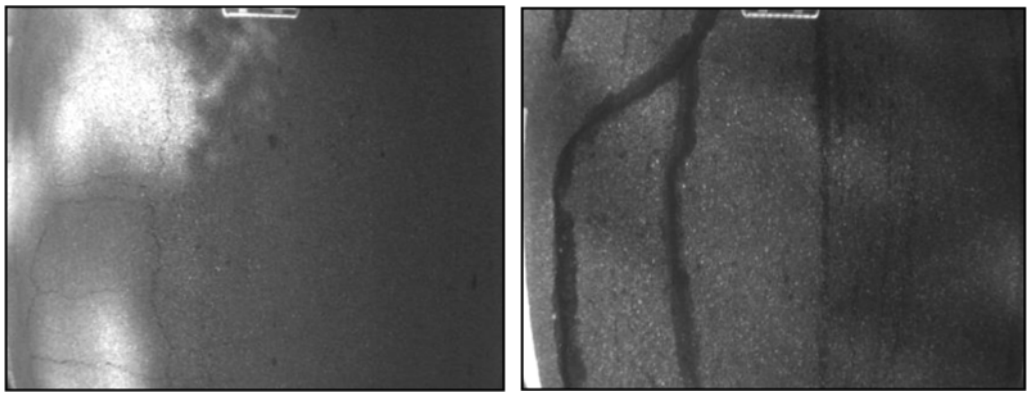
With the advanced development of computer technologies, new solutions are gradually being proposed in both theoretical and application aspects. The most advanced technology is the widely developed artificial intelligent technologies recently. In the past, deep neural networks were considered to be impossible of realization because of the limitations of the computational capabilities of the old computers. In recent years, the highly advanced Graphic Processing Unit (GPU) technology made it possible for deeper network architectures being used in practical applications. Therefore, the appropriate configuration of the hardware component is the basic requirement of a deep learning system. As a branch of machine learning techniques, the most significant characteristic of artificial intelligent technologies is the utilization of deep neural networks (DNNs) for feature extractions. In literature, several architectures of deep convolutional neural networks have been proposed for pavement surface distress detection. Remarkable implementation of DL technology for the intelligent analysis of defects of the infrastructure has been conducted by Feng's

research (Feng, Liu, Kao, & Lee, 2017). They used a novel neural network architecture called deep residual network (ResNet) to achieve active learning of the defects. The network is trained in two steps, which are 1) initial training of non-defect images filtering, and 2) second human experts' analysis-based model adjustment. By this way, Feng's method can reduce the overall data processing time and the computational resource requirement of the model architecture. Therefore, their method has been adopted by Bai' s team (Bai, 2017) to analyse pavement cracks collected by low-cost smartphones. Ground Penetrating Radar (GPR) technology (Tong, Gao, & Zhang, 2018) has been extensively used in several engineering research areas for precision defect detection and decision making, as well as in pavement management aspects. Some researchers developed novel CNN architectures for pavement crack detection and feature characterization using images acquired by the GPR system. Similarly, the effectiveness of CNNs for the detection of pavement cracks have been proved by a body of researchers in recent years.

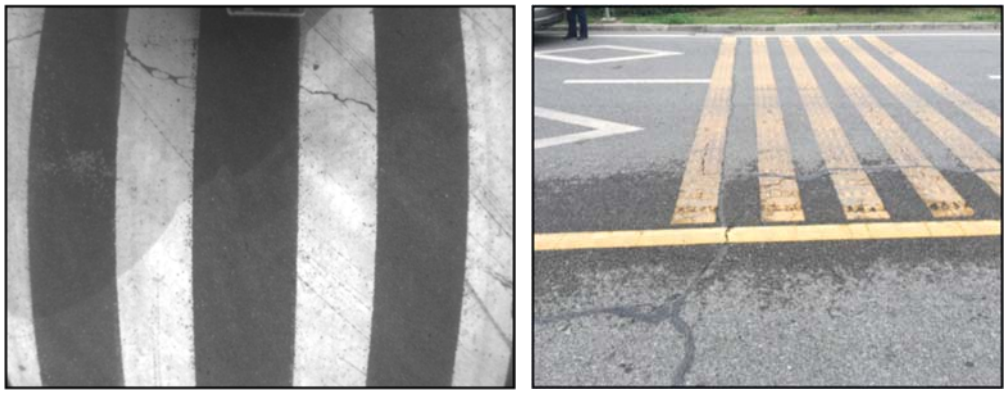
In view of the current progress of automatic pavement crack detection, no specified research has been conducted focused on the intelligent analysis of both unsealed and sealed pavement cracks with diverse pavement background conditions. The most commonly seen road background conditions in the collected images are unbalanced illuminations, makings, and the shadings, as demonstrated in Figure 8.1. In literature, most researches only focused on the analysis of normal uninfluenced pavement crack images of Figure 8.1(a). However, their methods often fail to achieve the expected results when dealing with the conditions of Figure 8.1(b)(c)(d). What's more, the sealed cracks and unsealed cracks are similar as seen in the images, especially in terms of the crack length and orientation. Thus, extensive challenges are exerted on the detection and classification of both sealed and the unsealed cracks. Therefore, this research aims to bridge the gaps in current research.



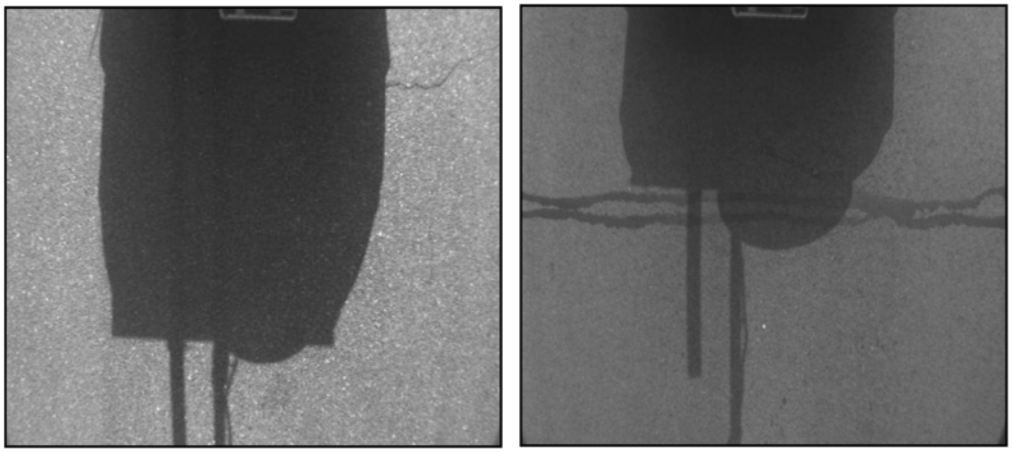
(a) Normal Background



(b) Crack and sealed crack with unbalanced illumination



(c) Crack and sealed crack with marking



(d) Crack and sealed crack with shading

Figure 8.1: Different conditions of crack and sealed crack

Based on the above research gaps found in literature, this research aims to develop DNN models for the detection of sealed and the unsealed pavement cracks initiated on diverse pavement conditions.

8.2. Deep Convolutional Neural Network

Deep Learning Techniques (DLTs) learn multi-level features of the input image through layer-by-layer feature extraction, which excludes any expert interventions required (Hinton et al., 2006) in the traditional approaches. The widely applied areas of DLTs are image feature extraction and recognition, objects detection, trend prediction, and speech recognition. Restricted Boltzman Machines (RBMs), Autoecoders (AE), Deep Belief Networks (DBNs) and Deep Convolutional Neural Networks (DCNNs) are the most widely used DLTs (Liu, Wang, Liu, Zeng, Liu, & Alsaadi, 2017). Among those approaches, DCNNs usually function for image detection, feature extraction, and objection recognition tasks (Goh, Thome, Cord, & Lim, 2014). Therefore, the architecture of this research is

developed based on the basic components of DCNNs.

Figure 8.2 shows the basic components of a CrackDN architecture for crack detection. The raw crack image without any preprocessing is used as the input. Then, the convolution layers (including activation layers), pooling layers and the fully connected layers are subsequently connected for feature learning. As a result, the detected pavement cracks can be obtained at the output of the network. The convolutional layers are realized through specifically sized convolution kernels and the activation functions. DCNNs use millions of parameters for image feature extraction, which may lead to the computational expensive and overfitting of the final model. Therefore, the pooling layers are included after the convolutional layers to maintain only high information conveyed feature values, thereby reducing the computational complexity and the probability of overfitting. The fully connected layers are not a necessity in every architecture, which is designed based on different purposes.

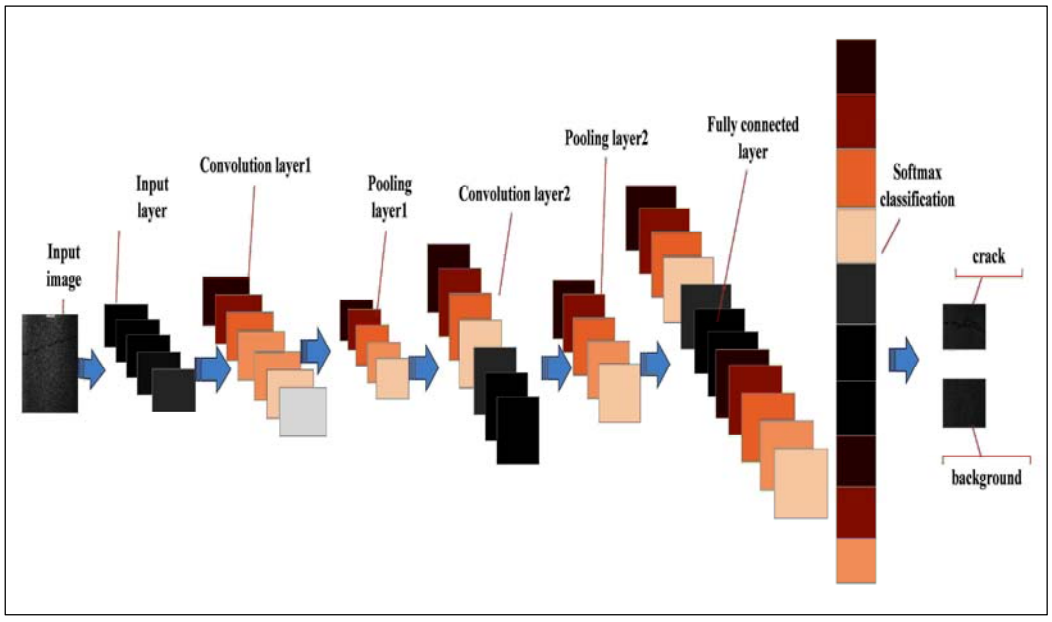
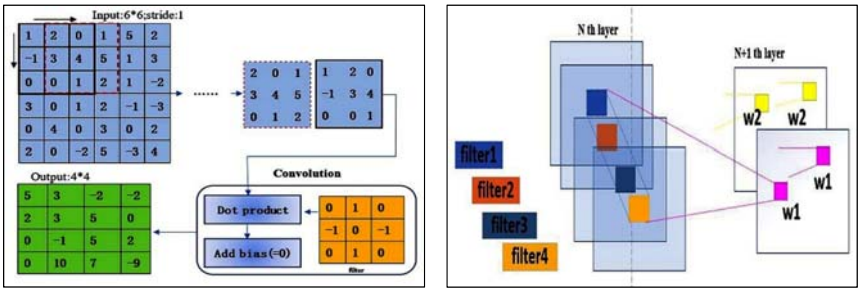


Figure 8.2: Basic structure of a deep convolutional neural network

8.2.1 Convolution

Define a series of filter with specific size, whose weights values are learnable. The convolution operation is realized by doing a dot product between the image values and the filter within the field of view. The viewing field can be considered as a window defined by the filter size, and the pixels outside of the window are considered in a certain step.

It should be noted that the convolution operation of the image data is quite different from the traditional one-dimension convolution used in signal processing. The convolution operation of the images can be considered as the multi-channel convolution operations. Figure 8.3 shows the multi-channel image convolution operation. The left side shows the convolution in the signal channel, where the filter size is 3×3 and the stride (which represents the steps for moving the convolutional filter) is set to be 1.0 in the example. Based on the signal channel convolutional shown in the left, the right side of Figure 8.3 shows the multi-channel convolution used in dealing with the image data. Specifically, the multi-channel convolution is defined by the number of convolution kernels being used. The convolution operation of each channel is conducted using the method shown in Figure 8.3(a). Finally, the multi-scale features of the input image can be obtained with the number of channels equal to the number of convolution kernels being used.



(a) Conventional convolution **(b) Multi-channel convolution**

Figure 8.3: Convolution example

8.2.2 Max-pooling

Pooling operation has two important functions, which are 1) Reducing the data size of the feature map, by which the computation complexity of the overall network can be reduced significantly; 2) Reducing the probability of the model overfitting, which means the robustness of the model can be maintained.

Max-pooling and mean-pooling are the most widely used pooling methods in the current literature. These two methods are similar in implementation, which can be explained as: Define a certain field of view in the feature map in the beginning. Then, slide the field window through the whole feature map while conducting the Max-pooling (or meaning-pooling) operations for every stop. The max and mean pooling operations are distinguished by choosing the maximum value or the mean value as the output result of that field of view. Finally, the pooled feature map can be obtained. Figure 8.4 illustrates a Max-pooling operation of 3×3 window which used 1.0-pixel stride and 0 boundary padding. Max-pooling operation is more widely used than mean-pooling because the maximum values usually convey more significant information of an image, such as the edges and boundaries of the objects. Therefore, this research also selected Max-pooling in the developed architecture.

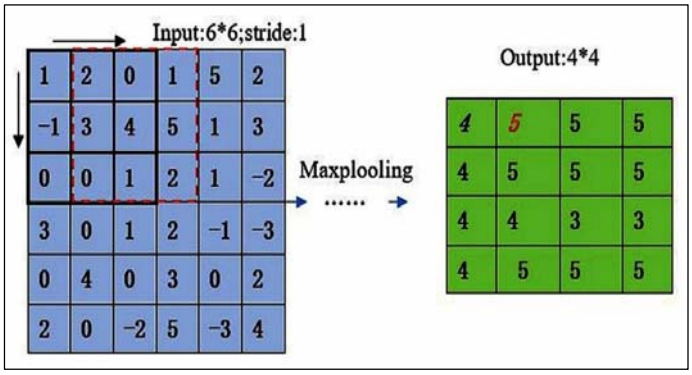


Figure 8.4: An example of Max-pooling operation

8.3. CrackDN Architecture Design

8.3.1 Crack Deep Network (CrackDN) Architecture

Remarkable researches have been achieved for the DNN based object detection in recent years. Faster-RCNN architecture can be considered as one of the most accepted highly robust and accurate image object detection methods. Therefore, the proposed CrackDN architecture is based on current Faster-RCNN architecture. Figure 8.5 shows the basic framework of the CrackDN method, which can be considered the integration of two stages. The first stage is the sensitivity detection stage, which aims to extract sensitive regions of the given image. The second stage is the region refinement stage, which refines the results of the previous stage and finishes the final classification and regression operations.

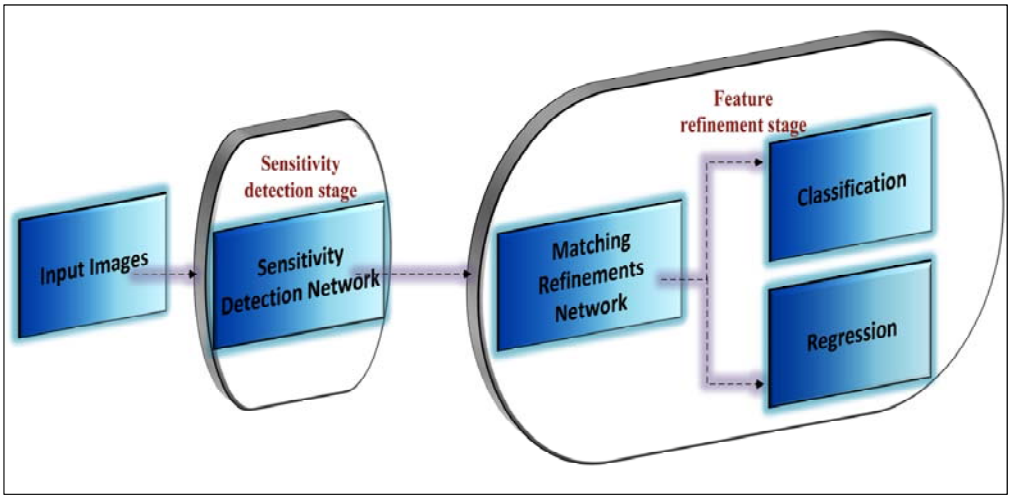


Figure 8.5: Framework of proposed CrackDN

A more detailed explanation of the CrackDN architecture can be seen from Figure 8.6. As can be seen, the feature refinement is fulfilled by a network called region proposal refinement network (RPRN). RPRN accepts three kinds of maps as the input: the basic feature maps from CNN, the region proposals and the sensitivity maps from the sensitivity

detection. The functions of some important operations are: RoI pooling operation ensures all the feature maps are revised to the same size for being used as the input of the fully connected layer. The sensitivity maps provide the crack sensitive regions to facilitate the model training procedure. The SoftMax function acts as the classifier in the last fully connected layer to obtain the final classification between the cracks and the pavement background. Meanwhile, the bounding box regression can obtain the final box region of the crack based on the initial boxes from the proposal extraction.

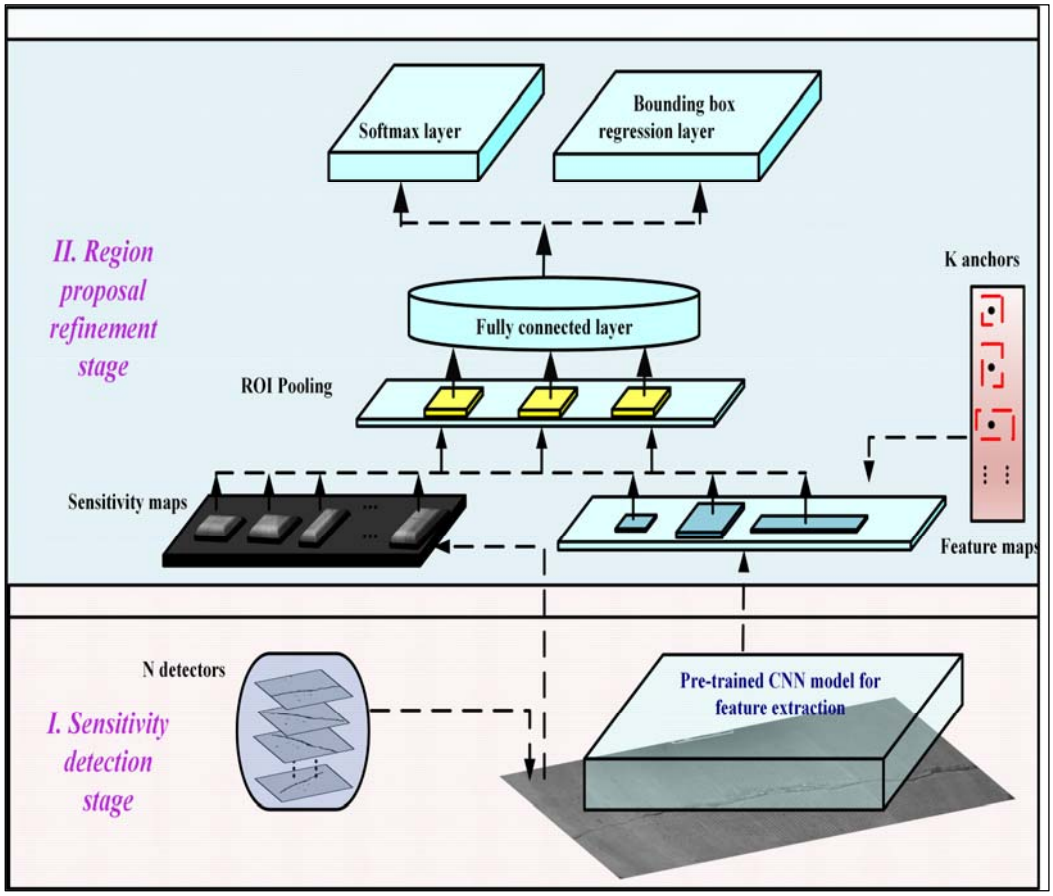


Figure 8.6: Detailed architecture of CrackDN

The proposed CrackDN architecture is trained on Ubuntu 16.04 system with NVIDIA Titan X GPU. The network is programmed using python and the architecture is set up by

TensorFlow 1.0. The weights of the convolution kernels of CNN and fully-connected layers are initialized by zero. The learning rate is used as 0.002. The weights updating momentum is set to 1.0.

8.3.2 Sensitivity Maps Extraction Network

The sensitivity detection map of this research is realized by a series of line filters formed by symmetric sigmoid curves. The mathematical function of the symmetric sigmoid curves can be expressed by Equation (8.1)

$$f(x) = 1 - \frac{1}{1 + e^{-\beta(1-\frac{|x|}{\gamma})}} \tag{8.1}$$

where x denotes the distance from between the curve profile and the centre point. β is the scaling parameter that can control the shape of the curve; and γ is a constant filter value, which is fixed to 0.5 in this research.

The most important characteristic of sensitivity detection curve is that it does not require any parameter to be learned. Therefore, the computational cost of this stage can be considered as zero compared to the convolutional operations. Moreover, the crack detection accuracy of the overall network can be improved as the crack sensitive regions are extracted through this sensitivity detection and used as references for the following refinement stage.

The intrinsic mechanism of symmetric filters is that the sharp contrast of intensity between the cracked area and the pavement background can be simulated by the symmetric curves shown in Figure 8.7. The curves tend to flatten if the viewing window is beyond $2 \times \gamma$. Hence, the width of the filter is selected as $3 \times \gamma$ and the parameter β is set to 8.

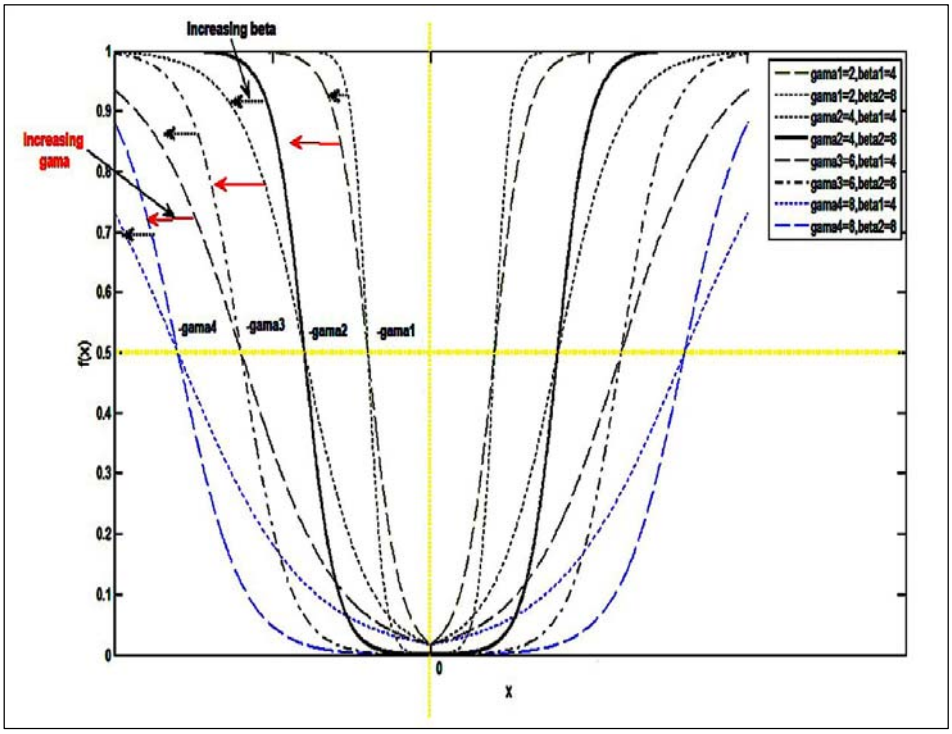


Figure 8.7: Symmetric sigmoid curves with different β and γ

Sensitivity maps act as an initial feature extraction procedure, which can be followed by ROI pooling, together with crack information obtained from Region Proposal Network. Therefore, the sensitivity maps are generated from a series of line filters formulated.

Based on the symmetric sigmoid curves defined previously, the line filters are initially generated by collecting the identical profiles along the y axis. Meanwhile, two ways are used to facilitate this procedure: 1) The filters are shifted with zero mean value, thereby the response of the curve to the noise information can be zero; 2) The same width is used for all of the line filters, by which the model computational complexity can be reduced. It should be noted that the detection accuracy cannot be influenced because the second refinement works as well. After that, the initial line filters are shifted from the orientation of 0 degrees to 170 degrees with a step of 10 degrees. Figure 8.8 shows an illustration of the line filters.

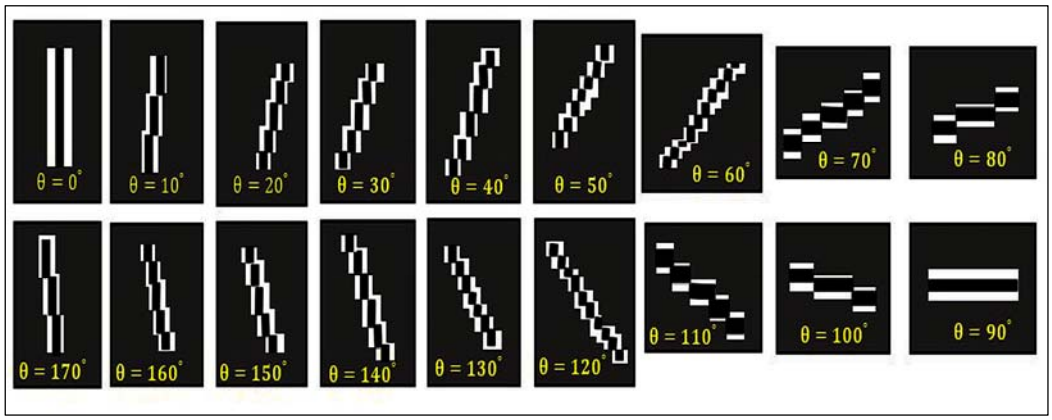


Figure 8.8: Representative line detectors used by sensitivity maps

8.3.3 Region Proposal Refinement Network (RPRN)

RPRN is conducted based on the generated anchors, sensitive maps and feature maps. The anchors are used as the detectors to measure the fitness condition between the feature maps and the sensitive maps, score the sensitive feature maps (which are called proposals) and conduct the bounding box regression for the sensitive feature maps.

For an n-anchor based RPRN detector, the operations include: Firstly, 2n scores for the fitness condition between each sensitive map and the corresponding feature map are obtained. After that, the feature maps which scored lower than 0.7 are screened out, resulting in the selected proposals. Next, two aspects of operations should be conducted, which are (1) Calculate 2n classification scores for each proposal, which represent the probabilities for both crack and the pavement background. (2) Conduct bounding box regression for the proposals using coordinates of the generated anchors. Benefited by the initial sensitivity detection and the feature sharing of CNN layers, the RPRN can obtain higher crack detection efficiency than the traditional methods.

The ReLU (Equation 8.3) is used as the activation function to ensure the accuracy of the feature fusion between the sensitivity map and the feature map. The feature fusion is

conducted as the first step of the RPRN, during which the centre of the sensitive map is used as the basis for generating the anchors with different aspect ratios. As a result, the insensitive feature maps can be eliminated from the original feature maps to form the sensitive feature maps. After that, the detected sensitive feature maps are used as the input of the fully connected layers, which consist of a total of 2n classification filters and 4n regression filters. The classification filters output the crack positions that have the highest probabilities. The regression filters output the best suitable bounding box with the highest IOU . The mechanism of anchors for box regression is illustrated in Figure 8.9.

$$IOU = \frac{\mathrm{Pr}edictedBox \cap GroundTruth}{\mathrm{Pr}edictedBox \cup GroundTruth} \tag{8.2}$$

$$\mathrm{Re}LU = \begin{cases} x & x > 0 \\ 0 & x \le 0 \end{cases} \tag{8.3}$$

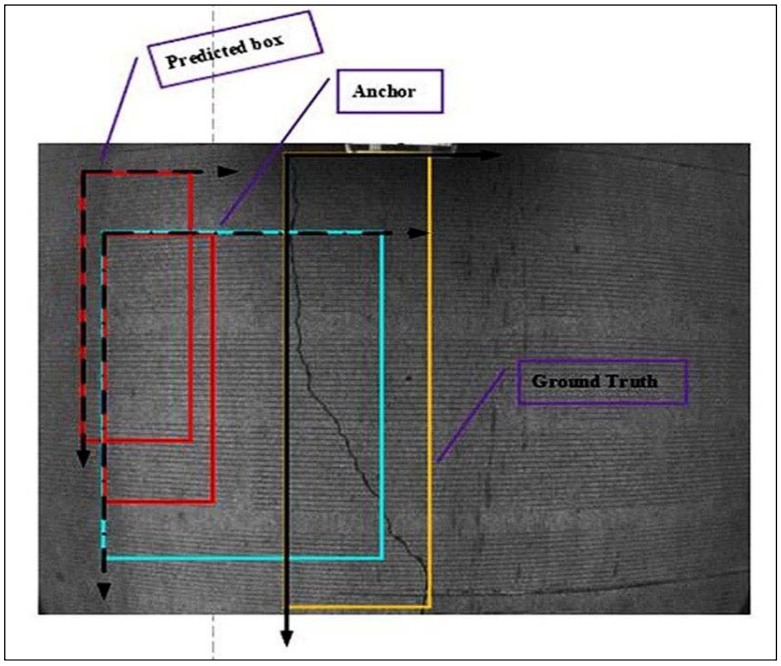


Figure 8.9: Anchor based regression mechanism

The multi-task loss function as expressed in Equation (8.4) is used to train the RPRN. It can be seen that this multi-task loss function includes both the classification loss and the regression loss, therefore, these two objectives can be optimized simultaneously.

$$L(\{p_i\},\{b_i\}) = -\frac{1}{N_C}\sum_{i=1}^{n}\log[p_i^* p_i + (1-p_i^*)(1-p_i)] + \lambda\frac{1}{N_b}R(b_i - b_i^*) \tag{8.4}$$

where i denotes the i^{th} anchor. p_i^* denotes the ground truth label. p_i denotes the probability of the i^{th} anchor. b_i^* and b_i are the coordinates of the ground truth and the detected bounding box, respectively. $R(b_i - b_i^*)$ denotes loss calculation of the bounding box regression. In this research, the $Smooth_{L1}$ function is used to calculate the box regression loss, which is defined in Equation (5):

$$Smooth_{L1} = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{8.5}$$

8.4. Crack Detection and Discussion

8.4.1 Crack Image Dataset Preparation

Since DNNs must be trained on the large dataset to obtain high performance and robust models, data collection is always regarded as one of the major procedures. Meanwhile, the model trained by high correlated images usually tends to overfit. Since the images collected by the same device usually have high correlations, this research used pavement crack images of diverse sources. Smartphones and camera mounted high-speed data collection vehicles are the main approaches of data collection. For quality insurance, only non-overlapped pavement crack images are selected from the high-speed vehicle collected

images. LTPP pavement distress protocols are being referred to when collecting various types of pavement cracks. After the initial data collection, two rounds of quality checks are conducted by trained operators. In the first round of checking, every pavement image is examined by well-trained operators to screen out the overlapped and non-crack images. In the second round of checking, the examiners try to balance the number of unsealed cracks and the sealed cracks, as well as the number of different background conditions. All this preparation work needs almost 8 months to be completed.

After data collection and initial double checking, 3000 pavement crack images with the resolution of 2000(pixel) ×1500(pixel) are prepared. Then, the raw images are cropped into 500(pixel) ×375 (pixel), resulting in 12000 images. Those images are used in the model training, validation and testing procedures. Inspired by He's research (He, Zhang, Ren, & Sun, 2014), a simplified data augmentation is conducted, which can increase the diversity of the training dataset, thereby minimized the probability of model overfitting. The data augmentation operation of this part only includes simple rotations with reference to the original images. Seven kinds of rotations are conducted as shown in Figure 8.10, by which the comprehensive pavement crack dataset is generated.

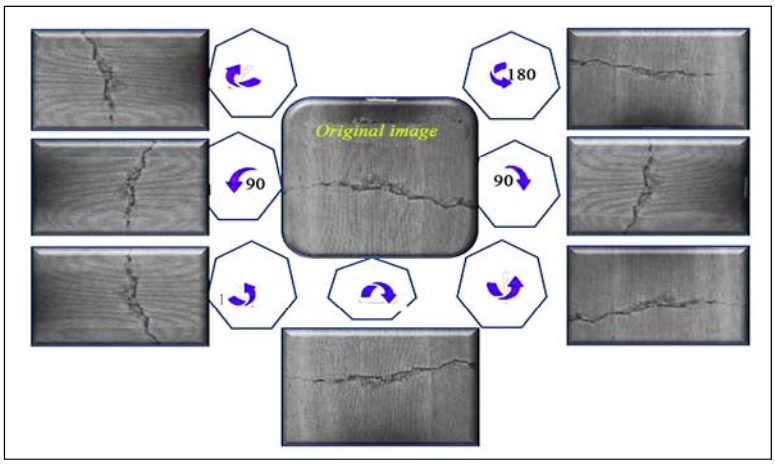


Figure 8.10: Illustration of pavement crack data augmentation

8.4.2 Experiments and Discussion

The pretrained CNN models from ImageNet are used as the feature extraction models because the performance of those models has been validated by a body of scholars. The random numbers which obey Gaussian distribution are used for model initialization. The dynamic learning rate begins from 0.001 and decays 0.1 after each epoch of training. The momentum is set to be 0.8.

For both cracks and sealed cracks images, four kinds of the most commonly seen pavement crack conditions are included in the dataset. To be specific, considering the cracks, those conditions are :1) Normal Background (NB) representing the fact that crack images are of high quality without significant influences from the environment; 2) Unbalanced Illumination (UI), which represents the images collected at daytime without artificial lighting source being used and the pavement background intensity is unbalanced; 3) Cracks on the markings (M), which represents cracks occurring on the road markings; 4) Shading (S), representing the cracks are inflenced by the shadows from the roadway. For sealed cracks, the backgroud conditions are similar to the cracks. Therefore, 8 kinds of cracks are included in the dataset. Accuracy and speed are the top two important indicators for evaluating the performance of a method. Therefore, the performance of the proposed CrackDN is compared with Faster-RCNN and SSD in terms of those two indicators. The results are shown in Table 8.1. It is noteworthy that the proposed CrackDN outperforms Faster-RCNN and SSD300 for both cracks and the sealed cracks considering mAP values. The detection speed is measured by frames per second (fps), namely the rate value as shown in the last column of Table 8.1. It can be seen that SSD300 is slightly faster than CrackDN. However, the mAP values of CrackDNs are significantly higher than the values obtained by SSD300 method. Therefore, it is reasonabled to conclude that CrackDN performs best for the detection of different kinds of cracks with diverse backgroud conditions.

Table 8.1: Performance comparative experiments

Method	***Type***	***Class***	***mAP***	***Rate(fps)***
Faster-RCNN	Crack	C-NB	0.9546	3.2
		C-UI	0.9475	2.9
		C-M	0.9179	2.5
		C-S	0.9347	2.7
	SealedCrack	SC-NB	0.9641	3.4
		SC-UI	0.9416	3.0
		SC-M	0.9217	2.9
		SC-S	0.9342	3.1
SSD300	Crack	C-NB	0.8947	6.8
		C-UI	0.8826	6.2
		C-M	0.8716	5.8
		C-S	0.8748	6.3
	SealedCrack	SC-NB	0.8997	6.8
		SC-UI	0.8946	6.3
		SC-M	0.8742	5.7
		SC-S	0.8629	6.2
CrackDN	Crack	C-NB	0.9756	6.2
		C-UI	0.9673	5.8
		C-M	0.9476	5.4
		C-S	0.9642	5.7
	SealedCrack	SC-NB	0.9878	6.4
		SC-UI	0.9736	6.0
		SC-M	0.9541	5.7
		SC-S	0.9748	5.9

During the data analysis, it has been observed that the number of proposals has a significant influence on the recall values of the model. More proposal literally can increase the recall of the model. However, the computation complexity of the model tends to increase at the same time. Therefore, the average recall values with different proposals are compared as shown in Figure 8.11.

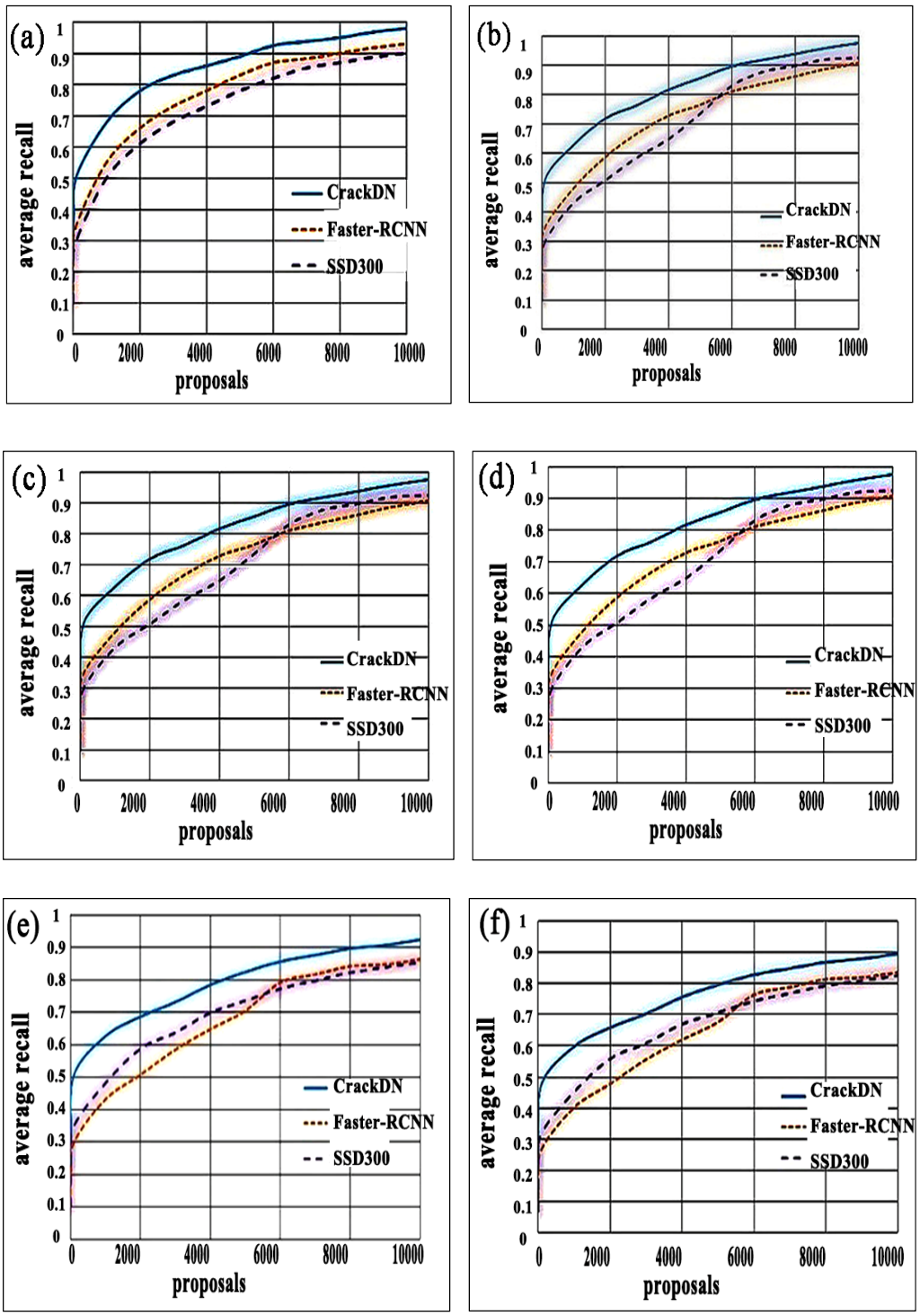
Overall, it has been found that the average recalls of sealed cracks detection are higher than cracks detection with the same pavement background condition (Figure 11(a and b) (c and d) (e and f), and (g and h)). The results are reasonable because the sealed cracks are darker than cracks in terms of intensity. Therefore, the CrackDN is more robust and flexible in dealing with the diverse conditions of practical crack images.

Faster-RCNNs can achieve a high recall value rapidly with few proposals, which seems better than SSD300s in most situations. Stepping forward, it is observed that the average recalls of SSD300s are higher than those of Faster-RCNNs when the number of anchors exceeds 5800 (see from Figure 8.11b, c, d, f, g, h). These results are benefited by the comprehensive information conveyed by massive proposals.

From another perspective, observing the results of different background conditions for cracks detection. It can be found that the detection results of cracks on markings(C-M) rank in the least place compared with the others (see Figure 8.11 f). Meanwhile, the detection of cracks on the normal background (C-NB) has the best performance (see Figure 8.11b). The performance for detection of cracks with unbalanced illumination (C-UI) and cracks on Shadings (C-S) are close to each other (see Figure 8.11 d and h). Similar results are observed for the sealed crack detection. Specifically, the SC-M (Figure 8.11e) has the worst performance. SC-NB has the best performance (see Figure 8.11a), while the detection performance of SC-UI and SC-S are close to each other (see Figure 8.11 c and g).

In conclusion, the proposed CrackDN outperforms both traditional SSD300 and Faster-

RCNN for dealing with diverse background condition of cracks and sealed cracks.



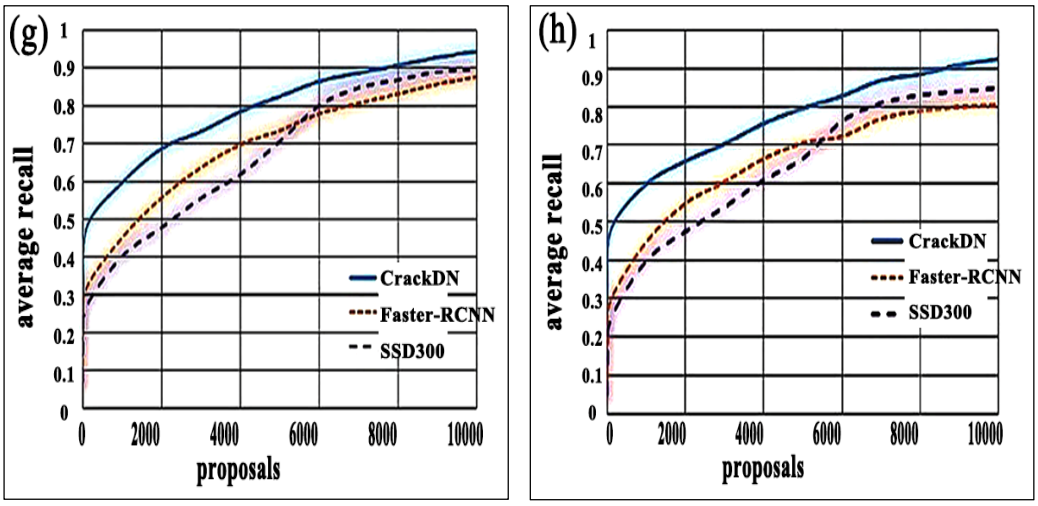
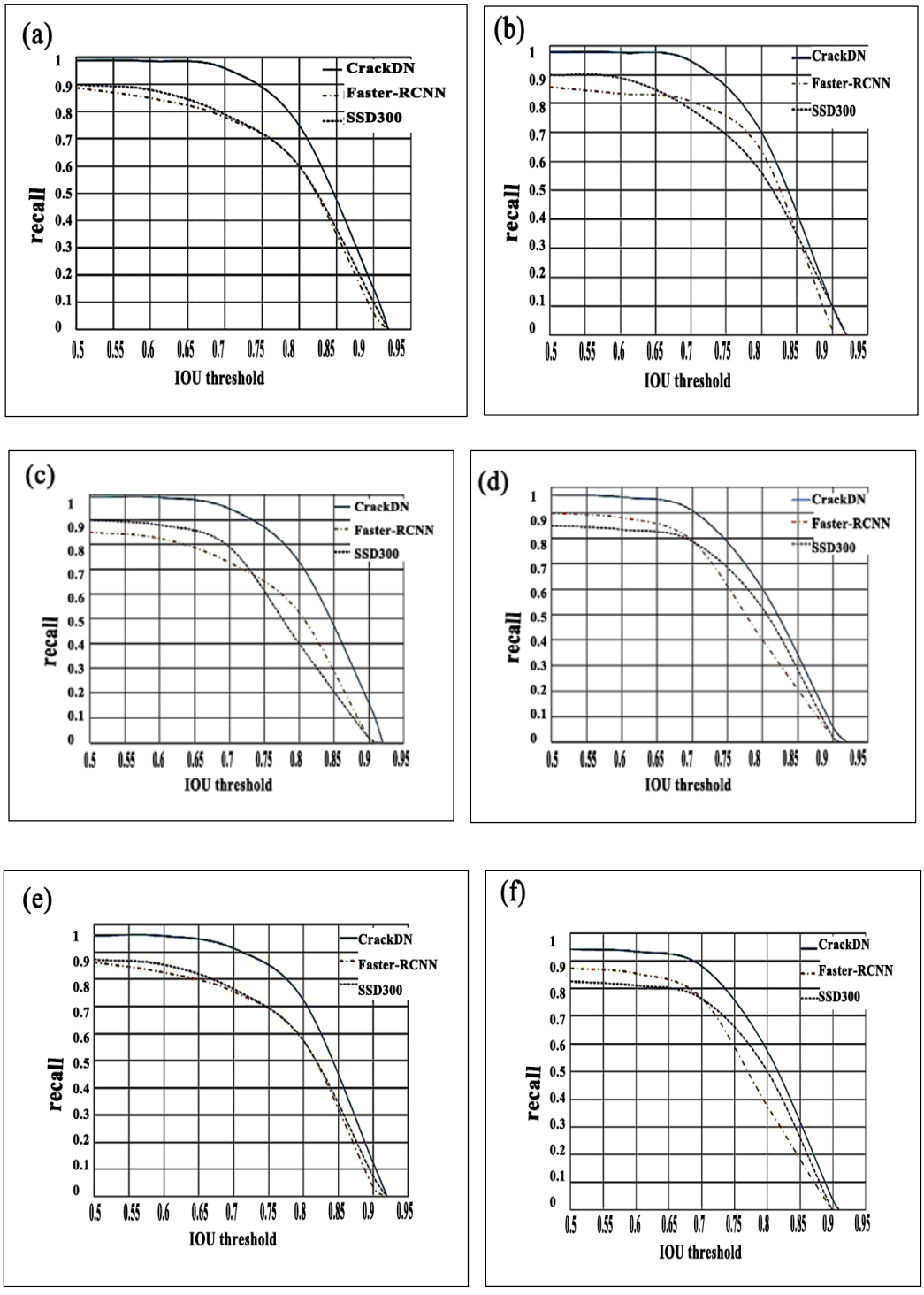


Figure 8.11: Average recalls with different number of proposals

The value of IOU usually varies from 0 to 1.0, which is decided by the intersection between the predicted box and the ground truth. A threshold value is selected to obtain the final best bounding box for the cracks. Therefore, the selection of threshold value has a strong influence on the performance of the model. Literally, the recall value of the model tends to decrease with the increase of the threshold. Considering this, a series of thresholds varying from 0.1 to 0.95 with a step of 0.05 are tested. The recall curves of the three kinds of methods for the detection of different cracks and sealed cracks are shown in Figure 8.12. As an overall conclusion, the detection results of sealed cracks are always better than cracks, which agrees with the previous results. It can be seen from the curves in Figure 8.12 that 0.7 can be regarded as the best suitable threshold value for almost all kinds of cracks and sealed cracks.

Similar conclusions can be arrived as Figure 8.11 obtained. Sealed cracks have relative high recall values than cracks for the same condition. The road markings have the strongest influence on the detection results among all those conditions considered in this research. Meanwhile, it is noteworthy that the recall value obtained from CrackDN can be higher than 0.9 for most conditions when the threshold is set to 0.7. A significant advantage of the

proposed CrackDN than the other two methods is obvious.



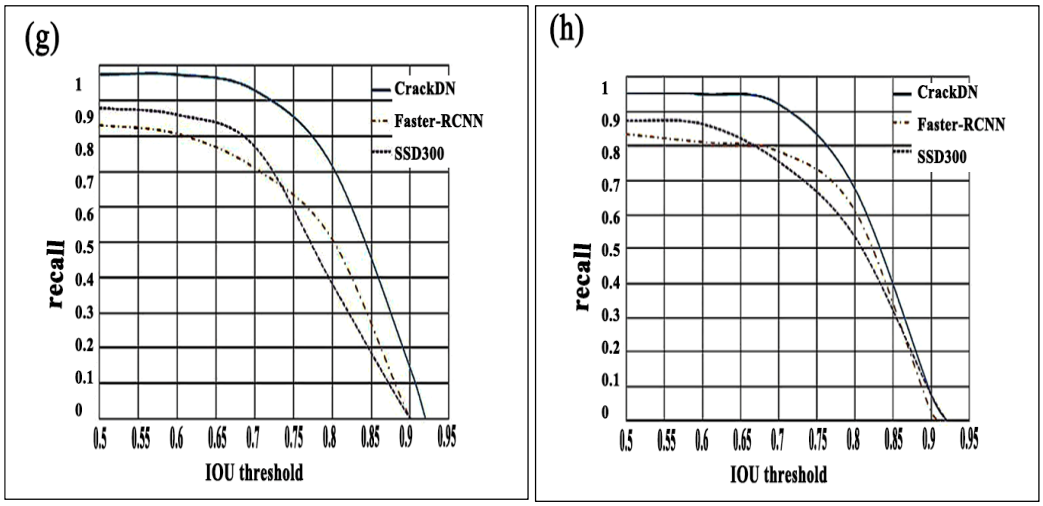


Figure 8.12: Recalls with different IOU thresholds

Figure 8.13 and 8.14 show some results of crack and sealed crack detection using the proposed CrackDN, respectively. The most important property observed is that CrackDN can detect cracks and sealed crack collected from both HMA and cement concrete pavements with diverse conditions.

Take the crack detection results in Figure 8.13 as an example: see the images in the first row, which shows the results of the cracks with the normal background (C-NB). Best crack detection results can be obtained under conditions of 1) when the data collection camera is perpendicular to the tested road surface, and 2) when the contrast between the cracked region and the road background is significant. Observing the images in the second row, which shows the detection of cracks with unbalanced illumination (C-UI), it can be found that minor cracks can also be detected with high accuracy.

The third row of Figure 8.13 shows the detection of cracks with the markings (C-M). Obviously, pavement markings significantly affect the accuracy of crack detection. Fabricated minor cracks can decrease the detection accuracy significantly which can be observed from the first two images. However, the results are still acceptable (detection precisions of 0.879 and 0.885, respectively) since the detection of minor cracks is still a

great challenge. The last row of Figure 8.13 shows the results of cracks with shadings (C-S). Both high contrast and blurred cracks can be detected by CrackDN whatever the cracks are partial or completely under shadings. To summarize, C-NB can obtain the best detection results; Followed by C-UI and C-S, which are similar and in the second and third position; finally, the detection of C-M rank in the last place.

Similarly, the images in Figure 8.14 shows the detection results of diverse sealed cracks. From the first row to the last, the results are for SC-NB in the first row, SC-UI in the second row, SC-M in the third row and SC-S in the last row. The overall results agree with those in Figure 8.13. Moreover, an interesting phenomenon being found is that CrackDN can detect the sealed cracks even when there exist unsealed cracks, which can be observed from the last image of the first row. This indicates that CrackDN can help to distinguish sealed cracks from the unsealed cracks. The accuracy for detecting SC-UI and SC-S is closed to that of C-NB, which are all higher than 0.90.

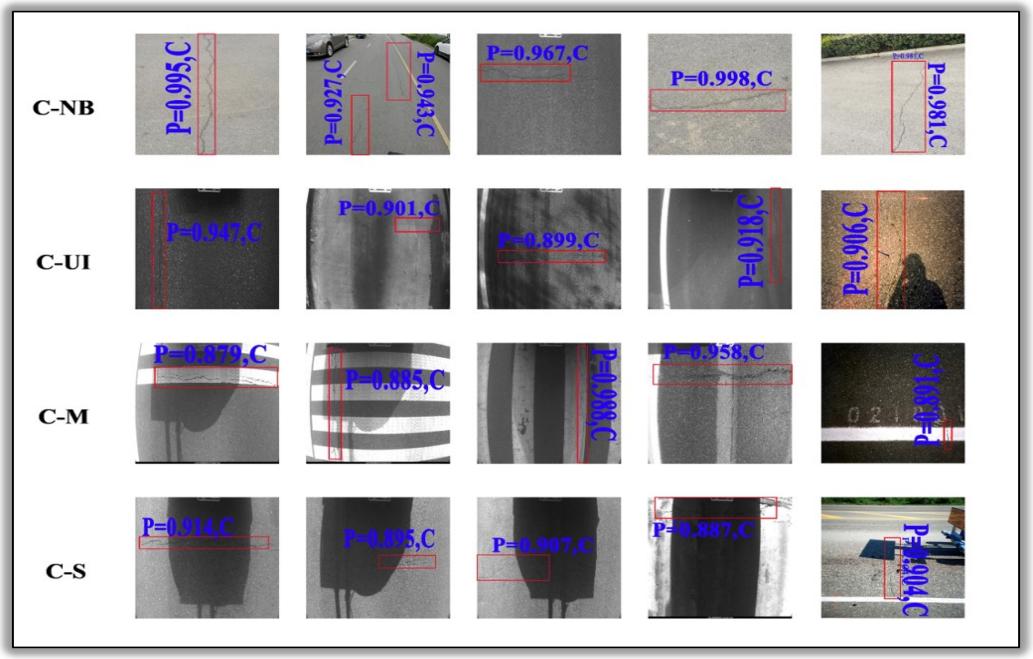


Figure 8.13: Example images of crack detection

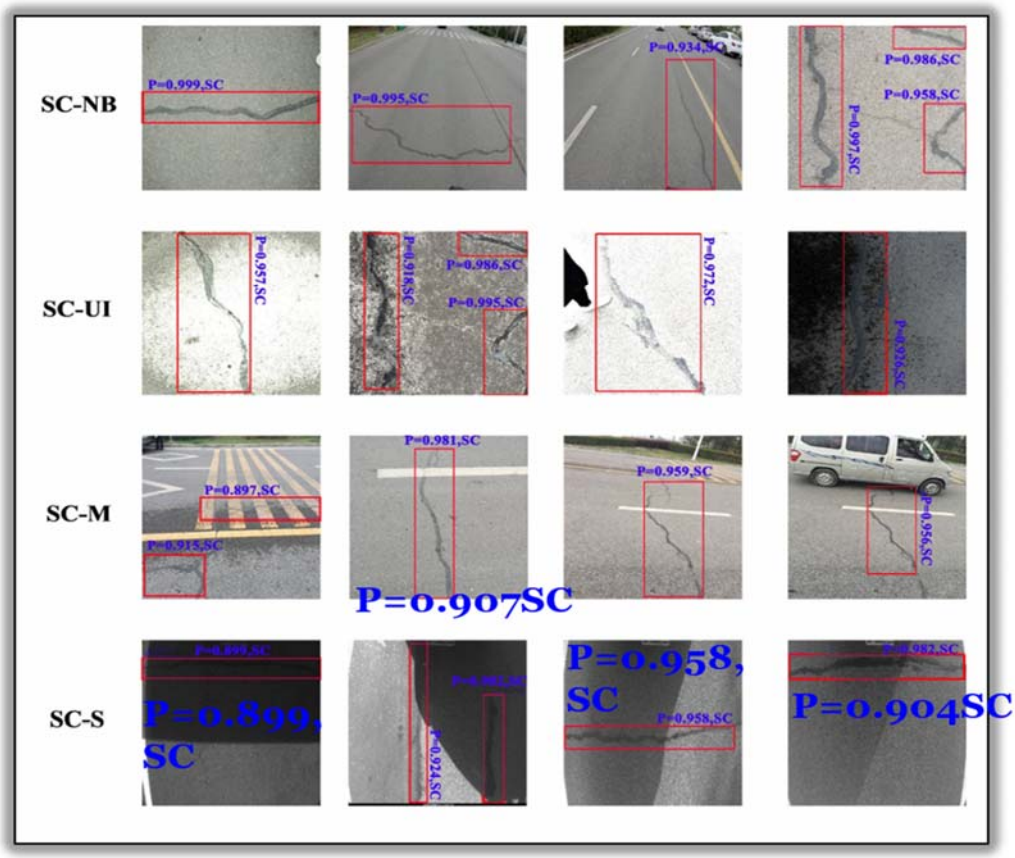


Figure 8.14: Example of images of sealed crack detection

8.5. Chapter Summary

An integrated Deep Crack Network (CrackDN) is developed in this chapter for pavement crack detection. CrackDN consists of two parts, which are sensitivity detection network and Region Proposal Refinement Network (RPRN).

The first part of the sensitivity detection network is composed of a group of line filters, which are generated by a symmetric sigmoid function. This part acts as a sensitivity detector to extract the crack sensitive regions of the input image, thereby facilitating the following region refinement stage. The second part of the feature refinement stage is

achieved by the feature matching operation and bounding box regression operation. Finally, the correct crack region is detected by the bounding boxes.

Analysis results proved that CrackDN can detect both cracks and sealed cracks that occurred on several kinds of background conditions. At the same time, CrackDN outperforms the state-of-the-art faster-RCNN and SSD300 models in terms of precision to a large extent, though slightly slower than SSD300 in detection speed.

In the future, more conditions are suggested to be considered to improve current CrackDN, thereby obtaining more robust models to be used in the automated pavement condition survey systems.

CHAPTER 9. CrackU-net for Pixel-level Crack Detection

9.1. Introduction

As has been discussed in the previous chapter, periodic detecting and recording of pavement surface cracks are tedious tasks if all implemented manually. However, these tasks are quite significant for effective pavement management and budget schedule. The robust and precise road crack detection is a critical task for automatic pavement management systems. Specially, pavement surface crack block-level location and the pixel-level segmentation are two different aspects of function. Therefore, pixel-level crack segmentation and analysis are important aspects other than block level location. However, the pixel-level road crack detection still remains a challenging task. The reason is that pavement surface images have heterogeneous intensities influenced by complex roadway environmental conditions.

In traditional specific criteria-based crack detection and analysis methods, prior knowledge about the characteristics of different crack types is required for setting the criteria. Such approaches include the histogram based (Kirschke & Velinsky, 1992), threshold based, detection operator based (Sinha & Fieguth, 2006; Yu, Jang, & Han, 2007; Santhi, Krishnamurthy, Siddharth, & Ramakrishnan, 2012; Adhikari, Moselhi, & Bagchi, 2014; Nisanth & Mathew, 2014) and the transform-based crack detection methods, which are significant criteria based achievements for crack analysis. Among those conventional approaches, transform based methods have been proved to be of the best performance, which is represented by fast Haar transform (FHT) and fast Fourier transform (FFT) methods. Though the performance of those methods can obtain acceptable crack analysis results, there still is room for improvement. Therefore, three aspects of improvements have been published in the literature, which are proposing novel global transforms (Ito, Aoki, &

Hashimoto, 2002; Yamaguchi, Nakamura, Saegusa, & Hashimoto, 2008; Li, He, Ju, & Du, 2014; Li, Zhao, Du, Ru, & Zhang, 2017), developing diverse crack specialized image filters (Huang & Xu, 2006; Zalama, Gómez‐García‐Bermejo, Medina, & Llamas, 2014) and putting forward hybrid methods based on several base methods (Nejad and Zakeri, 2011; Wu, Mokhtari, Nazef, Nam, & Yun, 2014; Chen, Su, Cao, Hsu, & Lu, 2017).

Criteria-based approaches have been embedded into traditional automatic road crack analyzers, which have proven to be fast and effective. However, the criteria- based methods are criticized for their low robustness and flexibility. The main reason is that extensive system calibration work should be conducted with regards to the variance of image resolutions, background conditions, and geographic conditions of the monitored road sections. Thus, the overall effort putting into pavement condition evaluation work is increased to a large extent. Therefore, machine learning based methods have attracted researchers' attention to develop more intelligent and feasible solutions.

In recent years, with the emergence and advancement of machine learning technologies, the artificial intelligence (AI) -based methods have shown significant potential in object detection, classification and recognition tasks (Liao, 2017) in engineering. Several machine learning based crack detection models have also been developed, such as the CrackTree (Zou, Cao, Li, Mao, & Wang, 2012) and SVM crack analysis model (O'Byrne, Schoefs, Ghosh, & Pakrashi, 2013). Recently, remarkable achievements have been observed using convolutional neural networks (CNNs) for infrastructure damage detection. Researchers have discovered that deep layer CNNs perform better that shallow layers, therefore, several multi-layer crack detection CNNs have been developed such as five-layer CNN based 3D surface defects detector proposed by Zhang et al. (Zhang & Wang, 2017), six-layer CNN road crack detection architecture from Zhang et al. (Zhang, Wang, Ji, & Li, 2016), eight-layer CNN crack block detector developed by Cha et al. (Cha, Choi, Suh, Mahmoudkhani, & Büyüköztürk, 2018), and 11-layer Native Bayes integrated CNN damage inspector

proposed by Chen et al.(Chen, Jahanshahi, Wu, & Joffe, 2017).

The capability of multi-level crack feature extraction is facilitated by the advanced developed hardware technologies. Therefore, parallel processing enables the multi-dimensional input data (Huang, Liu, & Sun, 2014; Jahanshahi & Masri, 2012; Ouyang & Xu, 2013; Jiang & Tsai, 2015; Erkal & Hajjar, 2017), such as RGB three channel image information, to be feasible, and provides multi-level output information, such as image level, block level and pixel level, as shown in Figure 9.1.

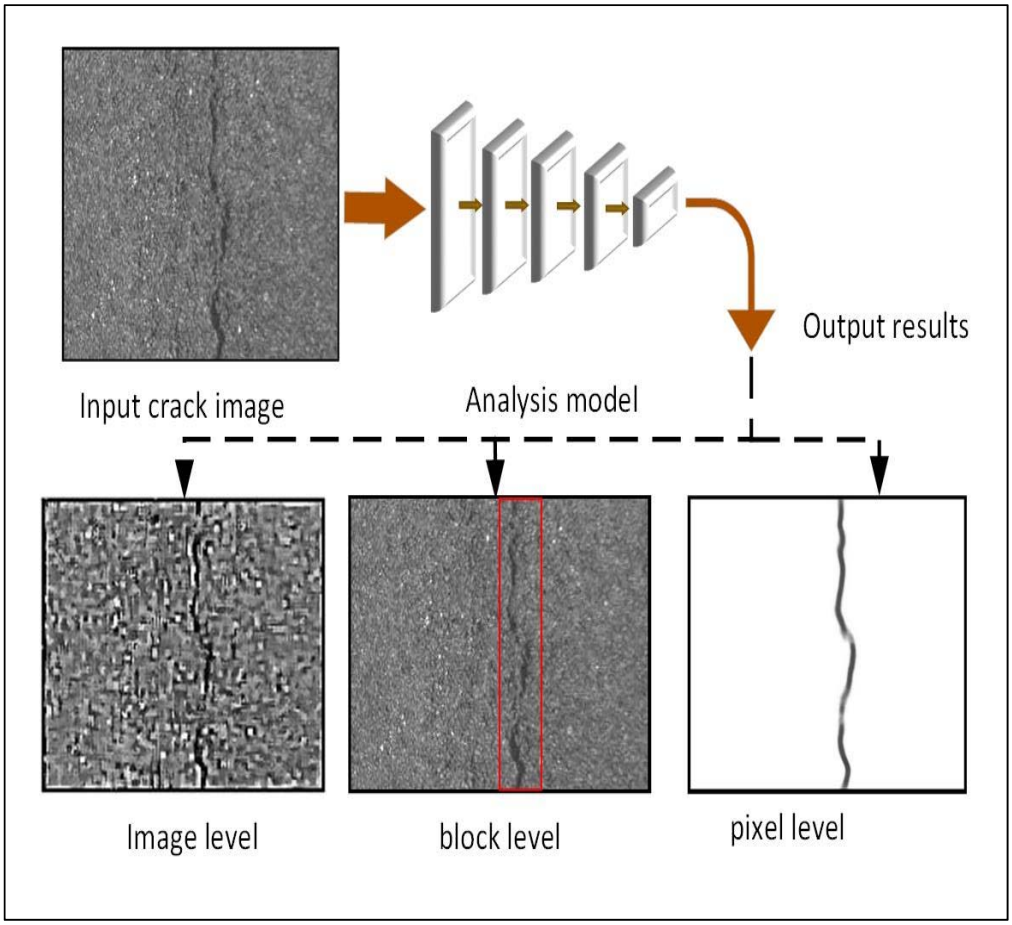


Figure 9.1：Crack detection trend with different analysis levels

In literature, most DNNs for crack analysis are focused on image level and block level.

Those achievements are significant in terms of crack counting, location, and rough road condition evaluation. Meanwhile, they can lay the foundation for the more precise pixel level crack analysis. However, DNN based pixel level crack analysis remains a challenging task. The most important capability of DNNs is that they can identify pixel-level information of the image and judge the categories it belongs to. Therefore, though more layers and parameters are used compared with traditional methods, DNNs are more suitable for big dada utilization and comprehensive information extraction, which are high required nowadays. Meanwhile, precise pixel-level judgement can be obtained.

9.2. Data Preparation

9.2.1 Data Collection

In this part, pavement surface crack data collection work has been conducted as Chapter 8. The main difference is that the pavement surface crack images used in this chapter are of higher quality than that in Chapter 8. The reason is that this chapter deals with crack image segmentations for more homogenous background crack images. Therefore, the road crack images used in this part are mainly collected by manually holding smartphones and vehicle mounted action camera for image capture. Figure 9.2 shows the photos of the data collection work and a demonstration of the collected images on the right side.

In order to train more robust crack segmentation DNN architecture, the collected original images are not used directly for model training. Instead, data augmentation has been applied to the raw crack images to generate a more comprehensive model training dataset.

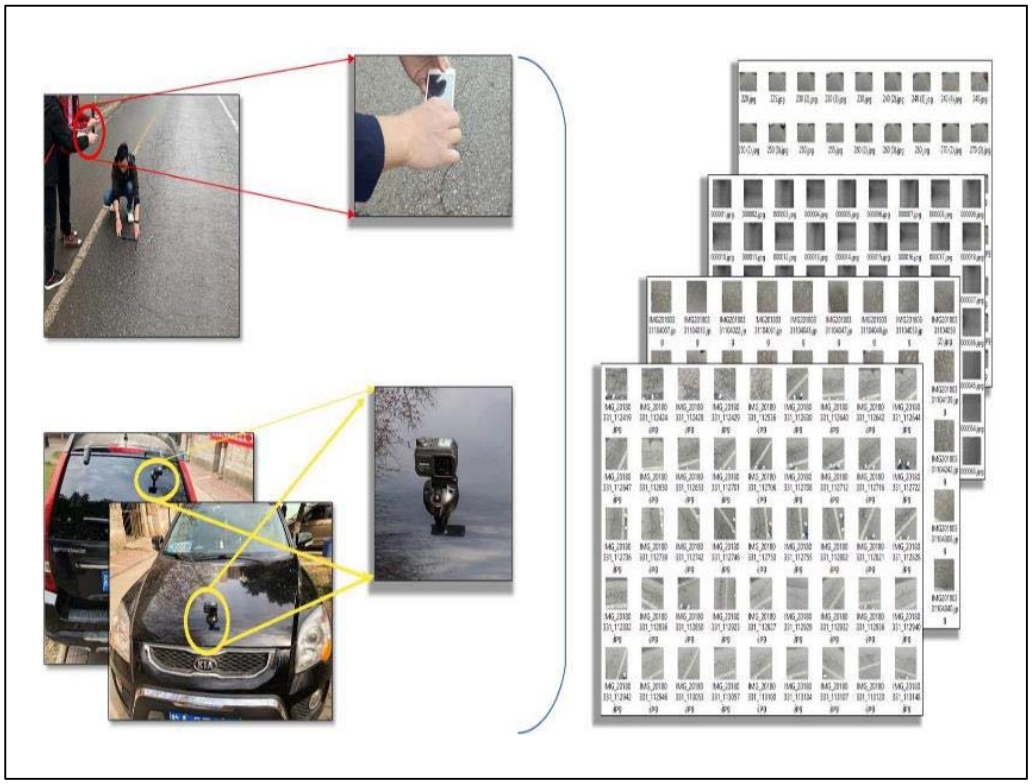


Figure 9.2: Data collection

9.2.2 Data Augmentation

The data augmentation operations are realized through two main approaches. The first approach is the crack image preparation method used in Chapter 8. The Moving Least Square (MLS) image deformation method proposed by Schaefer, et al. (Schaefer, McPhail, & Warren, 2006), which includes three transform methods, is shown in Table 9.1. These MLS-based image deformation methods generate a new image based on the original image using a transform matrix. These transform matrixes are defined in finding certain deformation mechanism between the control points of the original image and the new image.

According to the MLS based image deformation methods shown in Table 9.1, another part of data augmentation has been conducted. Figure 9.3 shows a demonstration of the data augmentation proposed in this chapter. The images in the first line of Figure 9.3 show the

affine, similarity and rigid deformations of the randomly generated image (original). The images below are the data augmentation of road crack images. As can be seen, two main procedures were included, which are MLS based image deformations and bounding fillings. The bounding of the images become dark can affect the effectiveness of the modeling training process. Therefore, the dark regions after image deformation are filled using the corresponding pavement background information. Finally, 3000 pavement crack images are obtained for model training.

Table 9.1: MLS image deformation method

Method	Transform matrix	Deformed image function	parameters
Affine MLS	$M = (\sum_i \hat{p}_i^T w_t \hat{p}_i)^{-1} \sum_j w_j \hat{p}_j^T \hat{p}_j$	$f_a(v) = (v - p_*) * M + q_*$ $= (v - p_*)(\sum_i \hat{p}_i^T w_t \hat{p}_i)^{-1} \sum_j w_j \hat{p}_j^T \hat{p}_j + q_*$	p: control point of the original image; q : control point after drag from the original image; v : pixel point of the original image $w_i = \frac{1}{\lvert p_i - v \rvert^{2a}}$:weights values M : linear transform matrix $p_* = \sum w_i p_i / \sum w_i$ $q_* = \sum w_i q_i / \sum w_i$ $\hat{p}_i = p_i - p_*$ $\hat{q}_i = q_i - q_*$
Similar MLS	$M = \frac{1}{\mu_s} \sum_i (\begin{matrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{matrix})(\hat{p}_i \quad -\hat{p}_i^{\perp})$ $\mu_s = \sum_i w_i \hat{p}_i \hat{p}_i^T$	$f_s(v) = \sum_i \hat{q}_i (\frac{1}{\mu_s} A_i) + q_i$ $A_i = w_i (\begin{matrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{matrix})(\begin{matrix} v - p_* \\ -(v - p_*)^{\perp} \end{matrix})^T$	
Rigid MLS	No transform involved, only constraint: $MM^T = I$	$f_r(v) = \lvert v - p_* \rvert \frac{\vec{f}_r(v)}{\lvert \vec{f}_r(v) \rvert} + q_*$ $where, \vec{f}_r(v) = \sum_i \hat{q}_i A_i$	

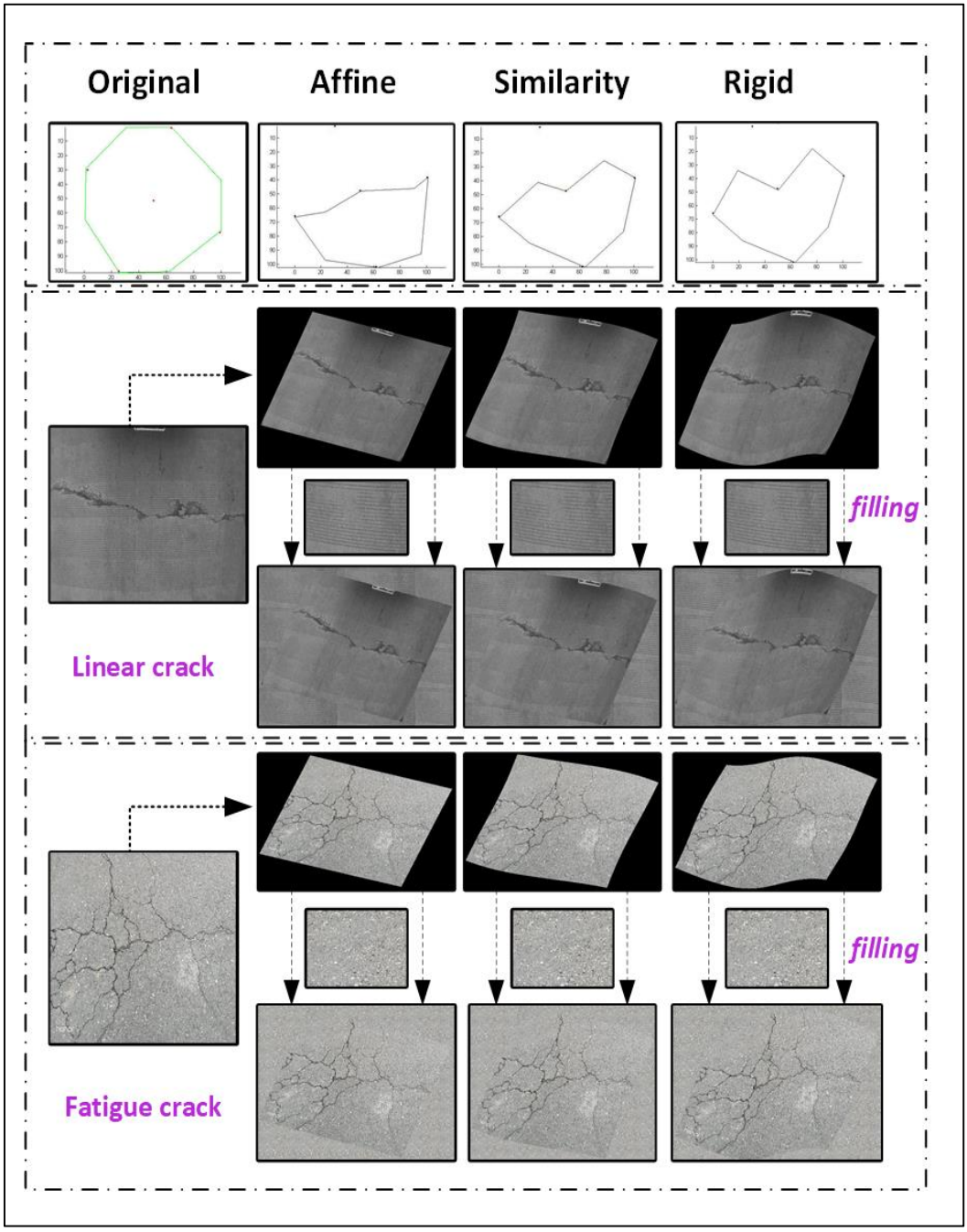


Figure 9.3: Data augmentation

9.3. CrackU-net Architecture Design

9.3.1 CrackU-net Model Structure

As has been discussed previously, pavement surface image data collection is usually costly, which caused limitations in obtaining large model training dataset, even though data augmentations are applied to the raw dataset to create more comprehensive data. Olaf Ronneberger (Ronneberger et al. 2015), together with his research team members, proposed a state-of-the-art deep neural network architecture called U-net to solve the problem of large data requirement in most network architectures.

U-net is a network structure evolved from full convolution network (FCN) (Long, et al. 2015) without full convolutional layers as that in the traditional models. Therefore, it can deal with arbitrary sized images and performs well on small datasets. The most significant feature of U-net is the integration of two directions of image processing, which are the shrinkage and the expansion network layers. The shrinking direction is composed by convolutional and pooling layers for feature extraction as tradition architecture does, while the expansion network layers merge the multi-level features obtained from different shrinking layers in combination with the up-convolution operations. Moreover, the training resource requirement of U-net is much less, which is benefited from its simpler structure and fewer parameters compared with the other networks. Therefore, this research proposed a network for road crack segmentation based on U-net called CrackU-net.

The basic structure of CrackU-net is shown in Figure 9.4.

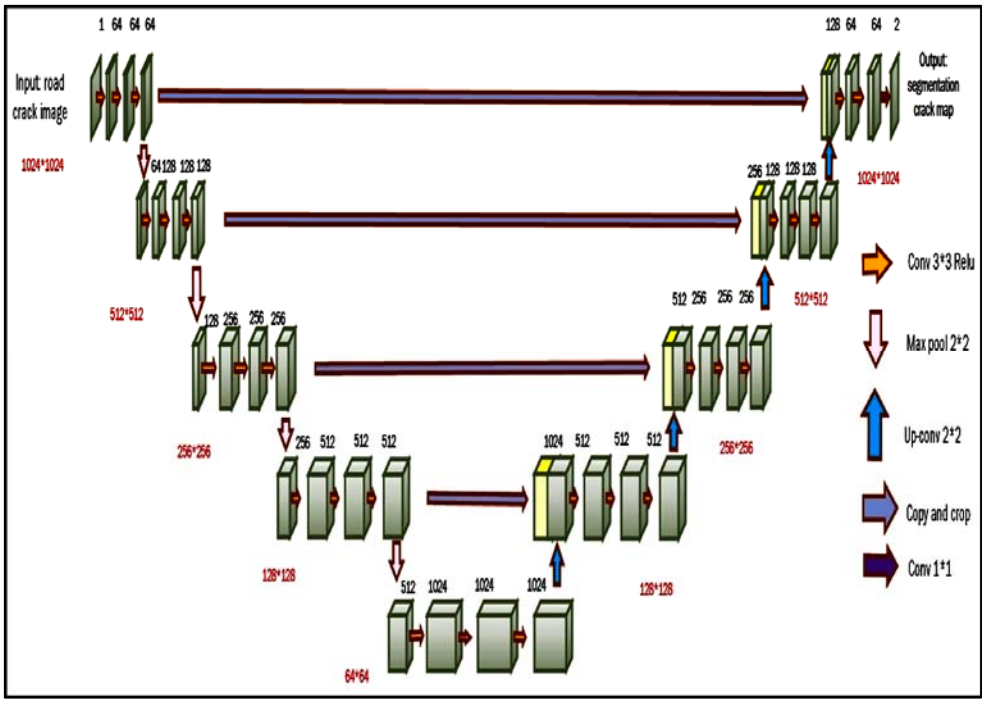


Figure 9.4: CrackU-net architecture

9.3.2 CrackU-net Network Details

Table 9.2 explains sizes of the feature maps changing of the CrackU-net. It can be seen from this table that CrackU-net include four convolutional layers. The sizes of the feature map changing can be observed from Table 9.2 combing with the information provided in Figure 9.4. It can be seen that the sizes of feature maps do not change after each convolutional operation. Only the Max-pooling operations change the size of the feature maps. This is the significant difference between CrackU-net and U-net. In U-net, the size of feature maps reduces gradually by both convolutional and pooling operations. This kind of network design brings more computation time const during the model training, because the size storage should be updated after each operation. However, for CrackU-net, the size of the feature maps is the same as the input image after convolutional operation but reduces to the haft of the input size after Max-pooling operation. Therefore, CrackU-net is easier to

train compared with traditional U-net.

Table 9.2: Size of feature maps in CrackU-net

Layer name	Input size	Output size
Max-pooling 1	1024*1024	512*512
Max-pooling 2	512*512	256*256
Max-pooling 3	256*256	128*128
Max-pooling 4	128*128	64*64
Up-conv. 1	64*64	128*128
Up-conv. 2	128*128	256*256
Up-conv. 3	256*256	512*512
Up-conv. 4	512*512	1024*1024

9.3.3 Normalized Loss Function

Loss function, also called cost function or objective function, is developed to measure the difference between the ground truth value and the output value from the network. On this basis, the network can be trained using specific optimization algorithm with the purpose of minimizing the defined loss function. When the global minimized loss value is obtained from the modeling training procedure, the best model matrices, or the best network model, can be obtained.

The most traditional loss function is formulated by calculating the average square error between the ground truth value and the output value, which is called the mean square error (MSE). Considering a single neuron node network, the MSE loss function of this simplest model can be expressed by Equation (9.1)

$$L_C = \frac{(f(W \bullet X + b) - y_{gt})^2}{2} \tag{9.1}$$

where f denotes the activation function of the neuron node. W and b denote the weight matrix and the bias value. That means $f(W \bullet X + b)$ is the actual output value of this neural network and y_{gt} is the ground truth, or the expected output value.

It can be observed that the value of the loss function is based on the difference of actual output and the expected output. The larger the difference between these two values, the higher the calculated loss. Therefore, the model training aims to minimize the loss by updating the weights of the network until it arrives at an acceptable level. The derivate of the loss function with respect to weights and the bias should be calculated to achieve this purpose, which can be expressed as

$$\frac{\partial L_C}{\partial W} = (f(W \bullet X + b) - y_{gt})\sigma'(f)X^T \tag{9.2}$$

$$\frac{\partial L_C}{\partial b} = (f(W \bullet X + b) - y_{gt})\sigma'(f) \tag{9.3}$$

It can be seen from Equation (9.2) and (9.3) that the gradient of the activation function ($\sigma'(f)$) should be calculated successively throughout the model training process. However, in the most widely used activation functions, such as sigmoid function, the $\sigma'(f)$ tends to vanish with the increase of training iterations. This phenomenon leads to less efficient weights updating.

In order to solve the gradient vanishing problem of traditional activation function, the cross-entropy loss calculation method is introduced. Considering a classification problem, the cross-entropy loss function can be expressed as:

$$L = -\sum_{i=1}^{N} y_{gt}^{i} \log y_i i \tag{9.4}$$

where y_i and y_{gt}^{j} denote the actual output and the expected output of the neural network for the category $i(i = 1, 2, ...N)$. The derivate of the above loss function can be calculated as:

$$L^{'} = \frac{1}{N}\sum (y_N - y_{gt}^{N}) = \frac{1}{N}\sum (\sigma(z_N) - y_{gt}^{N}) \tag{9.5}$$

Therefore, there is no derivative calculation item being included in Equation (9.5). Thus, gradient vanishing phenomenon is avoided effectively. That means the weight updating step can be large if the difference is high, otherwise smaller steps can be used to update the weights. By this way, the global optimal weights values can be obtained in the final model.

Assuming the training dataset of a two-category classification problem which is denoted as: $[(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)]$, the cross-entropy loss function can be expressed as:

$$J(\theta) = -\frac{1}{N}[\sum_{i=1}^{m} y_i \log h_\theta(x_i) + (1 - y_i)\log(1 - h_\theta(x_i))] \tag{9.6}$$

In the general condition of the multi-category classification problem, the above loss function can be revised as:

$$J(\theta) = -\frac{1}{N}[\sum_{i=1}^{N}\sum_{j=1}^{M} 1\{y_i = j\} \log \frac{e^{\theta_{j=1}^{M} x_i}}{\sum_{l=1}^{M} e^{\theta_{l=1}^{M} x_i}}] \tag{9.7}$$

where $1\{\bullet\}$ is an indication function, which means that $1\{true\} = 1$ and $1\{\text{false}\} = 0$.

The normalization term should be included into the loss function to balance the model

complexity and the loss level. The most widely used normalization methods are Lasso Regularization (L1 normalization) and Ridge Regularization (L2 normalization). In comparison, the L2 normalization method can avoid the overfitting problem. Therefore, this research selected L2 normalization approach. Then, the loss function with L2 normalization can be expressed as:

$$L = L_0 + \frac{\lambda}{2N}\sum_w w^2 \tag{9.8}$$

where L_0 denotes the original loss function. w denotes the weights parameter. L denotes the revised loss function. N is the number of training data. λ is the normalization parameter. The derivate of Equation (9.8) with respect to the weight w can be calculated as:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial w} + \frac{\lambda}{N} w \tag{9.9}$$

Then, the weights updating function can be expressed as:

$$w \to w - \eta \frac{\partial L_0}{\partial w} - \frac{\eta\lambda}{N} w = (1 - \frac{\eta\lambda}{N}) w - \frac{\partial L_0}{\partial w} \tag{9.10}$$

where η denotes the learning rate. $1 - \frac{\eta\lambda}{N}$ is a parameter smaller than 1, which is used to reduce the values of the weights.

9.3.4 Adam Optimization Algorithm

In the artificial intelligence-based methods, the primary purpose of most tasks are to find the best weight parameters for the constructed architecture. In the beginning, the model parameters are randomly initialized according to certain principles. Then, these parameters are updated during the model training procedure using a specific method. The most widely

used method is Gradient Descent (GD) algorithm, which is an effective iteration method suitable for solving the unconstrained optimization problems.

Considering a linear regression problem, the model can be expressed as:

$$y_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{9.11}$$

where θ_i represents the weight parameters. The above function can be simplified as:

$$y(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x_1 \tag{9.12}$$

Then, the cross-entropy loss can be calculated as:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (y_\theta(x_i) - y_{gt}^i)^2 \tag{9.13}$$

The purpose of model training is to find the best suitable θ_i, with which the calculated $J(\theta)$ can be the minimum value. The GD method for weights updating is achieved by Equation (9.14)

$$\theta_j := \theta_j + \eta \frac{\partial}{\partial \theta_j} J(\theta) \tag{9.14}$$

The derivate of $J(\theta)$ with respect to θ can be calculated as:

$$\begin{aligned}\frac{\partial}{\partial\theta_j}J(\theta)&=\frac{\partial}{\partial\theta_j}\frac{1}{2}(y_\theta(x)-y)^2\\&=2\bullet\frac{1}{2}(y_\theta(x)-y)\bullet\frac{\partial}{\partial\theta_j}(y_\theta(x)-y))\\&=(y_\theta(x)-y)\bullet\frac{\partial}{\partial\theta_j}(\sum_{i=0}^{n}\theta_i x_i-y)\\&=(y_\theta(x)-y)x_j\end{aligned}\tag{9.15}$$

Then the weights can be updated as:

$$\theta_j:=\theta_j+\eta(y_i-y_\theta(x_i))x_j^i \tag{9.16}$$

However, two significant drawbacks of GD have been observed in actual cases. One is that model training tends to be of low efficiency since each iteration deals with all of the data. Another problem is that local optimum results are usually obtained instead of global optimum values. Therefore, the stochastic gradient descent (SGD) method is developed to solve the above problems. The key improvement of the SGD method is that it uses a proportion of the data in each iteration of the model training process.

Therefore, the dataset is divided into several batches in the SGD method. Each batch of weights is updated simultaneously as:

$$\theta_j:=\theta_j+\eta\sum_{i=1}^{m}(y_i-y_\theta(x_i))x_j^i \tag{9.17}$$

where m denotes the batch size which uses all the training data.

The SGD method has been proved to be efficient in many optimization problems. However, researchers noticed that the performance of the SGD method is highly dependent on the quality of the original dataset. If the original dataset contains too much noise information, wrong results tend to be obtained. This phenomenon is caused by the random batch division

operation. Therefore, several approaches have been proposed to improve the traditional SGB method. Those improvement methods include the momentum facilitated SGD, AdaGrade and RMSProp method. In 2015, the adaptive moment estimation (Adam) optimization is proposed by Kingma, et al. (Kingma & Ba, 2014). Adam can be considered as a combination of AdaGrad and RMSProp methods. The most significant characteristic of Adam method differing from the traditional ways is its ability of adaptive setting of independent learning rates for the first and the second moment estimates of the parameters based on the properties of the learning dataset.

Specially, the weights updating function of Adam algorithm can be expressed as：

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{9.18}$$

where $\hat{v}_t$ denotes the square variance of the second moment. $\hat{m}_t$ denotes the average of the gradient in the first moment. η is the learning rate. ε is a minor constant value (equals to 0.0001 in this research).

Assume m samples have been selected from the training dataset. The samples with the corresponding labels can be expressed as: $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_m, y_m)\}$.

Adam optimization algorithm outperforms the other approaches in terms of both model convergence speed and effectiveness. This method has been proved to be of a brilliant performance in dealing with different problems in literature. Therefore, it has become the most widely used model optimization method.

The Adam algorithm can be explained using the following pseudocode:

Adam Algorithm
$f(x_i;\theta)$: the loss function; θ: the parameter to be optimized; s : the first moment variable; r : the second moment variable; β_1: the exponential decay rate of the first moment; β_2: the exponential decay rate of the second moment;

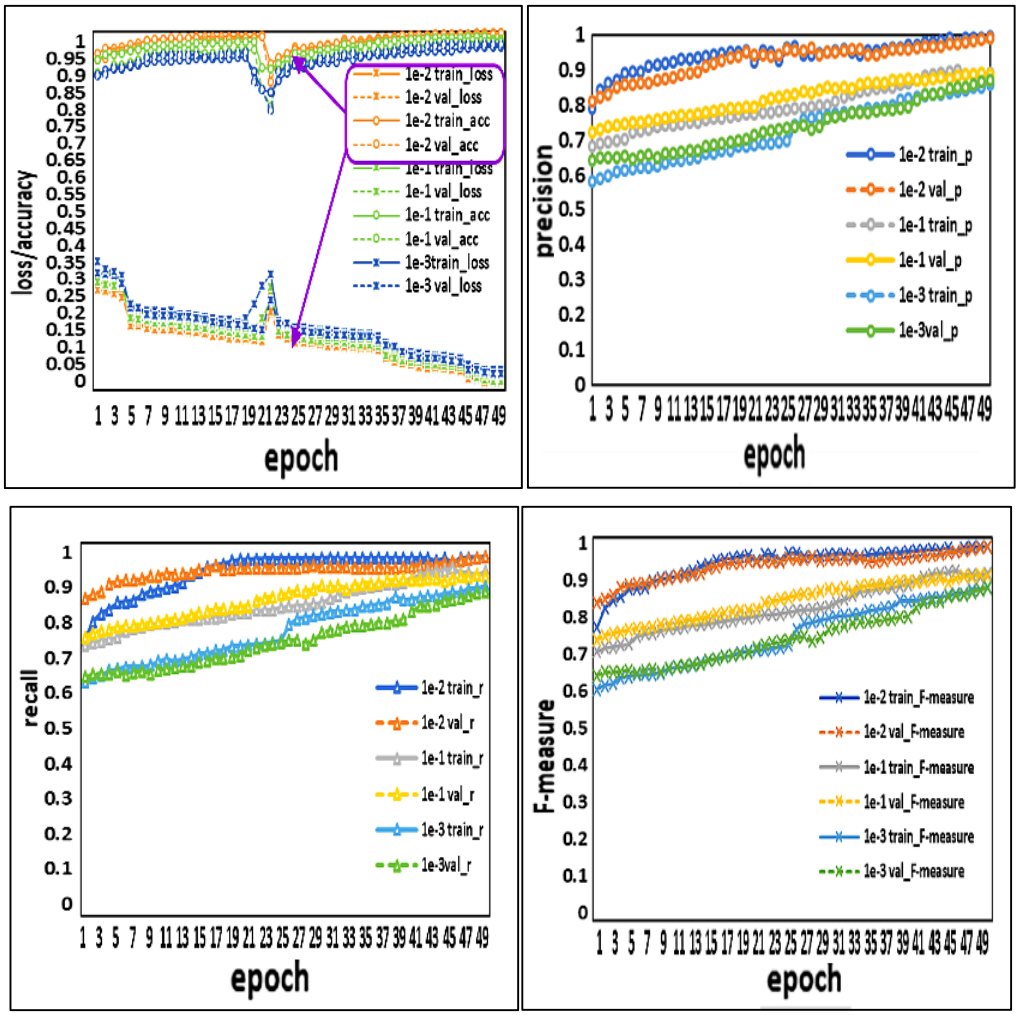
Do {
Calculate the gradient value: $g \leftarrow \frac{1}{m}\nabla_\theta \sum_i L(f(x_i;\theta), y_i)$;
Update the first moment estimate: $s \leftarrow \beta_1 s + (1-\beta_1) g$;
Update the secondary moment estimate: $r \leftarrow \beta_2 r + (1-\beta_2) g \odot g$;
Revise the first moment variance: $\hat{s} \leftarrow \frac{s}{1-\beta_1^t}$
Revise the secondary moment variance: $\hat{r} \leftarrow \frac{r}{1-\beta_2^t}$
Calculate the updating part of the parameter: $\Delta\theta = -\eta \frac{\hat{s}}{\sqrt{\hat{r}+\varepsilon}}$
Update the parameter: $\theta \leftarrow \theta + \Delta\theta$
}
while the network not converge

9.4. Crack Segmentation Experiments and Discussion

9.4.1 CrackU-net Model Training

We used the Adam optimization method in the model training process. A batch size of 2 has been selected. The model learning rate is 0.0001. Figure 9.5 shows the crack segmentation accuracy and the loss changing during the first 50 epochs of the model training process.

Since we used 10-fold cross validation during the model training process, both the validation curve and the training curve can be obtained simultaneously. It can be observed from these curves that the accuracy of both the training set and the validation set are close to 0.99, while the loss is lower than 0.03.



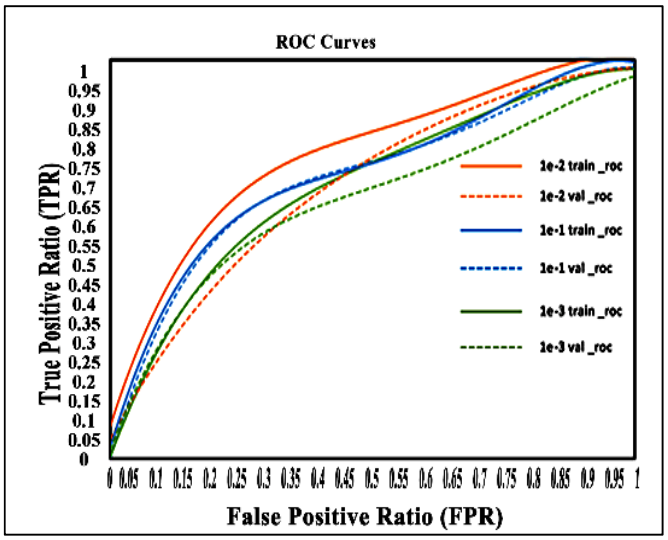
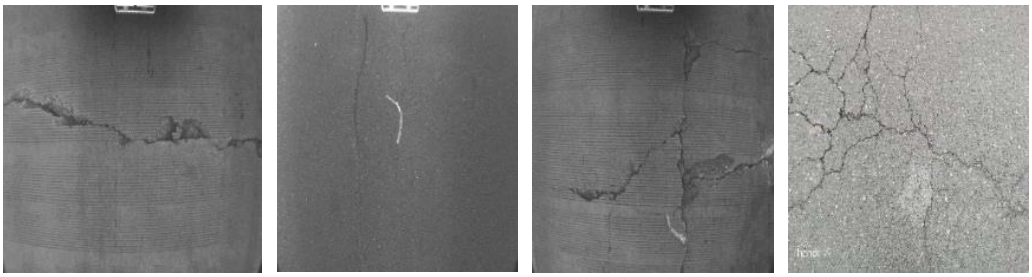


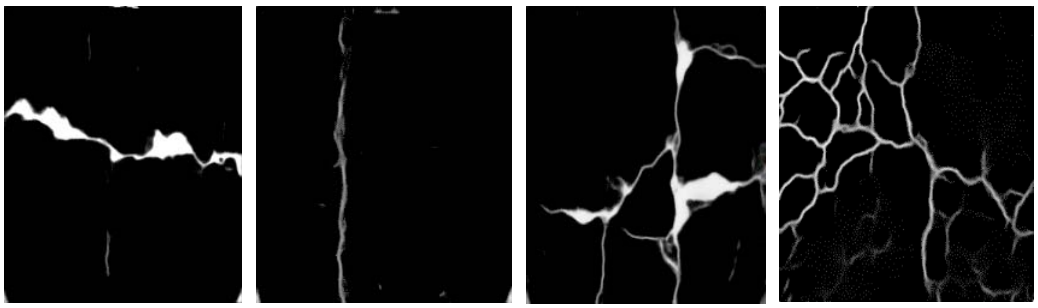
Figure 9.5: Training curve of CrackU-net

9.4.2 Crack Segmentation Results

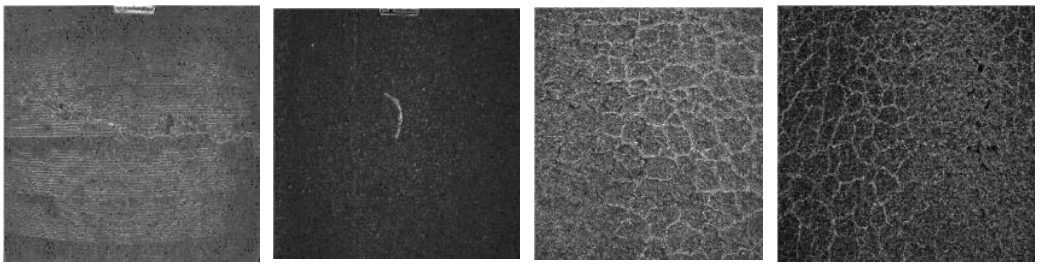
In this part, we compared the performance of CrackU-net with traditional image segmentation methods. The results demonstrate CrackU-net outperforms all the other methods. Some examples of comparative experiments are shown in Figures 9.6 and 9.7. Both linear cracks (transverse cracks and longitudinal cracks) and non-linear cracks (fatigue cracks and other complex map cracks) can be segmented precisely from the road background by CrackU-net. In comparison, traditional image segmentation methods can hardly get satisfying results, especially when the road surface contains high noise.



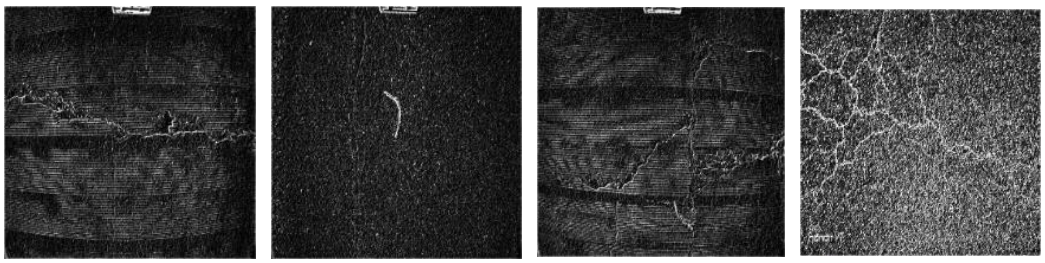
(a) Original crack images



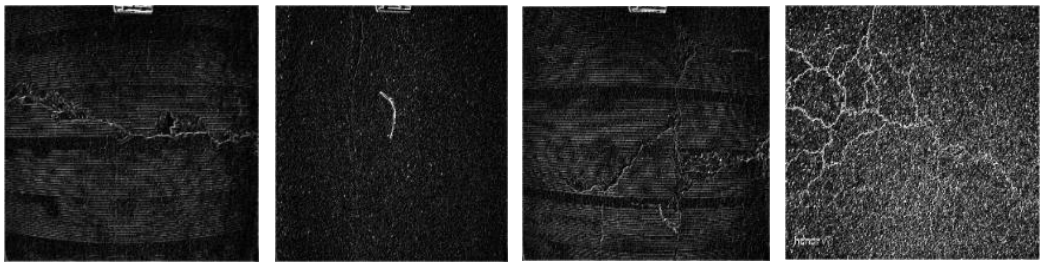
(b) Proposed method



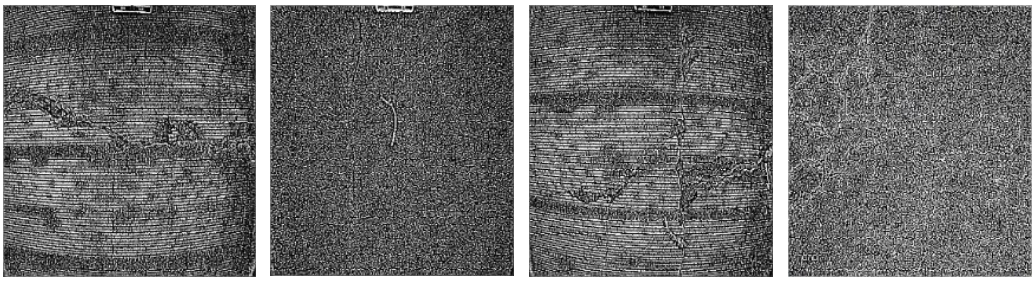
(c) Roberts



(d) Sobel



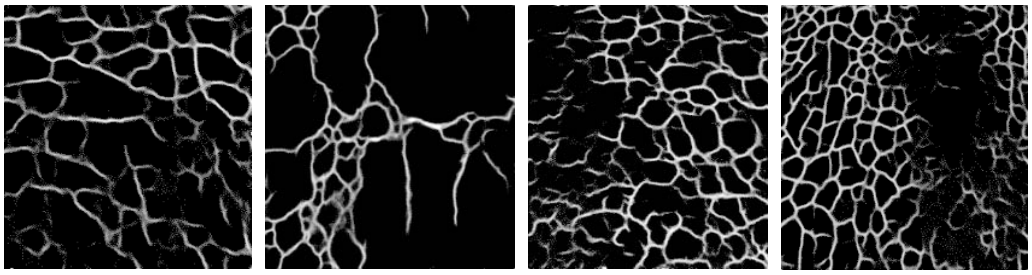
(e) Prewitt



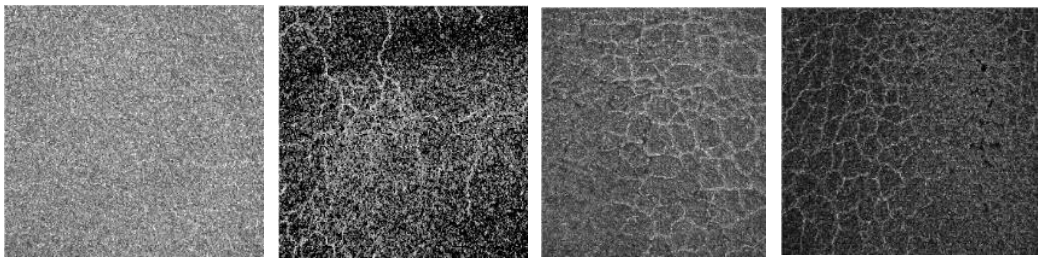
(f) LG

Figure 9.6: Comparative pavement crack segmentation example I

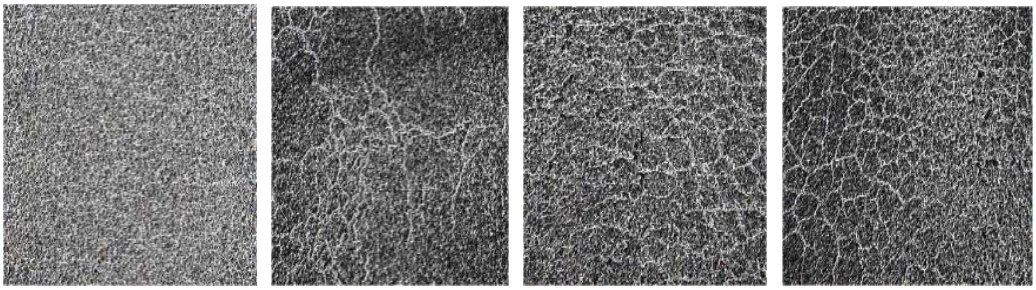
(a) Original crack images



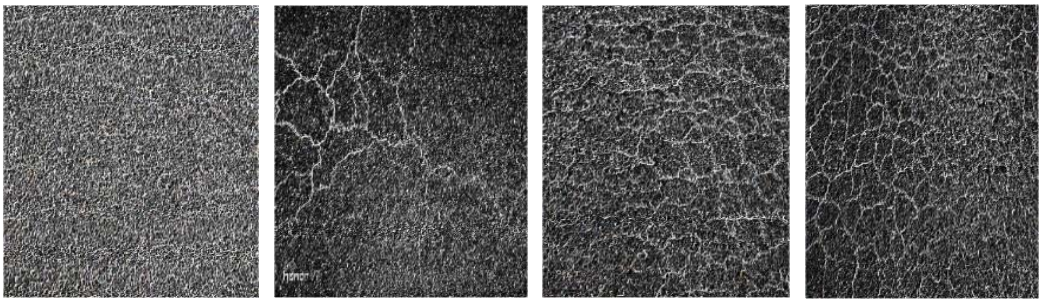
(b) Proposed method



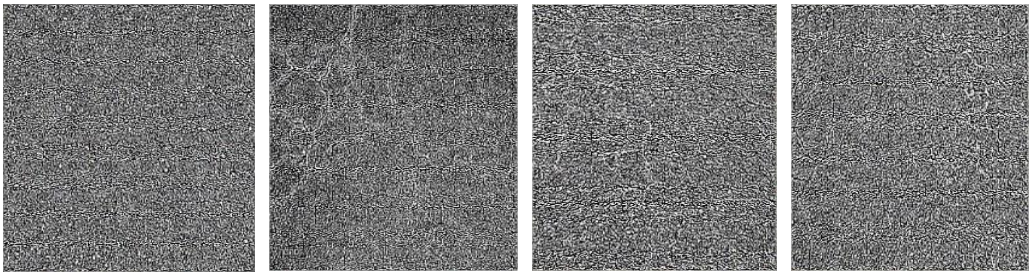
(c) Roberts



(d) Sobel



(e) Premitt



(f) LG

Figure 9.7: Comparative pavement crack segmentation example II

In Table 9.3, comparisons are made of the F1-score and the crack segmentation error between the proposed CrackU-net and the other state-of-the-art methods: FCN and U-net. As can be seen, the F1-score (Equation (9.19)) of CrackU-net is much higher than the other

two network models. In terms of errors, these three methods are close to each other, especially the crack segmentation error between U-net and CrackU-net which only have 0.002 in difference. However, considering both crack segmentation accuracy and the error, CrackU-net outperforms all these methods.

$$F_1 - score = 2 \times \frac{Precision \times Recall}{\Pr ecision + \mathrm{Re} call} \tag{9.19}$$

Table 9.3: Performance comparison with FCN and traditional U-net

Method	F1-score	Error
FCN	0.5691	0.028
U-net	0.7145	0.020
CrackU-net	0.9815	0.018

9.5. Chapter Summary

This chapter discussed the deep learning method for the pixel-level pavement crack image segmentation.

Pavement surface images are collected by the automated pavement investigation vehicles, action camera, and smartphones. Both crack images from asphalt concrete pavement surface and the PCC surface are included in the raw dataset.

The initially collected dataset is processed with data augmentation methods in two aspects. On one hand, the simplest data augmentation method used in chapter 8 is adopted to increase the training dataset. On the other hand, MLS image deformation method is also

used for data augmentation.

The CrackU-net architecture is proposed as an improved version of the traditional U-net model. The developed CrackU-net is composed of four convolutional layers, four Max-pooling layers, and four up-convolutional layers. Cross-entropy loss with L2 normalization is designed as the cost function Adam algorithm is used for the model optimization in the model training procedure. The trained CrackU-net can segment crack images with the accuracy of close to 0.99 and error lower than 0.03.

The performance of CrackU-net is compared with both traditional pixel-level segmentation approaches and other state-of-the-art methods. The results prove that CrackU-net outperforms all the other methods. Therefore, the CrackU-net shows a promising future in improving the overall performance of pavement management systems.

CHAPTER 10. Conclusions and Recommendations

This thesis involved a comprehensive literature review on pavement management systems, pavement surface distresses manifestations, pavement condition assessment methods and the application of machine learning technologies in pavement engineering. On this basis, the most significant research gaps in current automated pavement condition evaluation, distress detection, and management systems are outlined. The main reason that caused those gaps are the imbalance between the highly advanced computerized technologies and the less intelligent pavement condition analytical methods used in current pavement engineering practice. Hence, the main objective of this research is to develop innovative pavement condition evaluation and performance prediction analytical tools, which can fill the gaps in current literature by using the state-of-the-art machine learning techniques.

To address the objectives of this research, the whole thesis developed six analytical tools based on two kinds of input data formats, which are structured data and the non-structured data (image data). The reason is that structured data can be processed using shallow machine learning techniques, which are not computationally expensive. However, the non-structured data, referring to pavement images, contain more information than the structured data. Hence, deep learning approaches that require for specific system hardware configurations are adopted for model development.

To ensure the reliability of the tools developed in this research, data acquired from multiple sources are used. To be specific, the structured pavement performance data include: 1) the historical IRI data extracted from the InfoPave website; 2) pavement surface distresses and the performance data acquired by ARAN 9000; and 3) pavement surface modulus data acquired by LWD. The non-structured data (pavement surface images) data are acquired by 1) high speed pavement condition inspection vehicles; 2) action cameras; and 3) smartphones. Besides raw data collection, data augmentation operations are also conducted

in the non-structured data-based pavement crack detection. The reason is that the performance of the deep convolutional models is highly related to the training dataset. Through data augmentation, more comprehensive model training dataset can be obtained, thus the trained model can be more robust.

Based on the well-prepared dataset, several analytical tools are developed and validated. IRI prediction tool is developed based on the historical data using the multi-granularity fuzzy time series method. The image indicators-based pavement roughness and the overall performance prediction tools are developed using the XGBoost approach. The dominant surface distress indicators are analyzed using BPNN and MIVs analysis. The pavement structural performance evaluation tools are developed by the heuristic method optimized SVM models. Both box-level (CrackDN) and pixel-level (CrackU-net) pavement crack detection tools are developed using the deep convolutional neural networks. All these analytical tools are trained with satisfactory performance.

The findings of this research provide the following suggestions:

1) Pavement structure performance analytical tools should be integrated into current PMS.

2) Image indicators are promising for being used to predict the pavement roughness conditions and the overall performance condition.

3) Pavement surface cracking, especially the wheel path cracking, should be treated as priorities to other surface defects in order to prolong the pavement service life.

4) Both box-level and pixel-level deep learning-based crack analytical tools should be integrated into the pavement condition analyzers.

The reason can be explained by the problems shown in Figure 10.1 and Figure 10.2. As Figure 10.1 shows, there are two significant problems being observed in the box-level crack detection results, which are incomplete crack locations and multiple boxes being found for

the single cracks. Therefore, there will be a high probability that the system tends to be less reliable if only include the box-level crack detection. Specially, the records for the number of cracks for the tested pavement segments are less reliable. However, if only include the pixel-level crack detections, as Figure 10.2 shows, the false positive and the true negative errors of the detection results tend to either lose crack information or mis-detect some backgrounds as the crack regions. As a result, the number of cracks tends to be wrongly recorded, which leads to the incorrect pavement damage condition assessments. Therefore, both two aspects of crack analyses should be included into the pavement management systems. Meanwhile, the box-level crack detections should be conducted first for crack location and counting. After that, pixel-level crack detections should be conducted for each crack location, and the detection results should be for crack length, width, area calculations, damage density and percent evaluation. Only by combing the results of these two aspects of analyses, the pavement cracking conditions can be given comprehensive evaluation.

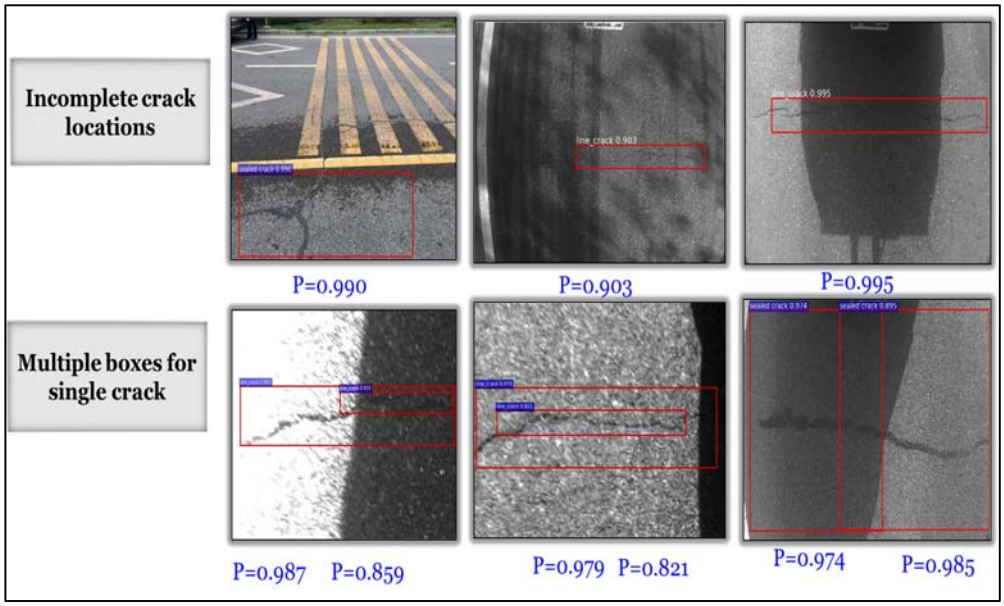


Figure 10. 1: Problem of CrackDN

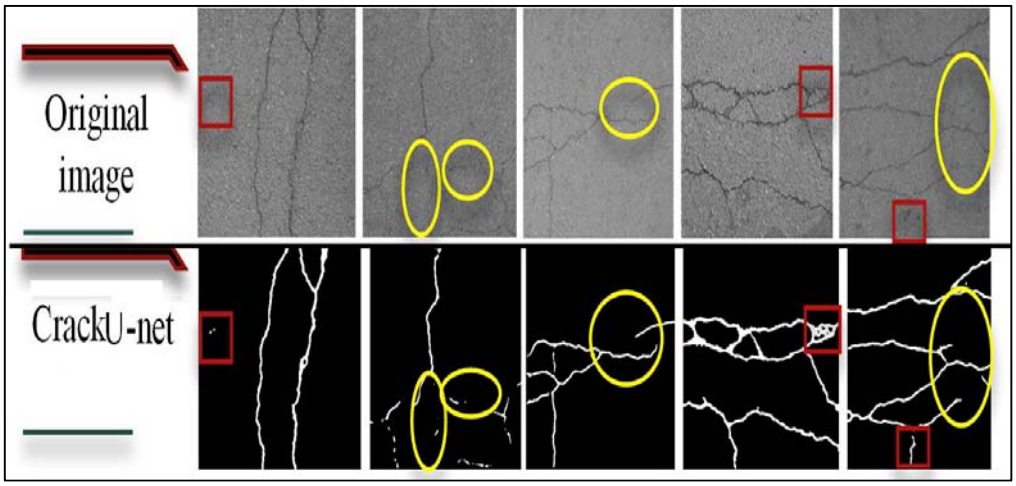


Figure 10.2: Problems in CrackU-net (rectangle: false positive error; circle: true negative error)

5) Two aspects of researches are recommended to be given high attention in the future work, which are: 1) Incomplete crack locations and the multiple crack detections for the single crack, which are observed in the box-level crack detection results; 2) false positive and the true negative errors observed in the pixel-level crack detection results.

References

Abdel-Qader, I., Abudayyeh, O. & Kelly, M.E. (2003). Analysis of edge-detection techniques for crack identification in bridges. Journal of Computing in Civil Engineering, 17 (4), 255-263.

Adhikari, R., Moselhi, O. & Bagchi, A. (2014). Image-based retrieval of concrete crack properties for bridge inspection. Automation in construction, 39, 180-194.

Adu-Gyamfi, Y., Okine, N. A., Garateguy, G., Carrillo, R., & Arce, G. R. (2011). Multiresolution information mining for pavement crack image analysis. Journal of Computing in Civil Engineering, 26(6), 741-749.

Adu-Gyamfi, Y., Kambhamettu, C., & Okine, N. A. (2013). Performance assessment of flexible pavements using active contour models. Airfield and Highway Pavement 2013: Sustainable and Efficient Pavements, 887-902.

Agathos, K., Chatzi, E., & Bordas, S. P. A. (2018). Multiple crack detection in 3D using a stable XFEM and global optimization. Computational Mechanics, 62(4), 835-852.

Alpaydin, E. (2009). Introduction to machine learning, MIT press.

American Association of State Highway and Transportation Officials (AASHTO). 2008. Mechanistic-Empirical Pavement Design Guide—A Manual of Practice. Interim Edition. American Association of State Highway and Transportation Officials, Washington, DC.

Amhaz, R., Chambon, S., Idier, J., & Baltazart, V. (2014). A new minimal path selection algorithm for automatic crack detection on pavement images. Proc., IEEE International Conference on Image Processing (ICIP), 788-792.

Andrews, R., Diederich, J., & Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-based systems, 8(6), 373-389.

Ashrafian, A., Taheri Amiri, M. J., Rezaie-Balf, M., Ozbakkaloglu, T., & Lotfi-Omran, O. (2018). Prediction of compressive strength and ultrasonic pulse velocity of fiber reinforced concrete incorporating nano silica using heuristic regression methods. Construction and

Building Materials, 190, 479-494.

Attoh-Okine, N., Barner, K., Bentil, D., & Zhang, R. (2008). The empirical mode decomposition and the Hilbert-Huang transform. Springer.

Ayenu-Prah, A. & Attoh-Okine, N. (2008). Evaluating pavement cracks with bidimensional empirical mode decomposition. EURASIP Journal on Advances in Signal Processing, 2008 (1), 861701.

Bai, S. (2017). Growing random forest on deep convolutional neural networks for scene categorization. Expert Systems with Applications, 71, 279-287.

Barakat, N., & Bradley, A. P. (2010). Rule extraction from support vector machines: a review. Neurocomputing, 74(1-3), 178-190.

Calvini, M., Carpita, M., Formentini, A., & Marchesoni, M. (2015). PSO-based self-commissioning of electrical motor drives. IEEE Transactions on Industrial Electronics, 62(2), 768-776.

Cha, Y.J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Computer-Aided Civil and Infrastructure Engineering, 33 (9), 731-747.

Chambon, S., & Moliard, J.-M. (2011). Automatic road pavement assessment with image processing: review and comparison. International Journal of Geophysics.

Chambon, S., Gourraud, C., Moliard, J. M., & Nicolle, P. (May 2010). Road crack extraction with adapted filtering and Markov model-based segmentation: introduction and validation. Proc., International Joint Conference on Computer Vision Theory and Applications, VISAPP.

Chapeleau, X., Blanc, J., Hornych, P., Gautier, J.-L., & Carroget, J. (2014). Use of distributed fiber optic sensors to detect damage in a pavement. 1847-1854.

Chatrchyan, S., Khachatryan, V., Sirunyan, A. M., Tumasyan, A., Adam, W., Aguilo, E., ... & Friedl, M. (2012). Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. Physics Letters B, 716(1), 30-61.

Chen, F.C., Jahanshahi, M.R., Wu, R.T. & Joffe, C. (2017a). A texture-based video processing

methodology using bayesian data fusion for autonomous crack detection on metallic surfaces. Computer-Aided Civil and Infrastructure Engineering, 32 (4), 271-287.

Chen, J.-H., Su, M.-C., Cao, R., Hsu, S.-C. & Lu, J.-C. (2017b). A self organizing map optimization based image recognition and processing model for bridge crack inspection. Automation in Construction, 73, 58-66.

Cheng, H.-D., & Miyojim, M. (1998). Automatic pavement distress detection system. Information Sciences, 108(1-4), 219-240.

Cheng, H., Wang, J., Hu, Y., Glazier, C., Shi, X., & Chen, X. (2001). Novel approach to pavement cracking detection based on neural network. Transportation Research Record: Journal of the Transportation Research Board (1764), 119-127.

Cheng, M. Y., Firdausi, P. M., and Prayogo, D. (2014). High-performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT). Engineering Applications of Artificial Intelligence, 29, 104-113.

Cheng, J. C. P., & Wang, M. (2018). Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques. Automation in Construction, 95, 155-171.

Chong, G. J., W. A. Phang, & G. A. Wong.(1989) Manual for Condition Rating of Flexible Pavements—Distress Manifestations. SP-024. Ontario Ministry of Transportation, Toronto, Canada, 1989.

Chua, K. M., & Xu, L. (1994). Simple procedure for identifying pavement distresses from video images. Journal of transportation engineering, 120(3), 412-431.

Cord, A., & Chambon, S. (2012). Automatic road defect detection by textural pattern recognition based on AdaBoost. Computer-Aided Civil and Infrastructure Engineering, 27(4), 244-259.

Cortes, C., & Vapnik, V. (1995). Machine learning. Support vector networks, 20, 273-297.

Deng, J. (2009). A large-scale hierarchical image database. Proc. of IEEE Computer Vision and Pattern Recognition, 2009.

Diederich, J. (Ed.). (2007). Rule extraction from support vector machines (Vol. 80). Springer.

Dorafshan, S., Thomas, R. J., Coopmans, C., & Maguire, M. (2018). Deep Learning Neural Networks for sUAS-Assisted Structural Inspections: Feasibility and Application. 874-882.

Erkal, B.G. & Hajjar, J.F. (2017). Laser-based surface damage detection and quantification using predicted surface properties. Automation in Construction, 83, 285-302.

Fan, H. W., Zhang, G. Y., Ding, A. L., Xie, C. R., & Xu, T. (2010). Improved BP algorithm and its application in detection of pavement crack. Chang'an Daxue Xuebao (Ziran Kexue Ban)/Journal of Chang'an University (Natural Science Edition), 30(1), 46-53.

Farran, M., & Zayed, T. (2012). New life-cycle costing approach for infrastructure rehabilitation. Engineering, Construction and Architectural Management, 19(1), 40-60.

Feng, C., Liu, M.-Y., Kao, C.-C., & Lee, T.-Y. (2017). Deep Active Learning for Civil Infrastructure Defect Detection and Classification. Computing in Civil Engineering 2017, 298-306.

Feng, P., Xiao-Ting, L., Qian, Z., Wei-Xing, L., & Qi, G. (2013). Analysis of standard particle swarm optimization algorithm based on Markov chain. Acta Automatica Sinica, 39(4), 381-389.

Flintsch, G. W., & McGhee, K. K. (2009). Quality management of pavement condition data collection, Transportation Research Board, (Vol. 401).

Forsyth, D. A., & Ponce, J. (2003). A modern approach. Computer vision: a modern approach, 88-101.

Gao, B., Li, X., Woo, W. L., & yun Tian, G. (2018). Physics-Based Image Segmentation Using First Order Statistical Properties and Genetic Algorithm for Inductive Thermography Imaging. IEEE Transactions on Image Processing, 27(5), 2160-2175.

Garcia, C., & Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. IEEE Transactions on pattern analysis and machine intelligence, 26(11), 1408-1423.

Gavilán, M., Balcones, D., Marcos, O., Llorca, D. F., Sotelo, M. A., Parra, I., Ocaña, M., Aliseda, P., Yarza, P., & Amírola, A. (2011). Adaptive road crack detection system by pavement

classification. Sensors, 11(10), 9628-9657.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

Goh, H., Thome, N., Cord, M., & Lim, J.-H. (2014). Learning deep hierarchical visual feature coding. IEEE transactions on neural networks and learning systems, 25(12), 2212-2225.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. Machine learning, 3(2), 95-99.

Gopalakrishnan, K., & Peeta, S. (Eds.). (2010). Sustainable and resilient critical infrastructure systems: simulation, modeling, and intelligent engineering. Springer Science & Business Media.

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

Grandsaert, P. J. (2015). Integrating pavement crack detection and analysis using autonomous unmanned aerial vehicle imagery. AIR FORCE INSTITUTE OF TECHNOLOGY WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT.

Greco, S., Matarazzo, B., & Słowiński, R. (2008). Dominance-based rough set approach to interactive multiobjective optimization. In Multiobjective optimization (pp. 121-155). Springer, Berlin, Heidelberg.

Haas, C., & Hendrickson, C. (1990). Computer-based model of pavement surfaces. Transportation Research Record, 1260, 91-98.

Haas, R., Hudson, W. R., & Kennedy, T. W. (1975). ROLE OF PAVEMENT EVALUATION IN A PAVEMENT MANAGEMENT SYSTEM. 732-742.

Hadjidemetriou, G. M., Vela, P. A., & Christodoulou, S. E. (2018). Automated Pavement Patch Detection and Quantification Using Support Vector Machines. Journal of Computing in Civil Engineering, 32(1).

Hajj, E., Loria, L., Sebaaly, P., Borroel, C., & Leiva, P. (2011). Optimum time for application of slurry seal to asphalt concrete pavements. Transportation Research Record: Journal of the Transportation Research Board, (2235), 66-81.

Hasni, H., Alavi, A. H., Jiao, P., & Lajnef, N. (2017). Detection of fatigue cracking in steel bridge girders: A support vector machine approach. Archives of Civil and Mechanical Engineering, 17(3), 609-622.

Haykin, S. (1999). Self-organizing maps: Neural networks-A comprehensive foundation 2nd Edition. Prentice-Hall.

He, K., Zhang, X., Ren, S., & Sun, J. (2014, September). Spatial pyramid pooling in deep convolutional networks for visual recognition. In European conference on computer vision (pp. 346-361). Springer, Cham.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6), 82-97.

Hoang, N. D. (2018). An Artificial Intelligence Method for Asphalt Pavement Pothole Detection Using Least Squares Support Vector Machine and Neural Network with Steerable Filter-Based Feature Extraction. Advances in Civil Engineering, 2018

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural networks, 2(5), 359-366.

Hsie, M., Ho, Y. F., Lin, C. T., & Yeh, I. C. (2012). Modeling asphalt pavement overlay transverse cracks using the genetic operation tree and Levenberg-Marquardt Method. Expert Systems with Applications, 39(5), 4874-4881.

Hu, Y., Chang, H., Nian, F., Wang, Y., & Li, T. (2016). Dense crowd counting from still images with convolutional neural networks. Journal of Visual Communication and Image

Representation, 38, 530-539.

Huang, J., Liu, W., and Sun, X. (2014). "A pavement crack detection method combining 2D with 3D information based on Dempster-Shafer theory." Computer-Aided Civil and Infrastructure Engineering, 29(4), 299-313.

Huang, S. M., Hsu, C. J., Lee, C., & Chang, C. C. (2005). Application of neural network for selection of airport rigid pavement maintenance strategies. Journal of Marine Science and Technology, 13(2), 125-132.

Huang, Y. & Xu, B., 2006. Automatic inspection of pavement cracking distress. Journal of Electronic Imaging, 15 (1), 013017.

Huang, Y. H. (1993). *Pavement analysis and design*.

Huang, Y., & Tsai, Y. (2011). Dynamic programming and connected component analysis for an enhanced pavement distress segmentation algorithm. Transportation Research Record: Journal of the Transportation Research Board(2225), 89-98.

Ito, A., Aoki, Y., & Hashimoto, S. (2002, November). Accurate extraction and measurement of fine cracks from concrete block surface image. In IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the] (Vol. 3, pp. 2202-2207). IEEE.

Ivanović, M., & Radovanović, M. (2014). Modern machine learning techniques and their applications. In International Conference on Electronic, Communication, and Network.

Jahanshahi, M.R., Jazizadeh, F., Masri, S.F. & Becerik-Gerber, B. (2012). Unsupervised approach for autonomous pavement-defect detection and quantification using an inexpensive depth sensor. Journal of Computing in Civil Engineering, 27 (6), 743-754.

Jahanshahi, M.R., & Masri, S.F. (2012). Adaptive vision-based crack detection using 3d scene reconstruction for condition assessment of structures. Automation in Construction, 22, 567-576.

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. Computer, 29(3), 31-44.

Jang, J., Yang, Y., Smyth, A. W., Cavalcanti, D., & Kumar, R. (2017). Framework of Data

Acquisition and Integration for the Detection of Pavement Distress via Multiple Vehicles. Journal of Computing in Civil Engineering, 31(2).

Jiang, C. & Tsai, Y.J. (2015). Enhanced crack segmentation algorithm using 3d pavement data. Journal of Computing in Civil Engineering, 30 (3), 04015050.

Kaseko, M. S., Lo, Z.-P., & Ritchie, S. G. (1994). Comparison of traditional and neural classifiers for pavement-crack detection. Journal of transportation engineering, 120(4), 552-569.

Kaul, V., Tsai, Y., & Yezzi, A. (2010). Detection of curves with unknown endpoints using minimal path techniques. Georgia Institute of Technology.

Khanna, V., Das, B., Bisht, D., & Singh, P. (2015). A three diode model for industrial solar cells and estimation of solar cell parameters using PSO algorithm. Renewable Energy, 78, 105-113.

Kim, Y.-S., Haas, C. T., & Greer, R. (1998). Path planning for machine vision assisted, teleoperated pavement crack sealer. Journal of Transportation Engineering, 124(2), 137-143.

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

Kirschke, K. & Velinsky, S. (1992). Histogram-based approach for automated pavement-crack sensing. Journal of Transportation Engineering, 118 (5), 700-710.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Advanced Engineering Informatics, 29(2), 196-210.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Lempert, A. A., Sidorov, D. N., Zhukov, A. V., & Nguyen, G. L. (2016). A combined work optimization technology under resource constraints with an application to road repair.

Automation and Remote Control, 77(11), 1883-1893.

Li, G., He, S., Ju, Y. & Du, K. (2014). Long-distance precision inspection method for bridge cracks with image processing. Automation in Construction, 41, 83-95.

Li, G., Zhao, X., Du, K., Ru, F. & Zhang, Y. (2017). Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine. Automation in Construction, 78, 51-61.

Li, G. (2013). New weighted mean filtering algorithm for surface image based on grey entropy. Sensors & Transducers, 161(12), 21.

Li, N., Hou, X., Yang, X., & Dong, Y. (2009, November). Automation recognition of pavement surface distress based on support vector machine. In Intelligent Networks and Intelligent Systems, 2009. ICINIS'09. Second International Conference on (pp. 346-349). IEEE.

Li, W., Huyan, J., Tighe, S. L., Ren, Q.-q., & Sun, Z.-y. (2017). Three-dimensional pavement crack detection algorithm based on 2D empirical mode decomposition. Journal of Transportation Engineering, Part B: Pavements, 143(2), 04017005.

Liao, T. Y. (2017). On-line vehicle routing problems for carbon emissions reduction. Computer-Aided Civil and Infrastructure Engineering, 32(12), 1047-1063.

Lin, J., & Liu, Y. (2010, August). Potholes detection based on SVM in the pavement distress image. In Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on (pp. 544-547). IEEE.

Ling, L., Peikang, H., Xiaohu, W., & Xudong, P. (2009). Image edge detection based on beamlet transform. Journal of Systems Engineering and Electronics, 20(1), 1-5.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234, 11-26.

Liu, Y., Zhao, T., Ju, W., Shi, S., Shi, S., & Shi, S. (2017). Materials discovery and design using machine learning. Journal of Materiomics, 3(3), 159-177.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern*

recognition (pp. 3431-3440).

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2017, November). Extraction of Road Maintenance Criteria using Machine Learning and Spatial Information. In Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics (p. 9). ACM.

Maeda, K., Takahashi, S., Ogawa, T., & Haseyama, M. (2017). Distress classification of road structures via adaptive Bayesian network model selection. Journal of Computing in Civil Engineering, 31(5).

Mahler, D. S., Kharoufa, Z. B., Wong, E. K., & Shaw, L. G. (1991). Pavement distress analysis using image processing techniques. Computer-Aided Civil and Infrastructure Engineering, 6(1), 1-14

Maini, R., & Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. International journal of image processing (IJIP), 3(1), 1-11.

Majidzadeh, K., Vedaie, B., & Kennedy Jr, J. C. (1990). Pavement Management System to Maximize Pavement Investment and Minimize Cost. Transportation Research Record (1272).

Miller, J. S., & Bellinger, W. Y. (2014). Distress identification manual for the long-term pavement performance program (No. FHWA-HRT-13-092). United States. Federal Highway Administration. Office of Infrastructure Research and Development.

Molina-Cabello, M. A., Luque-Baena, R. M., López-Rubio, E., & Thurnhofer-Hemsi, K. (2018). Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images. Integrated Computer-Aided Engineering, 25(4), 321-333.

Moussa, G., & Hussain, K. (2011). A new technique for automatic detection and parameters estimation of pavement crack. In 4th International Multi-Conference on Engineering Technology Innovation, IMETI (Vol. 2011).

Nabian, M. A., & Meidani, H. (2018). Deep Learning for Accelerated Seismic Reliability Analysis of Transportation Networks. Computer-Aided Civil and Infrastructure

Engineering, 33(6), 443-458.

Nasse, F., Thurau, C., & Fink, G. A. (2009, September). Face detection using gpu-based convolutional neural networks. In International Conference on Computer Analysis of Images and Patterns (pp. 83-90). Springer, Berlin, Heidelberg.

Nejad, F. M., & Zakeri, H. (2011). A comparison of multi-resolution methods for detection and isolation of pavement distress. Expert Systems with Applications, 38(3), 2857-2872.

Nejad, F.M. & Zakeri, H. (2011). An optimum feature extraction method based on wavelet–radon transform and dynamic neural network for pavement distress classification. Expert Systems with Applications, 38 (8), 9442-9460.

Ng, A. W. M., Tran, D. H., & Osman, N. Y. (2006). Modelling serviceability deterioration of concrete stormwater pipes using genetic algorithm trained neural networks. WSEAS Transactions on Systems, 5(7), 1662-1670.

Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., & Barbano, P. E. (2005). Toward automatic phenotyping of developing embryos from videos. IEEE Transactions on Image Processing, 14(9), 1360-1371.

Nisanth, A. & Mathew, A. (2014). Automated visual inspection on pavement crack detection and characterization. International Journal of Technology and Engineering System, 6 (1), 14-20.

Nishikawa, T., Yoshida, J., Sugiyama, T. & Fujino, Y. (2012). Concrete crack detection by multiple sequential image filtering. Computer-Aided Civil and Infrastructure Engineering, 27 (1), 29-47.

Nowlan, S. J., & Platt, J. C. (1995). A convolutional neural network hand tracker. Advances in neural information processing systems, 901-908.

O'Byrne, M., Schoefs, F., Ghosh, B., & Pakrashi, V. (2013). Texture analysis based damage detection of ageing infrastructural elements. Computer-Aided Civil and Infrastructure Engineering, 28 (3), 162-177.

Oliveira, H., & Correia, P. L. (2009, August). Automatic road crack segmentation using entropy

and image dynamic thresholding. In Signal Processing Conference, 2009 17th European (pp. 622-626). IEEE.

Oliveira, H., Caeiro, J. J., & Correia, P. L. (2010, September). Improved road crack detection based on one-class Parzen density estimation and entropy reduction. In Image Processing (ICIP), 2010 17th IEEE International Conference on (pp. 2201-2204). IEEE.

Oliveira, H., & Correia, P. L. (2013). Automatic road crack detection and characterization. IEEE Transactions on Intelligent Transportation Systems, 14(1), 155-168.

Ouma, Y. O., & Hahn, M. (2017). Pothole detection on asphalt pavements from 2D-colour pothole images using fuzzy c-means clustering and morphological reconstruction. Automation in Construction, 83, 196-211Ouyang, W. & Xu, B. (2013). Pavement cracking measurements using 3d laser-scan images. Measurement Science and Technology, 24 (10), 105204.

Piersanti, S., & Orlandi, A. (2018). Genetic Algorithm Optimization for the Total Radiated Power of a Meandered Line by Using an Artificial Neural Network. IEEE Transactions on Electromagnetic Compatibility, 60(4), 1014-1017.

Poli, R. (2007). An analysis of publications on particle swarm optimization applications. Essex, UK: Department of Computer Science, University of Essex.

Qiu, S., Wang, K. C., Zhang, A., Li, Q. J., & Moravec, M. M. (2018). A comprehensive system for AASHTO PP69-10-based pavement rut evaluation using 1-mm 3D pavement surface model. International Journal of Pavement Engineering, 19(6), 489-501.

Radopoulou, S. C., & Brilakis, I. (2017). Automated Detection of Multiple Pavement Defects. Journal of Computing in Civil Engineering, 31(2).

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2016). Neural network, machine learning, and evolutionary approaches for concrete material characterization. ACI Materials Journal, 113(6), 781-789.

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. Engineering Structures, 156, 598-607.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

Robinson, C., Beg, M., Dossey, T., & Hudson, W. (1996). Distress prediction models for rigid pavements for Texas pavement management information system. Transportation Research Record: Journal of the Transportation Research Board(1524), 145-151.

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

Salari, E., & Bao, G. (2011, February). Pavement distress detection and severity analysis. In Image Processing: Machine Vision Applications IV (Vol. 7877, p. 78770C). International Society for Optics and Photonics.

Salari, E., & Yu, X. (2011, October). Pavement distress detection and classification using a Genetic Algorithm. In Applied imagery pattern recognition workshop (AIPR), 2011 IEEE (pp. 1-5). IEEE.

Salari, E., & Ouyang, D. (2012, May). An image-based pavement distress detection and classification. In Electro/Information Technology (EIT), 2012 IEEE International Conference on (pp. 1-6). IEEE.

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

Samarasinghe, S. (2016). Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition. Auerbach publications.

Santhi, B., Krishnamurthy, G., Siddharth, S. & Ramakrishnan, P. (2012). Automatic detection of cracks in pavements using edge detection operator. Journal of Theoretical and Applied Information Technology, 36 (2), 199-205.

Sayers, W. (1987). BARGAINING FOR THE LIFE OF BRES IN'CATH MAIGE TUIRED'.

BULLETIN OF THE BOARD OF CELTIC STUDIES-BWLETIN Y BWRDD GWYBODAU CELTAIDD, 34, 26-40.

Schlotjes, M. R., Henning, T. F., Burrow, M. P., & Evdorides, H. T. (2015). Using support vector machines to predict the probability of pavement failure. Proceedings of the Institution of Civil Engineers: Transport, 168(3), 212-222.

Schaefer, S., McPhail, T., & Warren, J. (2006, July). Image deformation using moving least squares. In ACM transactions on graphics (TOG) (Vol. 25, No. 3, pp. 533-540). ACM.

Sermanet, P., Kavukcuoglu, K., Chintala, S., & LeCun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3626-3633).

Sets, F., & Zadeh, L. A. (1965). Inform. Control, 8, 338-353.

Sharifi, M., Fathy, M., & Mahmoudi, M. T. (2002, April). A classified and comparative study of edge detection algorithms. In Information Technology: Coding and Computing, 2002. Proceedings. International Conference on (pp. 117-120). IEEE.

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. IEEE Transactions on Intelligent Transportation Systems, 17(12), 3434-3445.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sinha, S.K. & Fieguth, P.W. (2006). Automated detection of cracks in buried concrete pipe images. Automation in Construction, 15 (1), 58-72.

Song, Q., & Chissom, B. S. (1993). Forecasting enrollments with fuzzy time series—part I. Fuzzy sets and systems, 54(1), 1-9.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.Subirats, P., Dumoulin, J., Legeay, V., & Barba, D. (2006, October). Automation of pavement surface crack detection using the continuous wavelet

transform. In Image Processing, 2006 IEEE International Conference on (pp. 3037-3040). IEEE.

Staniek, M., & Czech, P. (2018). Self-correcting neural network in road pavement diagnostics. Automation in Construction, 96, 75-87.Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3476-3483).

Sugimura, K., Obayashi, S., & Jeong, S. (2007, January). Multi-objective design exploration of a centrifugal impeller accompanied with a vaned diffuser. In ASME/JSME 2007 5th Joint Fluids Engineering Conference (pp. 939-946). American Society of Mechanical Engineers.

Sugimura, K., Obayashi, S., & Jeong, S. (2010). Multi-objective optimization and design rule mining for an aerodynamically efficient and stable centrifugal impeller with a vaned diffuser. Engineering Optimization, 42(3), 271-293.

Sun, Z. Y., Zhao, H. W., Li, W., Hao, X. L., & Huyan, J. (2015). 3D pavement crack identification method based on dual-phase scanning detection. China J Highway Transp, 28(2), 26-32.

Tabatabaee, N., Ziyadi, M., & Shafahi, Y. (2013). Two-stage support vector classifier and recurrent neural network predictor for pavement performance modeling. Journal of Infrastructure Systems, 19(3), 266-274.

Tang, J., & Gu, Y. (2013, October). Automatic crack detection and segmentation using a hybrid algorithm for road distress analysis. In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on (pp. 3026-3030). IEEE.

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1701-1708).

Tapkin, S., Çevik, A., & Uşar, U. (2009). Accumulated strain prediction of polypropylene modified marshall specimens in repeated creep test using artificial neural networks. Expert Systems with Applications, 36(8), 11186-11197.

Tedeschi, A., & Benedetto, F. (2017). A real-time automatic pavement crack and pothole

recognition system for mobile Android-based devices. Advanced Engineering Informatics, 32, 11-25.

Terzi, S., Karaşahin, M., Gökova, S., Tahta, M., Morova, N., & Uzun, I. (2013). Asphalt concrete stability estimation from non-destructive test methods with artificial neural networks. Neural Computing and Applications, 23(3-4), 989-997.

TIGHE, S. E. (2013). Pavement asset design and management guide.

Todkar, S. S., Le Bastard, C., Ihamouten, A., Baltazart, V., Dérobert, X., Fauchard, C., ... & Bosc, F. (2017, June). Detection of debondings with Ground Penetrating Radar using a machine learning method. In Advanced Ground Penetrating Radar (IWAGPR), 2017 9th International Workshop on (pp. 1-6). IEEE.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., & Bregler, C. (2015). Efficient object localization using convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 648-656).

Tong, Z., Gao, J., & Zhang, H. (2018). Innovative method for recognizing subgrade defects based on a convolutional neural network. Construction and Building Materials, 169, 69-82.

Transportation Association of Canada (TAC). (2013). Pavement Asset Design and Management Guide. Ottawa: Transportation Association of Canada (TAC).

Tsai, Y.-C., Kaul, V., & Mersereau, R. M. (2009). Critical assessment of pavement distress segmentation methods. Journal of transportation engineering, 136(1), 11-19.

Tsai, Y. C. J., & Li, F. (2012). Critical assessment of detecting asphalt pavement cracks under different lighting and low intensity contrast conditions using emerging 3D laser technology. Journal of Transportation Engineering, 138(5), 649-656.

Tsai, Y. C., Jiang, C., & Huang, Y. (2012). Multiscale crack fundamental element model for real-world pavement crack classification. Journal of Computing in Civil Engineering, 28(4), 04014012.

Tsai, Y. J., Jiang, C., & Wang, Z. (2014, October). Implementation of automatic crack evaluation using crack fundamental element. In Image Processing (ICIP), 2014 IEEE International

Conference on (pp. 773-777). IEEE.

Tsai, Y. J., & Wang, Z. (2015). Development of an asphalt pavement raveling detection algorithm using emerging 3D laser technology and macrotexture analysis (No. NCHRP IDEA Project 163).

Tsai, Y., Wu, Y., & Lewis, Z. (2013). Full-lane coverage micromilling pavement-surface quality control using emerging 3D line laser imaging technology. Journal of Transportation Engineering, 140(2), 04013006.

Tsukimoto, H. (2000). Extracting rules from trained neural networks. IEEE Transactions on Neural Networks, 11(2), 377-389.

Turkan, Y., Hong, J., Laflamme, S., & Puri, N. (2018). Adaptive wavelet neural network for terrestrial laser scanner-based crack detection. Automation in Construction, 94, 191-202.

Tuvayanond, W., & Parnichkun, M. (2017). Position control of a pneumatic surgical robot using PSO based 2-DOF H∞ loop shaping structured controller. Mechatronics, 43, 40-55.

Uddin, W., Meyer, A., Hudson, W., & Stokoe, I. (1985). Project-Level Structural Evaluation of Pavements Based on Dynamic Deflections. Transportation Research Record(1007).

Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. International journal of computer vision, 104(2), 154-171.

Vaillant, R., Monrocq, C., & Le Cun, Y. (1994). Original approach for the localisation of objects in images. IEE Proceedings-Vision, Image and Signal Processing, 141(4), 245-250.

Varadharajan, S., Jose, S., Sharma, K., Wander, L., & Mertz, C. (2014, March). Vision for road inspection. In Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on (pp. 115-122). IEEE.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103). ACM.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1990). Phoneme recognition using time-delay neural networks. Readings in speech recognition, Elsevier, 393-404.

Wang, K. C., Li, Q., & Gong, W. (2007). Wavelet-based pavement distress image edge detection with a trous algorithm. Transportation Research Record, 2024(1), 73-81.

Wang, K. C., Li, Q. J., Yang, G., Zhan, Y., & Qiu, Y. (2015). Network level pavement evaluation with 1 mm 3D survey system. Journal of Traffic and Transportation Engineering (English Edition), 2(6), 391-398.

Wu, L., Mokhtari, S., Nazef, A., Nam, B., & Yun, H.-B. (2014). Improvement of crack-detection accuracy using a novel crack defragmentation technique in image-based road assessment. Journal of Computing in Civil Engineering, 30(1), 04014118.

Wang, K. C., Hou, Z., & Williams, S. (2010). Precision test of cracking surveys with the automated distress analyzer. Journal of Transportation Engineering, 137(8), 571-579.

Wang, K. C., Zhang, A., Li, J. Q., Fei, Y., Chen, C., & Li, B. (2017). Deep Learning for Asphalt Pavement Cracking Recognition Using Convolutional Neural Network. In Airfield and Highway Pavements 2017 (pp. 166-177).

Wang, L., Zhuang, L., & Zhang, Z. (2019). Automatic Detection of Rail Surface Cracks with a Superpixel-Based Data-Driven Framework. Journal of Computing in Civil Engineering, 33(1).

Wang, S., Qiu, S., Wang, W., Xiao, D., & Wang, K. C. P. (2017). Cracking Classification Using Minimum Rectangular Cover-Based Support Vector Machine. Journal of Computing in Civil Engineering, 31(5).

Wang, T., Gopalakrishnan, K., Smadi, O., & Somani, A. K. (2018). Automated shape-based pavement crack detection approach. Transport, 33(3), 598-608.

Wu, C., Lu, B., Chen, D., & Wang, L. (2011, July). Pavement image denoising based on shearlet treansform. In Electronics and Optoelectronics (ICEOE), 2011 International Conference on (Vol. 3, pp. V3-262). IEEE.

Wu, S., & Liu, Y. (2012, June). A segment algorithm for crack dection. In Electrical & Electronics Engineering (EEESYM), 2012 IEEE Symposium on (pp. 674-677). IEEE.

Wu, Z., Flintsch, G., Ferreira, A., & Picado-Santos, L. d. (2012). Framework for multiobjective

optimization of physical highway assets investments. Journal of Transportation Engineering, 138(12), 1411-1421.

Xinchao, Z. (2010). A perturbed particle swarm algorithm for numerical optimization. Applied Soft Computing, 10(1), 119-124.

Xu, J., Luo, X., Wang, G., Gilmore, H., & Madabhushi, A. (2016). A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. Neurocomputing, 191, 214-223.

Xu, G., & Yu, G. (2018). On convergence analysis of particle swarm optimization algorithm. Journal of Computational and Applied Mathematics, 333, 65-73.

Xue, Y., & Li, Y. (2018). A Fast Detection Method via Region-Based Fully Convolutional Neural Networks for Shield Tunnel Lining Defects. Computer-Aided Civil and Infrastructure Engineering, 33(8), 638-654.

Yan, B., Goto, S., Miyamoto, A., & Zhao, H. (2014). Imaging-based rating for corrosion states of weathering steel using wavelet transform and PSO-SVM techniques. Journal of Computing in Civil Engineering, 28(3).

Yang, G., Li, Q. J., Zhan, Y., Fei, Y., & Zhang, A. (2018). Convolutional Neural Network-Based Friction Model Using Pavement Texture Data. Journal of Computing in Civil Engineering, 32(6).

Yamaguchi, T., Nakamura, S., Saegusa, R., & Hashimoto, S. (2008). Image-based crack detection for real concrete surfaces. IEEJ Transactions on Electrical and Electronic Engineering, 3(1), 128-135.

Yao, M., Zhao, Z., Yao, X., & Xu, B. (2015). Fusing complementary images for pavement cracking measurements. Measurement Science and Technology, 26(2), 025005.

Yeum, C. M., Dyke, S. J., & Ramirez, J. (2018). Visual data classification in post-event building reconnaissance. Engineering Structures, 155, 16-24.

Ying, L. & Salari, E. (2010). Beamlet transform-based technique for pavement crack detection and classification. Computer-Aided Civil and Infrastructure Engineering, 25 (8), 572-580.

Yu, S.-N., Jang, J.-H. & Han, C.-S. (2007). Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. Automation in Construction, 16 (3), 255-261.

Zalama, E., Gómez-García-Bermejo, J., Medina, R., & Llamas, J. (2014). Road crack detection using visual features extracted by Gabor filters. Computer-Aided Civil and Infrastructure Engineering, 29(5), 342-358.

Zhang, A. & Wang, K.C. (2017). The fast prefix coding algorithm (fpca) for 3d pavement surface data compression. Computer-Aided Civil and Infrastructure Engineering, 32 (3), 173-190.

Zhang, A., Wang, K. C., Ji, R., & Li, Q. J. (2016). Efficient system of cracking-detection algorithms with 1-mm 3D-surface models and performance measures. Journal of Computing in Civil Engineering, 30(6), 04016020.

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.

Ziari, H., Sobhani, J., Ayoubinejad, J., & Hartmann, T. (2016). Prediction of IRI in short and long terms for flexible pavements: ANN and GMDH methods. International journal of pavement engineering, 17(9), 776-788.

Zhang, X., Jiang, D., Long, Q., & Han, T. (2017). Rotating machinery fault diagnosis for imbalanced data based on decision tree and fast clustering algorithm. Journal of Vibroengineering, 19(6).

Zhou, J., Huang, P. S., & Chiang, F. P. (2006). Wavelet-based pavement distress detection and evaluation. Optical Engineering, 45(2), 027007.

Zhang, D., Qu, S., He, L., & Shi, S. (2009). Automatic ridgelet image enhancement algorithm for road crack image based on fuzzy entropy and fuzzy divergence. Optics and Lasers in Engineering, 47(11), 1216-1225.

Zakeri, H., Nejad, F. M., Fahimifar, A., Torshizi, A. D., & Zarandi, M. F. (2013, June). A multi-stage expert system for classification of pavement cracking. In IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint (pp. 1125-1130). IEEE.

Zakeri, H., Nejad, F. M., & Fahimifar, A. (2017). Image based techniques for crack detection,

classification and quantification in asphalt pavement: a review. Archives of Computational Methods in Engineering, 24(4), 935-977.

Zhou, H., Yang, S., & Zhu, J. (2010, September). Illumination invariant enhancement and threshold segmentation algorithm for asphalt pavement crack image. In Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on (pp. 1-4). IEEE.

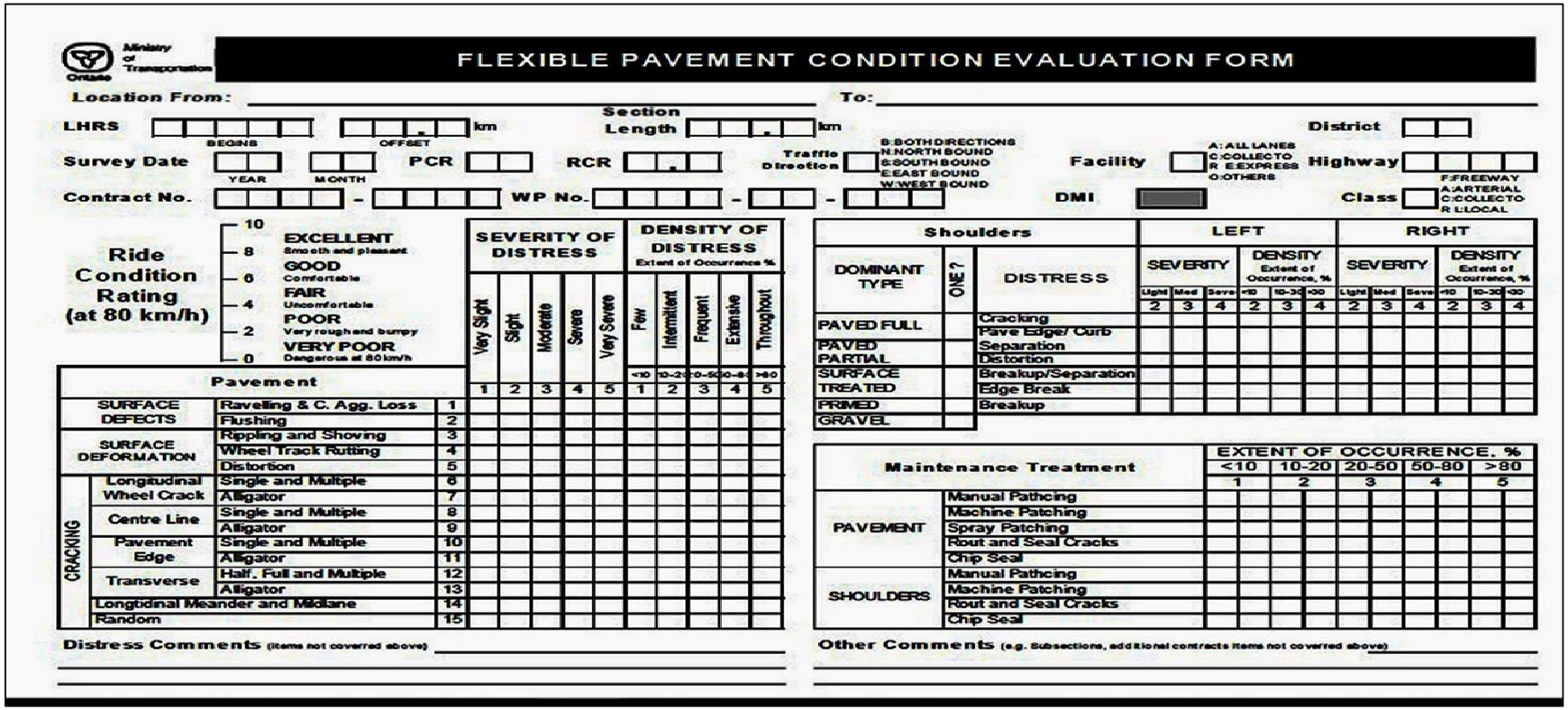
Zhou, J., Huang, P., & Chiang, F. P. (2005). Wavelet-based pavement distress classification. Transportation Research Record: Journal of the Transportation Research Board, (1940), 89-98.

Zhang, J., Sha, A., Sun, Z. Y., & Gao, H. G. (2009). Pavement crack automatic recognition based on wiener filtering. In ICCTP 2009: Critical Issues In Transportation Systems Planning, Development, and Management (pp. 1-7).

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. Pattern Recognition Letters, 33(3), 227-238.

Appendix I

a. Flexible pavement condition evaluation form



Ministry of Transportation Ontario

FLEXIBLE PAVEMENT CONDITION EVALUATION FORM

Location From: _____ To: _____

LHRS (BEGIN) km (OFFSET) Section Length km District

Survey Date (YEAR) (MONTH) PCR RCR Traffic Direction (B:BOTH DIRECTIONS N:NORTH BOUND S:SOUTH BOUND E:EAST BOUND W:WEST BOUND) Facility (A:ALL LANES C:COLLECTOR E:EXPRESS O:OTHERS) Highway (F:FREEWAY A:ARTERIAL C:COLLECTOR L:LOCAL)

Contract No. WP No. DMI Class

Ride Condition Rating (at 80 km/h)

10	EXCELLENT – Smooth and pleasant
8	GOOD – Comfortable
6	FAIR – Uncomfortable
4	POOR – Very rough and bumpy
2	VERY POOR – Dangerous at 80 km/h
0	

Pavement			SEVERITY OF DISTRESS: Very Slight	Slight	Moderate	Severe	Very Severe	DENSITY OF DISTRESS (Extent of Occurrence %): Few	Intermittent	Frequent	Extensive	Throughout
			1	2	3	4	5	<10	10-20	20-50	50-80	>80
SURFACE DEFECTS	Raveling & C. Agg. Loss	1										
	Flushing	2										
SURFACE DEFORMATION	Rippling and Shoving	3										
	Wheel Track Rutting	4										
	Distortion	5										
CRACKING – Longitudinal Wheel Crack	Single and Multiple	6										
	Alligator	7										
Centre Line	Single and Multiple	8										
	Alligator	9										
Pavement Edge	Single and Multiple	10										
	Alligator	11										
Transverse	Half, Full and Multiple	12										
	Alligator	13										
Longtidinal Meander and Midlane		14										
Random		15										

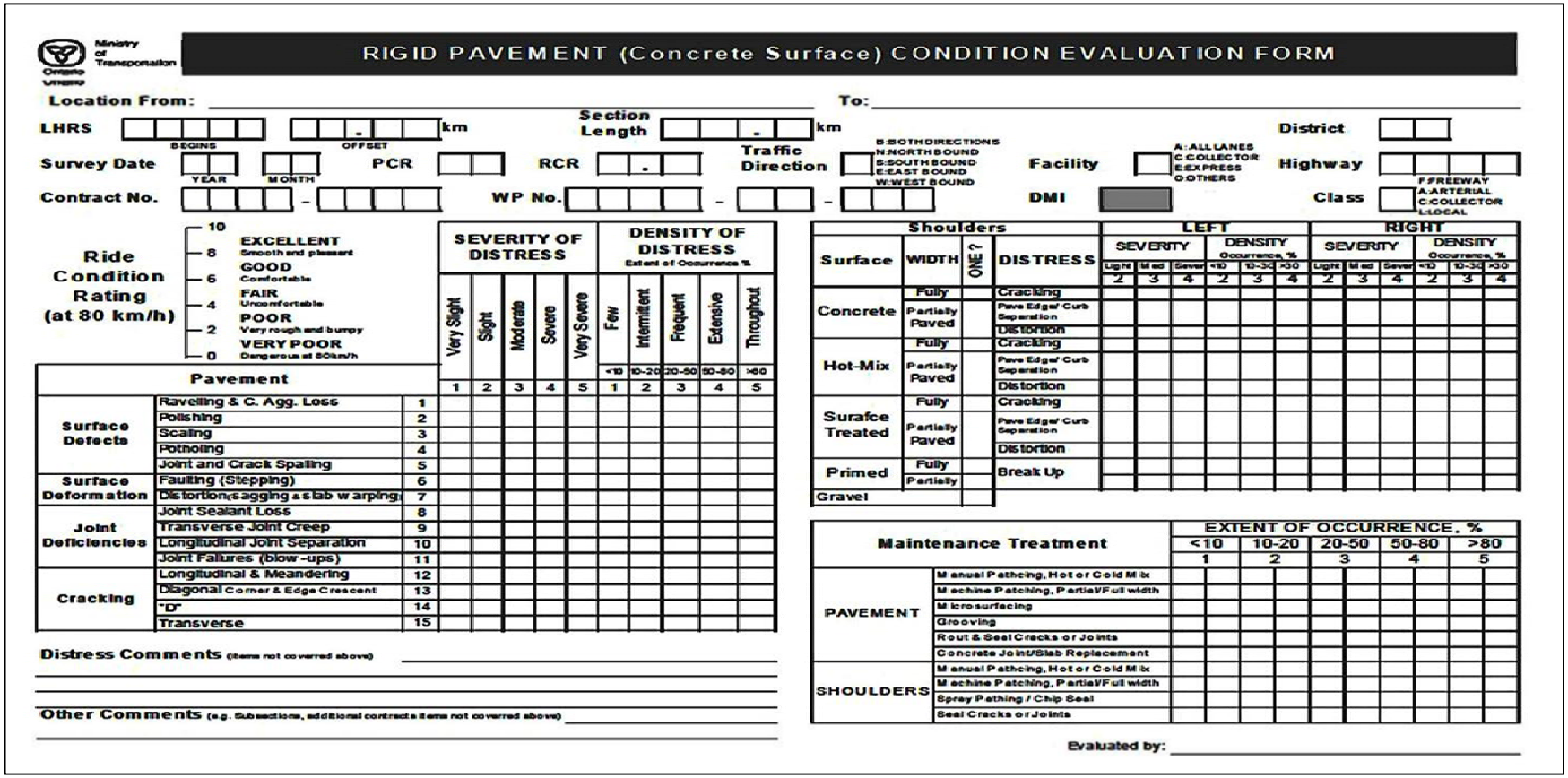
Distress Comments (Items not covered above) _____

Shoulders			LEFT SEVERITY: Light	Med	Severe	LEFT DENSITY (Extent of Occurrence, %): <10	10-30	>30	RIGHT SEVERITY: Light	Med	Severe	RIGHT DENSITY (Extent of Occurrence, %): <10	10-30	>30
DOMINANT TYPE	ONE?	DISTRESS	2	3	4	2	3	4	2	3	4	2	3	4
PAVED FULL		Cracking												
PAVED PARTIAL		Pave Edge/ Curb Separation												
SURFACE TREATED		Distortion												
PRIMED		Breakup/Separation												
GRAVEL		Edge Break												
		Breakup												

Maintenance Treatment		EXTENT OF OCCURRENCE, %: <10	10-20	20-50	50-80	>80
		1	2	3	4	5
PAVEMENT	Manual Patching					
	Machine Patching					
	Spray Patching					
	Rout and Seal Cracks					
	Chip Seal					
SHOULDERS	Manual Patching					
	Machine Patching					
	Rout and Seal Cracks					
	Chip Seal					

Other Comments (e.g. Subsections, additional contracts items not covered above) _____

b. Rigid pavement condition evaluation form



Ministry of Transportation

RIGID PAVEMENT (Concrete Surface) CONDITION EVALUATION FORM

Location From: _____ To: _____

LHRS [ ][ ][ ][ ][ ] BEGINS [ ][ ][.][ ][ ] km OFFSET

Section Length [ ][ ][.][ ] km

District [ ][ ]

Survey Date [ ][ ] YEAR [ ][ ] MONTH

PCR [ ][ ] RCR [ ][.][ ]

Traffic Direction [ ] B:BOTH DIRECTIONS N:NORTH BOUND S:SOUTH BOUND E:EAST BOUND W:WEST BOUND

Facility [ ] A:ALL LANES C:COLLECTOR E:EXPRESS O:OTHERS

Highway [ ][ ][ ][ ]

Contract No. [ ][ ][ ][ ] - [ ][ ][ ][ ]

WP No. [ ][ ][ ][ ] - [ ][ ] - [ ][ ][ ]

DMI [ ]

Class [ ] F:FREEWAY A:ARTERIAL C:COLLECTOR L:LOCAL

Ride Condition Rating (at 80 km/h)

- 10 EXCELLENT — Smooth and pleasant
- 8 GOOD — Comfortable
- 6 FAIR — Uncomfortable
- 4 POOR — Very rough and bumpy
- 2 VERY POOR — Dangerous at 80km/h
- 0

Pavement			SEVERITY OF DISTRESS: Very Slight	Slight	Moderate	Severe	Very Severe	DENSITY OF DISTRESS (Extent of Occurrence %): Few <10	Intermittent 10-20	Frequent 20-50	Extensive 50-80	Throughout >80
			1	2	3	4	5	1	2	3	4	5
Surface Defects	Ravelling & C. Agg. Loss	1										
	Polishing	2										
	Scaling	3										
	Potholing	4										
	Joint and Crack Spalling	5										
Surface Deformation	Faulting (Stepping)	6										
	Distortion(Sagging & Slab warping)	7										
Joint Deficiencies	Joint Sealant Loss	8										
	Transverse Joint Creep	9										
	Longitudinal Joint Separation	10										
	Joint Failures (blow-ups)	11										
Cracking	Longitudinal & Meandering	12										
	Diagonal Corner & Edge Crescent	13										
	"D"	14										
	Transverse	15										

Distress Comments (Items not covered above) _____

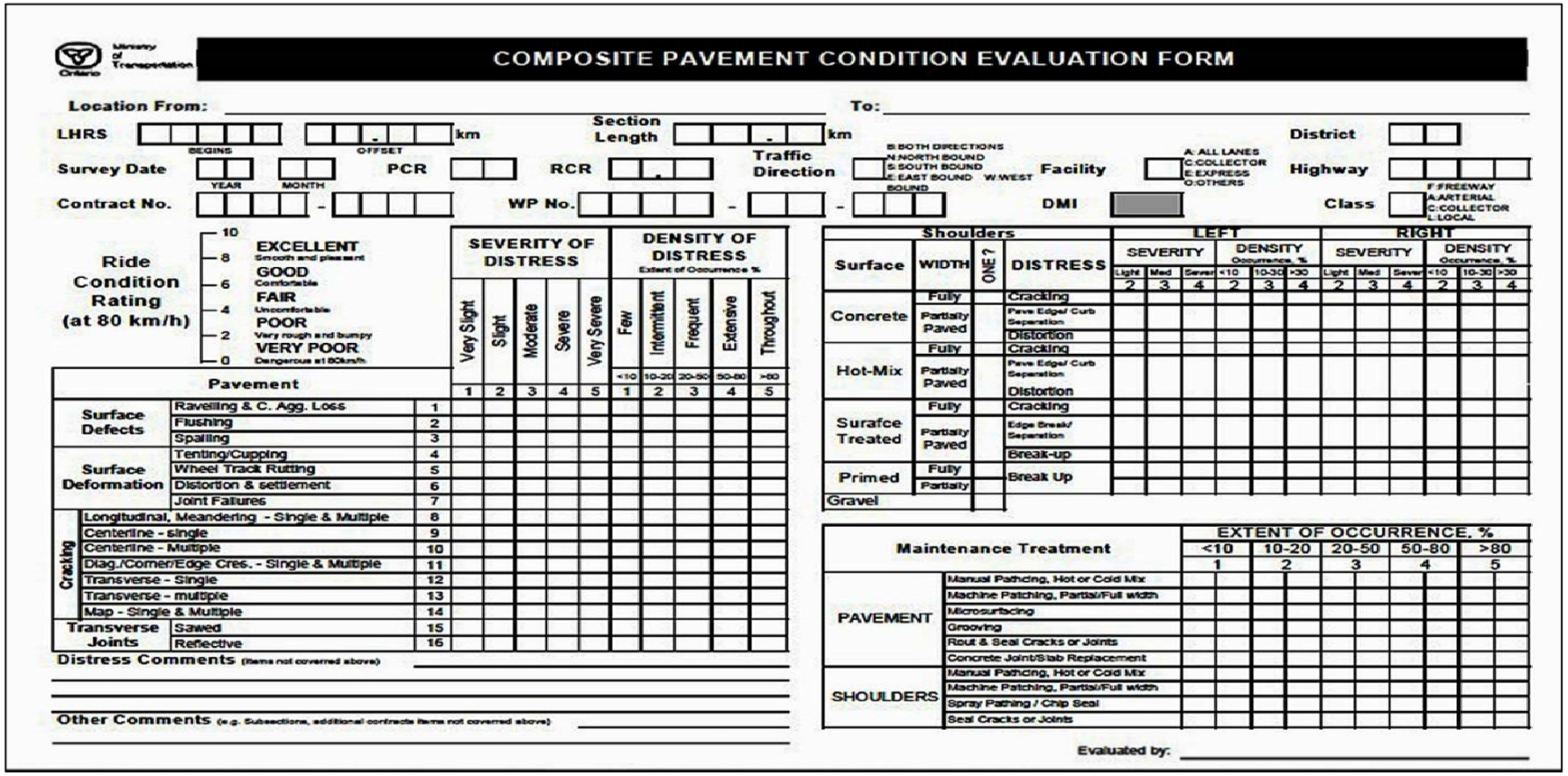
Other Comments (e.g. Subsections, additional contracts items not covered above) _____

Shoulders: Surface	WIDTH	ONE?	DISTRESS	LEFT SEVERITY: Light 2	Med 3	Severe 4	LEFT DENSITY (Occurrence, %): <10 2	10-30 3	>30 4	RIGHT SEVERITY: Light 2	Med 3	Severe 4	RIGHT DENSITY (Occurrence, %): <10 2	10-30 3	>30 4
Concrete	Fully		Cracking												
	Partially Paved		Pave Edge/ Curb Separation												
			Distortion												
Hot-Mix	Fully		Cracking												
	Partially Paved		Pave Edge/ Curb Separation												
			Distortion												
Surafce Treated	Fully		Cracking												
	Partially Paved		Pave Edge/ Curb Separation												
			Distortion												
Primed	Fully		Break Up												
	Partially														
Gravel															

Maintenance Treatment		EXTENT OF OCCURRENCE, %: <10 1	10-20 2	20-50 3	50-80 4	>80 5
PAVEMENT	Manual Patching, Hot or Cold Mix					
	Machine Patching, Partial/Full width					
	Microsurfacing					
	Grooving					
	Rout & Seal Cracks or Joints					
	Concrete Joint/Slab Replacement					
SHOULDERS	Manual Patching, Hot or Cold Mix					
	Machine Patching, Partial/Full width					
	Spray Pathing / Chip Seal					
	Seal Cracks or Joints					

Evaluated by: _____

c. Composite pavement condition evaluation form



Ministry of Transportation Ontario

COMPOSITE PAVEMENT CONDITION EVALUATION FORM

Location From: _____ To: _____

LHRS (BEGIN) ____ (OFFSET) ____ km Section Length ____ km District ____

Survey Date (YEAR) ____ (MONTH) ____ PCR ____ RCR ____ Traffic Direction (B: BOTH DIRECTIONS, N: NORTH BOUND, S: SOUTH BOUND, E: EAST BOUND, W: WEST BOUND) ____ Facility (A: ALL LANES, C: COLLECTOR, E: EXPRESS, O: OTHERS) ____ Highway ____

Contract No. ____ - ____ WP No. ____ - ____ - ____ DMI ____ Class (F: FREEWAY, A: ARTERIAL, C: COLLECTOR, L: LOCAL) ____

Ride Condition Rating (at 80 km/h)

10	EXCELLENT — Smooth and pleasant
8	GOOD — Comfortable
6	FAIR — Uncomfortable
4	POOR — Very rough and bumpy
2	VERY POOR — Dangerous at 80km/h
0	

Pavement			SEVERITY OF DISTRESS: Very Slight (1)	Slight (2)	Moderate (3)	Severe (4)	Very Severe (5)	DENSITY OF DISTRESS (Extent of Occurrence %): Few <10 (1)	Intermittent 10-20 (2)	Frequent 20-50 (3)	Extensive 50-80 (4)	Throughout >80 (5)
Surface Defects	Ravelling & C. Agg. Loss	1										
	Flushing	2										
	Spalling	3										
Surface Deformation	Tenting/Cupping	4										
	Wheel Track Rutting	5										
	Distortion & settlement	6										
	Joint Failures	7										
Cracking	Longitudinal, Meandering - Single & Multiple	8										
	Centerline - single	9										
	Centerline - Multiple	10										
	Diag./Corner/Edge Cres. - Single & Multiple	11										
	Transverse - Single	12										
	Transverse - multiple	13										
	Map - Single & Multiple	14										
Transverse Joints	Sawed	15										
	Reflective	16										

Distress Comments (Items not covered above) _____

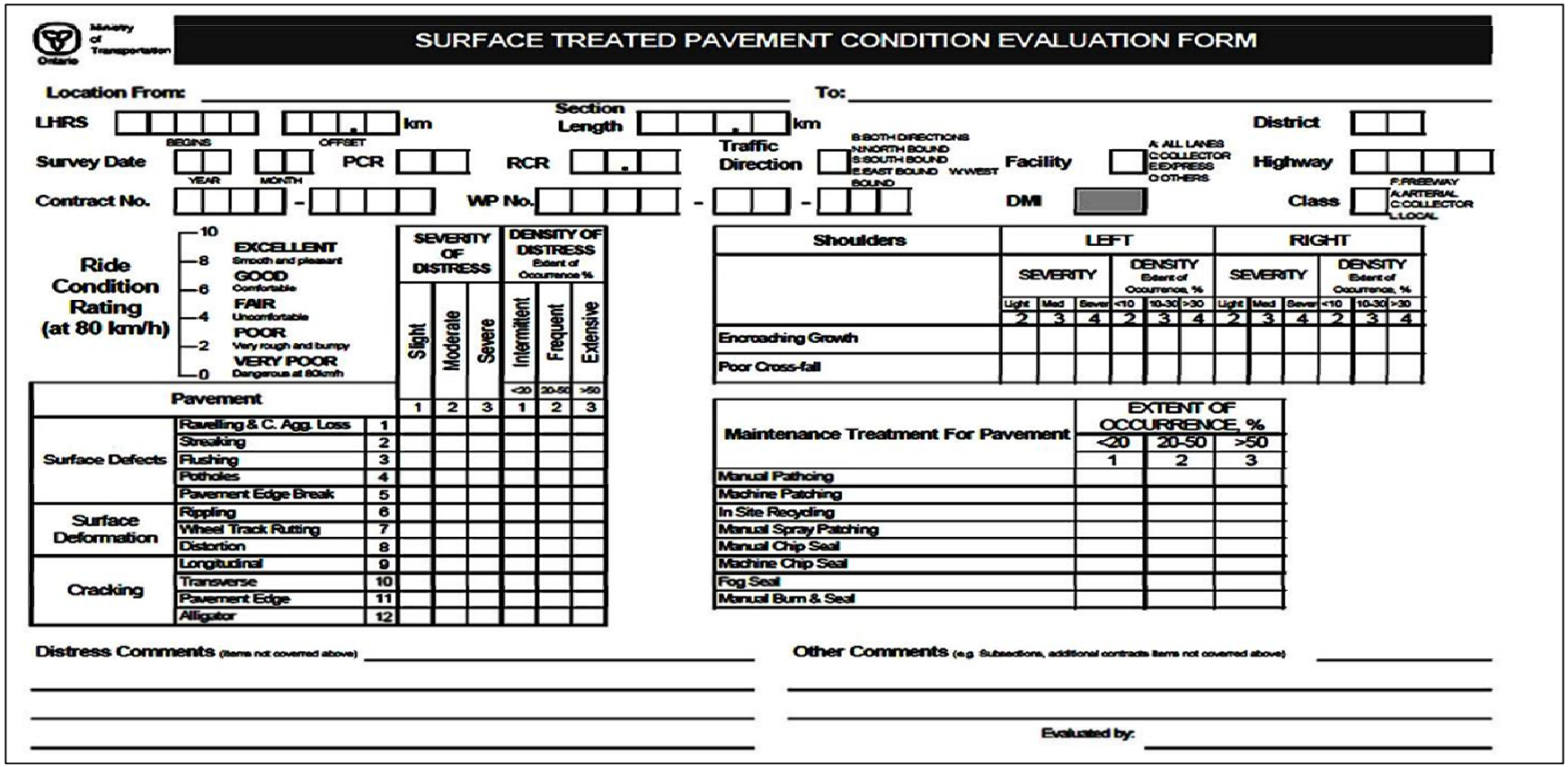
Other Comments (e.g. Subsections, additional contracts items not covered above) _____

Shoulders				LEFT SEVERITY: Light (2)	Med (3)	Severe (4)	LEFT DENSITY (Occurrence %): <10 (2)	10-30 (3)	>30 (4)	RIGHT SEVERITY: Light (2)	Med (3)	Severe (4)	RIGHT DENSITY (Occurrence %): <10 (2)	10-30 (3)	>30 (4)
Surface	WIDTH	ONE?	DISTRESS												
Concrete	Fully		Cracking												
	Partially Paved		Pave Edge/ Curb Separation												
			Distortion												
Hot-Mix	Fully		Cracking												
	Partially Paved		Pave Edge/ Curb Separation												
			Distortion												
Surafce Treated	Fully		Cracking												
	Partially Paved		Edge Break/ Separation												
			Break-up												
Primed	Fully		Break Up												
	Partially														
Gravel															

Maintenance Treatment		EXTENT OF OCCURRENCE, %: <10 (1)	10-20 (2)	20-50 (3)	50-80 (4)	>80 (5)
PAVEMENT	Manual Patching, Hot or Cold Mix					
	Machine Patching, Partial/Full Width					
	Microsurfacing					
	Grooving					
	Rout & Seal Cracks or Joints					
	Concrete Joint/Slab Replacement					
SHOULDERS	Manual Patching, Hot or Cold Mix					
	Machine Patching, Partial/Full Width					
	Spray Pathing / Chip Seal					
	Seal Cracks or Joints					

Evaluated by: _____

d. Surface treated pavement condition evaluation form



Ministry of Transportation Ontario

SURFACE TREATED PAVEMENT CONDITION EVALUATION FORM

Location From: _____ To: _____

LHRS ____ BEGINS ____ . ____ OFFSET km Section Length ____ . ____ km District ____

Survey Date ____ YEAR ____ MONTH PCR ____ RCR ____ . ____ Traffic Direction ____ B:BOTH DIRECTIONS N:NORTH BOUND S:SOUTH BOUND E:EAST BOUND W:WEST BOUND Facility ____ A: ALL LANES C:COLLECTOR E:EXPRESS O:OTHERS Highway ____

Contract No. ____ - ____ WP No. ____ - ____ - ____ DMI ____ Class ____ F:FREEWAY A:ARTERIAL C:COLLECTOR L:LOCAL

Ride Condition Rating (at 80 km/h)

- 10 EXCELLENT — Smooth and pleasant
- 8 GOOD — Comfortable
- 6 FAIR — Uncomfortable
- 4 POOR — Very rough and bumpy
- 2 VERY POOR — Dangerous at 80km/h
- 0

Pavement			SEVERITY OF DISTRESS: Slight	Moderate	Severe	DENSITY OF DISTRESS (Extent of Occurrence %): Intermittent <30	Frequent 20-50	Extensive >50
			1	2	3	1	2	3
Surface Defects	Ravelling & C. Agg. Loss	1						
	Streaking	2						
	Flushing	3						
	Potholes	4						
	Pavement Edge Break	5						
Surface Deformation	Rippling	6						
	Wheel Track Rutting	7						
	Distortion	8						
Cracking	Longitudinal	9						
	Transverse	10						
	Pavement Edge	11						
	Alligator	12						

Shoulders	LEFT SEVERITY: Light 2	Med 3	Sever 4	LEFT DENSITY (Extent of Occurrence, %): <10 2	10-30 3	>30 4	RIGHT SEVERITY: Light 2	Med 3	Sever 4	RIGHT DENSITY (Extent of Occurrence, %): <10 2	10-30 3	>30 4
Encroaching Growth												
Poor Cross-fall												

Maintenance Treatment For Pavement	EXTENT OF OCCURRENCE, %: <20 1	20-50 2	>50 3
Manual Pathcing			
Machine Patching			
In Site Recycling			
Manual Spray Patching			
Manual Chip Seal			
Machine Chip Seal			
Fog Seal			
Manual Burn & Seal			

Distress Comments (Items not covered above) _____

Other Comments (e.g. Subsections, additional contracts Items not covered above) _____

Evaluated by: _____

Appendix II

a. IRI MIVs

Indicator	IRI contribution (MIV)
Wheelpath_Sev_(%)	0.0495
Wheelpath_Pattern_Severe_CrackWidth	0.0488
Wheelpath_Longitudinal_Severe_CrackWidth	0.0485
Transverse_Sev_%	0.0481
Wheelpath_Pattern_Sev_%	0.0468
Wheelpath_Longitudinal_Severe_CrackLength	0.0435
Wheelpath_Longitudinal_Moderate_CrackWidth	0.0426
Wheelpath_Pattern_Moderate_CrackWidth	0.0416
Wheelpath_Longitudinal_Moderate_CrackLength	0.0413
Wheelpath_Mod_(%)	0.0413
Wheelpath_Slight_ (%)	0.0403
Wheelpath_Longitudinal_Slight_CrackCount	0.0402
Wheelpath_Pattern_Slight_%	0.0396
Wheelpath_Longitudinal_Moderate_CrackCount	0.0394
Wheelpath_Pattern_Slight_CrackArea	0.0374
Wheelpath_Longitudinal_Slight_CrackLength	0.0373
Wheelpath_Pattern_Moderate_CrackArea	0.0366
Wheelpath_Pattern_Mod_%	0.0351
Wheelpath_Pattern_Slight_CrackWidth	0.0334

Wheelpath_Longitudinal_Slight_CrackWidth	0.0321
Edge_Pattern_Slight_CrackWidth	0.014
Transverse_Slight_CrackCount	0.014
Transverse_Moderate_CrackLength	0.0096
Midlane_Pattern_Sev_%	0.0096
Midlane_Pattern_Severe_CrackArea	0.0096
Midlane_Pattern_Severe_CrackWidth	0.0096
Midlane_Pattern_Slight_CrackWidth	0.0081
Transverse_Slight_CrackLength	0.0079
Transverse_Slight_%	0.0079
Midlane_Pattern_Slight_CrackArea	0.0079
Midlane_Pattern_Slight_%	0.0079
PavtEdge_Pattern_Mod_%	0.0074
Transverse_Mod_%	0.0074
Transverse_Moderate_CrackWidth	0.0056
Transverse_Slight_CrackWidth	0.0053
Ravelling_Slight_Percent	0.0051
Ravelling_Moderate_RI	0.0044
Midlane_Pattern_Moderate_CrackWidth	0.0041
Transverse_Moderate_CrackCount	0.0035
Combine Long Crk_Mod_%	0.0033
CombinedZones_Longitudinal_Moderate_CrackLength	0.0033
CombinedZones_Longitudinal_Moderate_CrackWidth	0.0028
Midlane_Pattern_Moderate_CrackArea	0.0023
Midlane_Pattern_Mod_%	0.0023
Centerline_Pattern_Slight_CrackWidth	0.0022
CombinedZone_Longitudinal_Moderate_CrackCount	0.0019

Edge_Pattern_Slight_CrackArea	0.0017
PavtEdge_Pattern_Slight_%	0.0019
Centerline_Pattern_Sev_%	0.0017
Combine Long Crk_Slight_%	0.0016
Edge_Pattern_Moderate_CrackWidth	0.0013
Edge_Pattern_Moderate_CrackArea	0.0008
Centerline_Pattern_Slight_CrackArea	0.0008
Centerline_Pattern_Slight_%	0.0008
Centerline_Pattern_Moderate_CrackWidth	0.0007
Combine Long Crk_Sev_%	0.0005
PavtEdge_Pattern_Sev_%	0.0005
Centerline_Pattern_Moderate_CrackArea	0.0005
Centerline_Pattern_Mod_%	0.0005

b. RUT MIVs

Indicator	RUT contribution (MIV)
Transverse_Moderate_CrackWidth	0.046
Wheelpath_Longitudinal_Severe_CrackWidth	0.0459
Wheelpath_Sev_ (%)	0.0444
Wheelpath_Longitudinal_Severe_CrackLength	0.0433
Wheelpath_Longitudinal_Moderate_CrackWidth	0.0424
Wheelpath_Pattern_Moderate_CrackWidth	0.0408
Transverse_Moderate_CrackLength	0.0401
Midlane_Pattern_Severe_CrackArea	0.0391
Wheelpath_Pattern_Severe_CrackWidth	0.0378
Wheelpath_Pattern_Sev_%	0.0358
Wheelpath_Slight_ (%)	0.0345
Wheelpath_Longitudinal_Slight_CrackLength	0.0343
Transverse_Slight_CrackWidth	0.034
Transverse_Slight_CrackCount	0.0333
Wheelpath_Longitudinal_Slight_CrackWidth	0.0316
Transverse_Slight_%	0.0264
Midlane_Pattern_Slight_CrackWidth	0.0241
Centerline_Pattern_Moderate_CrackWidth	0.024
Wheelpath_Pattern_Slight_CrackArea	0.0221
Centerline_Pattern_Slight_CrackWidth	0.0217
Edge_Pattern_Slight_CrackWidth	0.0188

Midlane_Pattern_Slight_CrackArea	0.016
Midlane_Pattern_Slight_%	0.016
Transverse_Sev_%	0.014
Combine Long Crk_Slight_%	0.0135
Centerline_Pattern_Slight_%	0.0128
Edge_Pattern_Slight_CrackArea	0.0122
PavtEdge_Pattern_Slight_%	0.0118
Midlane_Pattern_Moderate_CrackWidth	0.0105
Midlane_Pattern_Moderate_CrackArea	0.01
Midlane_Pattern_Mod_%	0.0097
Transverse_Mod_%	0.00969
CombinedZones_Longitudinal_Moderate_CrackWidth	0.00965
Midlane_Pattern_Severe_CrackWidth	0.00959
Midlane_Pattern_Sev_%	0.00948
Centerline_Pattern_Sev_%	0.00944
Wheelpath_Longitudinal_Moderate_CrackLength	0.00931
Wheelpath_Longitudinal_Slight_CrackCount	0.00921
Wheelpath_Longitudinal_Moderate_CrackCount	0.00915
Wheelpath_Mod_(%)	0.00906
Wheelpath_Pattern_Mod_%	0.00899
Wheelpath_Pattern_Moderate_CrackArea	0.00875
Centerline_Pattern_Slight_CrackArea	0.00865
Wheelpath_Pattern_Slight_CrackWidth	0.00824
Wheelpath_Pattern_Slight_%	0.00813
CombinedZone_Longitudinal_Moderate_CrackCount	0.00716
CombinedZones_Longitudinal_Moderate_CrackLength	0.00615

Combine Long Crk_Mod_%	0.0052
Wheelpath_Pattern_Slight_CrackArea	0.00512
Transverse_Slight_CrackCount	0.00418
Ravelling_Slight_Percent	0.00397
Edge_Pattern_Moderate_CrackWidth	0.00384
Transverse_Slight_CrackLength	0.00351
Centerline_Pattern_Moderate_CrackArea	0.00286
Centerline_Pattern_Mod_%	0.00157
Combine Long Crk_Sev_%	0.00148
Ravelling_Moderate_RI	0.00127
PavtEdge_Pattern_Mod_%	0.00114
PavtEdge_Pattern_Sev_%	0.00101

c. DMI MIVs

Indicator	DMI Contribution (MIV)
Transverse_Moderate_CrackWidth	0.0783
Wheelpath_Longitudinal_Severe_CrackLength	0.0724
Transverse_Sev_%	0.0717
Wheelpath_Longitudinal_Severe_CrackWidth	0.0717
Midlane_Pattern_Severe_CrackWidth	0.0699
Midlane_Pattern_Sev_%	0.066
Centerline_Pattern_Sev_%	0.0609
Wheelpath_Pattern_Sev_%	0.06
Wheelpath_Pattern_Severe_CrackWidth	0.0545
Midlane_Pattern_Severe_CrackArea	0.0545
Wheelpath_Pattern_Slight_CrackArea	0.0374
Wheelpath_Pattern_Slight_%	0.0374
Midlane_Pattern_Slight_CrackWidth	0.0332
Centerline_Pattern_Slight_CrackWidth	0.031
Wheelpath_Pattern_Moderate_CrackArea	0.0184
Combine Long Crk_Slight_%	0.0176
Edge_Pattern_Slight_CrackWidth	0.0135
Transverse_Moderate_CrackCount	0.0111
Midlane_Pattern_Slight_CrackArea	0.0109
Midlane_Pattern_Slight_%	0.0109
Centerline_Pattern_Slight_CrackArea	0.0104
Centerline_Pattern_Slight_%	0.0104
CombinedZones_Longitudinal_Moderate_CrackWidth	0.0088

Wheelpath_Longitudinal_Moderate_CrackLength	0.0087
Wheelpath_Pattern_Moderate_CrackWidth	0.0069
Transverse_Moderate_CrackLength	0.0062
Transverse_Mod_%	0.0062
Wheelpath_Longitudinal_Moderate_CrackWidth	0.004
CombinedZone_Longitudinal_Moderate_CrackCount	0.0039
Centerline_Pattern_Moderate_CrackArea	0.0038
Centerline_Pattern_Mod_%	0.0038
CombinedZones_Longitudinal_Moderate_CrackLength	0.0035
Combine Long Crk_Mod_%	0.0035
Edge_Pattern_Moderate_CrackArea	0.0028
Wheelpath_Pattern_Mod_%	0.0028
PavtEdge_Pattern_Sev_%	0.0025
Wheelpath_Mod_(%)	0.0025
Wheelpath_Longitudinal_Moderate_CrackCount	0.0024
Edge_Pattern_Slight_CrackArea	0.0023
PavtEdge_Pattern_Slight_%	0.0023
Transverse_Slight_%	0.002
Centerline_Pattern_Moderate_CrackWidth	0.0019
Transverse_Slight_CrackLength	0.0017
Midlane_Pattern_Moderate_CrackWidth	0.0017
Wheelpath_Pattern_Slight_CrackWidth	0.0015
Wheelpath_Sev_(%)	0.0015
Transverse_Slight_CrackWidth	0.0015
Midlane_Pattern_Moderate_CrackArea	0.0012
Midlane_Pattern_Mod_%	0.0012
Wheelpath_Longitudinal_Slight_CrackLength	0.0011

Edge_Pattern_Moderate_CrackWidth	0.0011
Combine Long Crk_Sev_%	8.27E-04
Wheelpath_Longitudinal_Slight_CrackWidth	5.52E-04
Ravelling_Moderate_RI	5.38E-04
Ravelling_Slight_Percent	4.64E-04
Wheelpath_Longitudinal_Slight_CrackCount	4.35E-04
Wheelpath_Slight_ (%)	4.35E-04
Transverse_Slight_CrackCount	4.35E-04
PavtEdge_Pattern_Mod_%	4.04E-04

d. PCI MIVs

Factor	PCI contribution (MIV)
Wheelpath_Pattern_Severe_CrackWidth	0.068423986
Wheelpath_Longitudinal_Moderate_CrackLength	0.068158559
Wheelpath_Pattern_Sev_%	0.062669216
Wheelpath_Longitudinal_Severe_CrackLength	0.062669216
Transverse_Sev_%	0.059602502
Wheelpath_Longitudinal_Severe_CrackWidth	0.05501106
Midlane_Pattern_Severe_CrackArea	0.05501106
Midlane_Pattern_Severe_CrackWidth	0.054485226
Midlane_Pattern_Sev_%	0.047049052
Centerline_Pattern_Sev_%	0.047049052
Midlane_Pattern_Slight_CrackWidth	0.038883013
Wheelpath_Longitudinal_Slight_CrackWidth	0.034898917
Edge_Pattern_Slight_CrackWidth	0.030940906
Transverse_Slight_CrackWidth	0.02953551
Midlane_Pattern_Slight_%	0.021768168
Midlane_Pattern_Slight_CrackArea	0.021768168
Centerline_Pattern_Slight_CrackWidth	0.019720596
Transverse_Moderate_CrackWidth	0.017177892
Wheelpath_Pattern_Moderate_CrackWidth	0.015051278
Wheelpath_Longitudinal_Moderate_CrackCount	0.012793676
CombinedZones_Longitudinal_Moderate_CrackWidth	0.01221948
Transverse_Mod_%	0.010979906
Transverse_Moderate_CrackLength	0.010979906

Transverse_Moderate_CrackCount	0.01023475
Edge_Pattern_Slight_CrackArea	0.008020086
PavtEdge_Pattern_Sev_%	0.007456176
Wheelpath_Longitudinal_Moderate_CrackWidth	0.007266122
Combine Long Crk_Slight_%	0.006605559
Wheelpath_Longitudinal_Slight_CrackCount	0.006238811
PavtEdge_Pattern_Mod_%	0.005811746
Edge_Pattern_Moderate_CrackArea	0.005154526
Wheelpath_Mod_(%)	0.004787617
Wheelpath_Pattern_Mod_%	0.004787617
Wheelpath_Pattern_Moderate_CrackArea	0.004787617
Centerline_Pattern_Moderate_CrackWidth	0.004679963
Centerline_Pattern_Slight_CrackArea	0.003782118
PavtEdge_Pattern_Slight_%	0.003782118
Centerline_Pattern_Slight_%	0.003782118
Wheelpath_Sev_(%)	0.003275326
Wheelpath_Slight_ (%)	0.003275326
Midlane_Pattern_Moderate_CrackWidth	0.003245351
Wheelpath_Longitudinal_Slight_CrackLength	0.00319586
Wheelpath_Pattern_Slight_CrackWidth	0.003100207
Wheelpath_Pattern_Slight_%	0.00307372
CombinedZone_Longitudinal_Moderate_CrackCount	0.002828965
CombinedZones_Longitudinal_Moderate_CrackLength	0.002776061
Combine Long Crk_Mod_%	0.002776061
Wheelpath_Pattern_Slight_CrackArea	0.002701786
Transverse_Slight_CrackCount	0.002701786
Transverse_Slight_%	0.002701786

Midlane_Pattern_Mod_%	0.002174511
Midlane_Pattern_Moderate_CrackArea	0.002174511
Ravelling_Slight_Percent	0.001603133
Edge_Pattern_Moderate_CrackWidth	0.001579331
Transverse_Slight_CrackLength	0.001406739
Centerline_Pattern_Moderate_CrackArea	0.001201315
Centerline_Pattern_Mod_%	0.001201315
Combine Long Crk_Sev_%	0.000976431
Ravelling_Moderate_RI	0.000764052

e. RCI MIVs

Indicator	RCI contribution (MIV)
Transverse_Sev_%	0.0648
Wheelpath_Longitudinal_Severe_CrackWidth	0.0629
Wheelpath_Sev_(%)	0.0629
Wheelpath_Longitudinal_Severe_CrackLength	0.0623
Wheelpath_Pattern_Sev_%	0.0623
Wheelpath_Pattern_Severe_CrackWidth	0.0598
Wheelpath_Longitudinal_Moderate_CrackWidth	0.0468
Wheelpath_Pattern_Moderate_CrackWidth	0.0463
Wheelpath_Longitudinal_Moderate_CrackCount	0.0414
Wheelpath_Longitudinal_Moderate_CrackLength	0.0395
Wheelpath_Mod_(%)	0.0392
Transverse_Slight_%	0.0313
Transverse_Slight_CrackLength	0.0313
Midlane_Pattern_Slight_CrackArea	0.0287
Midlane_Pattern_Slight_%	0.0287
Transverse_Slight_CrackCount	0.0216
Transverse_Moderate_CrackWidth	0.0191
Wheelpath_Longitudinal_Slight_CrackWidth	0.0184
Transverse_Slight_CrackWidth	0.0137
Transverse_Moderate_CrackLength	0.0129
Transverse_Mod_%	0.0129
Edge_Pattern_Slight_CrackArea	0.0128
PavtEdge_Pattern_Slight_%	0.0128

Transverse_Moderate_CrackCount	0.0125
Centerline_Pattern_Slight_CrackWidth	0.0123
Edge_Pattern_Slight_CrackWidth	0.0089
Wheelpath_Pattern_Moderate_CrackArea	0.0089
Wheelpath_Pattern_Slight_CrackWidth	0.0088
Centerline_Pattern_Moderate_CrackWidth	0.0073
Midlane_Pattern_Slight_CrackWidth	0.0072
Wheelpath_Longitudinal_Slight_CrackLength	0.007
CombinedZones_Longitudinal_Moderate_CrackLength	0.0063
Wheelpath_Pattern_Mod_%	0.0063
Ravelling_Slight_Percent	0.0057
Combine Long Crk_Mod_%	0.0057
Midlane_Pattern_Sev_%	0.0055
Midlane_Pattern_Severe_CrackArea	0.0055
Midlane_Pattern_Severe_CrackWidth	0.0055
Wheelpath_Slight_ (%)	0.0054
Wheelpath_Pattern_Slight_%	0.0054
Wheelpath_Pattern_Slight_CrackArea	0.0047
Midlane_Pattern_Moderate_CrackWidth	0.0043
CombinedZone_Longitudinal_Moderate_CrackCount	0.0037
Centerline_Pattern_Sev_%	0.0031
CombinedZones_Longitudinal_Moderate_CrackWidth	0.0031
Edge_Pattern_Moderate_CrackArea	0.003
PavtEdge_Pattern_Mod_%	0.003
Midlane_Pattern_Moderate_CrackArea	0.003
Midlane_Pattern_Mod_%	0.003
Ravelling_Moderate_RI	0.0029

Combine Long Crk_Sev_%	0.0022
Edge_Pattern_Moderate_CrackWidth	0.0019
Combine Long Crk_Slight_%	0.0014
Centerline_Pattern_Slight_CrackArea	0.0011
Centerline_Pattern_Slight_%	0.0011
Ravelling_Moderate_RI	6.65E-04
PavtEdge_Pattern_Sev_%	3.72E-04
Centerline_Pattern_Moderate_CrackArea	2.15E-04
Centerline_Pattern_Mod_%	2.15E-04