

# **ROAD DISTRESS IDENTIFICATION USING DEEP MACHINE LEARNING ALGORITHM**

by

**FARHANA MOZUMDER LIMA**

MASTER OF SCIENCE IN CIVIL ENGINEERING (TRANSPORTATION)



Department of Civil Engineering

**BANGLADESH UNIVERSITY OF ENGINEERING AND  
TECHNOLOGY (BUET)**

**March 2023**

# **ROAD DISTRESS IDENTIFICATION USING DEEP MACHINE LEARNING ALGORITHM**

by

**FARHANA MOZUMDER LIMA**

Student ID No: 0417042406 F

A Thesis Submitted to the Department of Civil Engineering in Partial  
Fulfillment of the Requirement for the Degree Of  
**MASTER OF SCIENCE IN CIVIL ENGINEERING (TRANSPORTATION)**

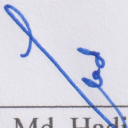
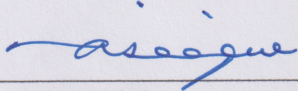
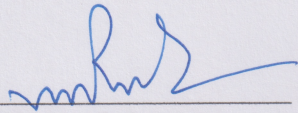
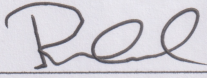


Department of Civil Engineering  
**BANGLADESH UNIVERSITY OF ENGINEERING AND  
TECHNOLOGY (BUET)**

**March 2023**

The thesis titled “ROAD DISTRESS IDENTIFICATION USING DEEP MACHINE LEARNING ALGORITHM” submitted by Farhana Mozumder Lima, Student number 0417042406 (F) and Session April, 2017 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of **Master of Science in Civil Engineering (Transportation)** on March 18, 2023.

**Board of Examiners**

1.   
\_\_\_\_\_  
Dr. Md. Haduzzaman  
Professor  
Department of Civil Engineering,  
BUET, Dhaka - 1000  
Chairman  
(Supervisor)
2.   
\_\_\_\_\_  
Dr. Abu Siddique  
Professor and Head  
Department of Civil Engineering,  
BUET, Dhaka - 1000  
Member  
(Ex-officio)
3.   
\_\_\_\_\_  
Dr. Md. Mizanur Rahman  
Professor  
Department of Civil Engineering,  
BUET, Dhaka - 1000  
Member
4.   
\_\_\_\_\_  
Lt. Col. Mohammad Russeel Islam  
Associate Professor  
Department of Civil Engineering,  
MIST, Dhaka-1216  
Member  
(External)

## DECLARATION

It is hereby declared that, except for the contents where specific references have been made to the work of others, the research work presented in this thesis has been carried out by the author under the supervision of Dr. Md. Hadiuzzaman, Professor, Department of Civil Engineering, BUET. The work presented in this thesis has been done by the author and whole or part of this thesis has not been submitted elsewhere for the award of any degree or diploma.

**March, 2023**

*Farhana Mozumder Lima*

---

Farhana Mozumder Lima  
Student No. 0417042406 (F)

## **ACKNOWLEDGEMENT**

I am deeply indebted to and want to express my utmost gratitude to my thesis supervisor, Dr. Md. Hadiuzzaman, Professor, Department of Civil Engineering, BUET, for his guidance, generous help, constructive comments, invaluable suggestions that have made it possible to prepare this concerted thesis. I would like to thank Nazmul Haque, Lecturer, Accident Research Institute (ARI), BUET, for his untiring support throughout the progress of the thesis work. His endurance, consistent efforts, ceaseless inspiration have helped me to work diligently for the completion of this research work. Without his continuous support and guidance, this thesis would not have materialized.

I want to thank the Department of Civil Engineering, BUET for providing me an opportunity to work on this contemporary thesis work.

I am grateful to my family, friends, and colleagues for their constant encouragement and cooperation in pursuing my goals.

## ABSTRACT

This study focuses on developing an automated road distress identification and classification framework using state-of-art edge detection and deep machine learning technique. In this study, the Convolutional Neural Network (CNN) and Sobel Edge Detection (SED) have been harmonized to automatically identify and classify road distresses using a moving camera. The Sobel Edge Detection techniques have been used to determine the 2-D spatial gradient of a pavement image to identify the distress from it. The pavement image has been converted into machine readable binary image, where the distress area contains a specific identification flag. This identification flag has then been recognized and categorized by blob analysis. A bounding box with measurable dimensions has also been created using the Blob analysis. The image within the bounding box has been given as input in the CNN architecture for classification. A layer of CNNs consists of three main sublayers, which include: convolutional layers, pooling layers, and fully connected layers.

In this research, two types of data have been collected: (i) Static image; and (ii) Dynamic image. Three thousand five hundred static pavement images having a resolution of  $3264 \times 2448$  were collected from different streets and highways within the Dhaka region. The dynamic image dataset is divided into two parts: (a) Synthetic data; and (b) Real-time data. An artificial video using Macromedia MX has been made and used as synthetic data. At the same time, two real-time videos were collected from DIT Road and Dhaka Mymensingh Highway from coordinates  $23.7547^\circ$  N,  $90.4154^\circ$  E to  $24.3654^\circ$  N,  $91.1641^\circ$  E. The static dataset has been enriched using the image augmentation technique. Training of the CNN model is done using the randomly selected static image data. The hyperparameters (i.e., number of layers, number of filters, number of epochs, initial learning rate, and percent of training data) have been tuned using the graphical optimization technique. The synthetic dataset has been used to estimate the initial guess value of the hyperparameters. The optimum parameter value has been found to be: Number of layers = 2, Max Epoch = 9; Training Data = 70%; Learning Rate =  $3.64E-05$ ; Number of Filters = 30. The training and testing accuracy were found to be 99.22% and 98.78%, respectively, with the optimum hyperparameters. The hyperparameter optimization process involved a total of 992 hours of processing time. Confusion Matrix (CM) and Receiver Operating Characteristics (ROC) analysis were done over the trained and tested results and the analysis shows consistent efficacy in classifying each distress

accurately. The results have been compared with the baseline method Support Vector Machine (SVM). The comparison shows that SVM achieved 94.2% and 80.3% accuracy in distress identification and classification, respectively. In contrast, the CNN model has achieved 98.1% and 97.7% accuracy in distress identification and classification, respectively, which shows that the developed CNN model performs better than the baseline method SVM. Pavement Relative Scoring has been conducted which is indicative of pavement health. Finally, A tool is developed named Road Distress Training and Classification (RoadDisTrac) using MATLAB runtime environment.

# TABLE OF CONTENTS

<b>DECLARATION</b> .....	ii
<b>ACKNOWLEDGEMENT</b> .....	iii
<b>ABSTRACT</b> .....	iv
<b>TABLE OF CONTENTS</b> .....	vi
<b>LIST OF FIGURES</b> .....	ix
<b>LIST OF TABLES</b> .....	x
<b>GLOSSARY</b> .....	xi
<b>Chapter 1 INTRODUCTION</b> .....	1
1.1 Background of the Study.....	1
1.1.1 Costly pavement maintenance.....	1
1.1.2 Absence of accurate pavement condition or health monitoring technique.....	3
1.1.3 Inefficient condition monitoring strategies .....	3
1.1.4 Limitations of manual visual survey .....	3
1.2 Research Objectives and Scope of Work.....	4
1.3 Organization of the Thesis .....	5
<b>Chapter 2 LITERATURE REVIEW</b> .....	6
2.1 Introduction .....	6
2.2 Crack Identification.....	6
2.2.1 Cracking group .....	8
2.2.2 Visco-plastic deformation group .....	9
2.2.3 Surface defect group.....	9
2.2.4 Miscellaneous distress .....	9
2.2.5 Automated devices in distress detection.....	9
2.3 Neural Networks .....	10
2.4 Edge Detection .....	15
2.5 Object Detection.....	17
2.6 Feature Extraction .....	19
2.7 Summary .....	23
<b>Chapter 3 METHODOLOGY</b> .....	25
3.1 Introduction .....	25
3.2 Sobel Filter Design.....	26
3.2.1 Method of the filter design .....	27
3.2.2 Pseudo-codes for sobel edge detection method.....	29



3.3 Classification Paradigm: Convolutional Neural Network.....	29
3.3.1 Convolutional neural network .....	30
3.4 Summary .....	36
<b>Chapter 4 DATA COLLECTION AND ANALYSIS.....</b>	<b>37</b>
4.1 Introduction .....	37
4.2 Data Collection and Dataset Preparation .....	37
4.2.1 Synthetic data .....	37
4.2.2 Real-time image data collection .....	41
4.2.2.1 Image resizing .....	41
4.2.2.2 Image augmentation .....	42
4.2.2.3 Data organization.....	42
4.3 Data Analysis .....	43
4.4 Summary .....	48
<b>Chapter 5 OPTIMIZATION OF THE MODEL PARAMETERS .....</b>	<b>49</b>
5.1 Introduction .....	49
5.2 Optimization Process .....	49
5.2.1 Parameters of the model .....	49
5.2.2 Parameter optimization using synthetic data .....	50
5.2.3 Parameter optimization using real time data .....	51
5.3 Summary .....	56
<b>Chapter 6 RESULTS AND DISCUSSION.....</b>	<b>58</b>
6.1 Introduction .....	58
6.2 Road Distress Identification Result.....	58
6.3 Comparison between CNN and the Baseline Method for the Classification Task.....	60
6.4 Field Testing and Pavement Scoring.....	65
6.5 Development of Tool for Practice.....	68
6.6 Summary .....	69
<b>Chapter 7 CONCLUSION AND RECOMMENDATIONS.....</b>	<b>70</b>
7.1 Concluding Remarks .....	70
7.2 Recommendations for Future Research .....	71
<b>REFERENCES.....</b>	<b>73</b>
<b>APPENDIX A-1: FATIGUE CRACKS .....</b>	<b>81</b>
<b>APPENDIX A-2: BLOCK CRACKING .....</b>	<b>90</b>
<b>APPENDIX A-3: CORRUGATION AND SHOVING.....</b>	<b>94</b>
<b>APPENDIX A-4: DEPRESSION.....</b>	<b>100</b>

<b>APPENDIX A-5: JOINT REFLECTION CRACKING .....</b>	<b>104</b>
<b>APPENDIX A-6: LONGITUDINAL CRACKING .....</b>	<b>109</b>
<b>APPENDIX A-7: PATCHING.....</b>	<b>119</b>
<b>APPENDIX A-8: POLISHED AGGREGATE.....</b>	<b>124</b>
<b>APPENDIX A-9: POTHOLES .....</b>	<b>133</b>
<b>APPENDIX A-10: RUTTING .....</b>	<b>166</b>

## LIST OF FIGURES

Figure 2.1: Simple Diagram of Feedforward Neural Network.....	10
Figure 2.2: Simple Diagram of Radial Basis Function Neural Network .....	11
Figure 2.3: Simple Diagram of Kohonen Self-Organizing Neural Network .....	12
Figure 2.4: Simple Diagram of Recurrent Neural Network.....	13
Figure 2.5: Mask for Eight-Connected Connectivity .....	23
Figure 2.6: Illustration of Provisional Labeling by the First Scan. (a) Connected- Component Example. (b) Provisional Labels Assigned after the First Scan.....	23
Figure 3.1: Convolution Process to Create Feature Map.....	31
Figure 3.2: Illustration of CNN Architecture.....	32
Figure 3.3: Pooling Layer .....	32
Figure 3.4: Non-Linear Activation Functions.....	33
Figure 4.1: Process of Generating Synthetic Data.....	41
Figure 5.1: Accuracy vs Training data (%).....	51
Figure 5.2: Classification Accuracy Vs Batch Accuracy For Different Learning Rates	51
Figure 5.3: Max Epoch vs Batch Accuracy for Different Learning Rates .....	52
Figure 5.4: Processing Time Vs Max Epoch for Different Learning Rates .....	53
Figure 5.5: Accuracy vs Learning Rate for Different Max Epoch .....	53
Figure 5.6: Elapsed time vs Learning rate for Different Epoch Numbers .....	54
Figure 5.7: Batch Accuracy Vs Elapsed Time for Different Epoch Numbers .....	54
Figure 5.8: Batch Accuracy vs Elapsed Time for Different Learning Rates.....	55
Figure 5.9: Accuracy vs Filter Number for Optimum Epoch and Learning Rate .....	56
Figure 6.1: Change in Mini-Batch Loss with Iteration.....	60
Figure 6.2: ROC Curve for (a) CNN model (b) and SVM model .....	63
Figure 6.3: Comparison between SVM and CNN Method with respect to ROC and AUC as a Measure of Performance .....	64
Figure 6.4: Crack Detection from the Collected Video Sequence.....	66
Figure 6.5: Interface of RoadDisTrac .....	69

## LIST OF TABLES

Table 4-1: Analysis Result of the Synthetic Data.....	43
Table 4-2: Analysis Result of the Real-Time Data.....	45
Table 5-1: Preliminary Calibrated Parameters Values Using Synthetic Data .....	50
Table 5-2: Optimized Parameters of the Model for both Real-Time data .....	56
Table 6-1: Sample Optimization Output of a CNN Train Process .....	58
Table 6-2: Confusion Matrix for Support Vector Machine (SVM).....	61
Table 6-3: Confusion Matrix for CNN .....	61
Table 6-4: Comparison of Accuracy between CNN and SVM Method for Crack Detection and Crack Classification.....	62
Table 6-5: Crack Classification of Abul Hotel to Kuril Road (Urban City Road) .....	67
Table 6-6: Crack Classification of Kuril Road to Abdullahpur (National Highway).....	67

# GLOSSARY

<b>Term</b>	<b>Definition</b>
<b>2D</b>	Two Dimensional
<b>3D</b>	Three Dimensional
<b>ADB</b>	Asian Development Bank
<b>ADP</b>	Annual Development Plan
<b>ANN</b>	Artificial Neural Network
<b>ARMP</b>	Annual Road Maintenance Plan
<b>ASTM</b>	American Society for Testing and Materials
<b>AUC</b>	Area Under the ROC Curve
<b>Background</b>	An image frame that remains stationary for a considerable period.
<b>Background Model</b>	A model which estimates the background image.
<b>Binary Image</b>	The image having its pixel value either 0 or 1.
<b>Camera Jitter</b>	Vibration of camera.
<b>Camouflage</b>	The foreground having pixel values and patterns near the background.
<b>CNNs</b>	Convolutional Neural Networks
<b>DSP</b>	Digital Signal Processing
<b>DMI</b>	Distance Measurement Instrument
<b>EU</b>	European Union
<b>FN</b>	False Negatives
<b>FP</b>	False Positives
<b>GPS</b>	Global Positioning System
<b>GPU</b>	Graphics Processor Unit
<b>GUI</b>	Graphical User Interface
<b>HDM</b>	Highway Development and Management

<b>IRI</b>	International Roughness Index
<b>ML</b>	Machine Learning
<b>Monochromatic Image</b>	An image having its color distribution from black (0) to white (255).
<b>PCC</b>	Percentage Of Correct Classification
<b>PCI</b>	Pavement Condition Index
<b>Pixel</b>	The smallest unit of an image.
<b>PMS</b>	Pavement Management System
<b>RBF</b>	Radial Basic Functions
<b>Resolution</b>	The total area of the image.
<b>RHD</b>	Roads And Highway Department
<b>ROC Curve</b>	Receiver Operating Characteristic Curve
<b>Run-Time</b>	A term used to indicate ‘during the run of a software.’
<b>SGD</b>	Stochastic Gradient Descent
<b>Shadow</b>	A shadow is a lower-intensity area in the background where an opaque object blocks light from a light source.
<b>SIFT</b>	Scale-Invariant Feature Transform
<b>SVM</b>	Support Vector Machine (SVM)
<b>TN</b>	True Negatives
<b>TP</b>	True Positives

# CHAPTER 1

## INTRODUCTION

---

### **1.1 Background of the Study**

Automatic recognition of various road distresses is of considerable interest since it facilitates preventive road maintenance before cracks and potholes become too severe, leading to economic benefits. The current approach of using human operators to categorize road distresses is both labor-intensive and time-consuming. Especially in Bangladesh, transportation agencies routinely collect road condition data as part of their pavement management activities to maintain pavements cost-effectively. Since pavements tend to deteriorate with time under the influence of repeated traffic loading and environmental variations, accurate condition or health monitoring techniques become critical for the timely detection of distress development in the pavements. Efficient condition monitoring strategies can aid engineers in developing appropriate scheduling of pavement maintenance and repair activities leading to a significant reduction in pavement life-cycle maintenance costs. Nowadays, pavement distress evaluation has come a long way from manual visual surveys to acquiring pavement images using downward looking high-speed digital camera attached to pavement data collection vehicles moving at highway speed (Chambon et al., 2011; Mahler et al., 1991). Currently, many highway agencies are moving towards pavement condition data collection using the so-called 3-D automated survey systems that can acquire high-resolution images containing more information elevation and intensity than 2-D images and are capable of achieving 1-mm crack identification (Barnich and Van Droogenbroeck, 2009). Statement of the problems and opportunities.

#### **1.1.1 Costly pavement maintenance**

Road management authorities, i.e., Roads and Highways Department (RHD), Local Government Engineering Department (LGED), Bangladesh Bridge Authority (BBA), and Dhaka City Corporation (DNCC, DSCC) allocate a large amount of budget for pavement expenditure related to road maintenance. Alarmingly, this expenditure is increasing each year due to the involvement of various direct and indirect costs. The road network administered by the Roads and Highways Department (RHD) consisted of about 22, 476 km of roads in 2022. A substantial amount of money is being spent every

year for repairs and maintenance of these roads. However, funding for maintenance remained insufficient, resulting in many roads not reaching their economic life. Unless periodic maintenance is undertaken regularly, roads may rapidly deteriorate and rehabilitation expenses can become very high. Periodic maintenance practices are inadequate due to a lack of strategic planning, financing, and execution.

The adequacy of funding for road maintenance is a critical issue. The government financed road maintenance from the revenue budget and the Annual Development Plan (ADP). The RHD of the Ministry of Road Transport and Bridges regularly uses project funding in the ADP to supplement its budget for periodic maintenance. Periodic maintenance works financed from the revenue budget and ADP fell short of the amount required to meet the sustainable level stated in the Annual Road Maintenance Plan (ARMP) and reduce the maintenance backlog. As a result, the government is using available funds to improve capacity by rehabilitating the existing road network rather than expanding the road network to meet demand increases. The maintenance backlog is gradually increasing, as the available funds could not meet the 13–16% growth in demand. To reduce the backlog, the government needed to access additional sources of external and domestic funding for maintenance. In order to do so, the government needed to implement several measures, including the following: (i) a policy commitment to prepare road maintenance and periodic maintenance budgets using the Highway Development and Management (HDM-4) model under the ARMP; (ii) transparent budgeting to enable monitoring of periodic maintenance expenditure; (iii) setting the periodic maintenance budget at the level required for asset sustainability; (iv) funding the periodic maintenance budget on a permanent basis from domestic sources; and (v) adequately resourcing the maintenance directorate. Asian Development Bank, ADB's assistance in addressing the backlog was based on the need for the government to improve road conditions and develop sustainable sources of funding from the public and private sectors to finance the maintenance backlog and meet periodic road maintenance requirements.



### **1.1.2 Absence of accurate pavement condition or health monitoring technique**

Existing road management authorities (RHD, LGED, BBA, DNCC, DSCC) use International Roughness Index (IRI) as a measure of pavement condition. The IRI is a widely used measure of the roughness of a longitudinal road surface profile. This index is obtained from the response of a specific quarter-car model traveling on the pavement profile. With increasing IRI values, it denotes decreasing pavement conditions. The extent of IRI value, whether pavement condition is good or bad, varies with the type of road. However, this health monitoring technique suffers from different limitations. Road profiles or surfaces have too much information to be used at a network level – at this level, a simple indicator of pavement condition is needed. Most transportation authorities use the International Roughness Index (IRI), a single number that is intended to provide a measure of pavement condition. However, IRI is not well correlated with pavement remaining service life and the correlation between damage and IRI is poor.

### **1.1.3 Inefficient condition monitoring strategies**

Different road management authorities have accepted the International Roughness index as the parameter for pavement monitoring. However, it was developed as an indicator of passenger comfort as they travel in a vehicle on the pavement. It does not represent precisely which pavements are more vulnerable to damage than others. In addition, RHD has defined the pavement condition according to a range of IRI value, i.e., for the national highway, if the IRI value is within the range of 0-3.9, 4-5.9, 6-7.9, 8-9.9 and greater than 10, they denote pavement condition is good, fair, poor, bad, very bad respectively. However, this single parameter cannot solely represent the actual condition of the pavement.

### **1.1.4 Limitations of manual visual survey**

The essence of digitizing data is to increase speed and efficiency to have more things done in a fraction of a second. Imagine a situation where manual data entry is used in an environment where a high-speed data processing system is installed; it will result in the system becoming slow and with a buildup queue that will defeat the whole process automation to increase speed and system efficiency. Therefore, the manual survey system has some disadvantages. It is undertaken by a human who is not infallible in the

performance of repetitive tasks. When a man faces a tedious job, he is bound to commit errors due to the dynamism of human nature, unlike a machine that follows a set of instructions and never gets tired unless when it breaks down. Lower data entry speed is another drawback of the manual survey. There is no way a man can compete with a machine in terms of processing speed. In a data security situation involving going through millions of items to check for data integrity and data validation to ensure high accuracy of data elements going into a system, the low-speed nature of manual data input will cause a system failure. Imagine a surveillance system where a human is allowed to check every person entering a premise manually; the low speed in reviewing everyone's incoming and outgoing people will create high traffic that may shut down the system. In an environment where consistency of data is crucial to the success of the system, manual data entry service is always a disadvantage as maintaining consistency for humans is a challenging task that must be avoided. One of the advantages of the automated system is the low cost of doing business, but using a manual data entry system means more personnel will be engaged to increase service delivery and each member of the workforce must be trained to meet the basic standard required to perform assigned tasks.

## **1.2 Research Objectives and Scope of Work**

This study is concerned with the development of a model that detect the distress from image. The specific objectives are:

- a) Detecting road distress from real-time video using dynamic illumination adaptive background model and foreground segmentation formulation.
- b) Developing algorithm to extract crack features (i.e., orientation, length, density, displacement, location), crack maps, and summary statistics.
- c) Developing a classification paradigm to identify different types of road distress.
- d) Developing a tool for real-time road distress detection and providing pavement health score.

It is expected that the outcome of this research will contribute into automatic road distress management strategies along with continuous road health monitoring.

### **1.3 Organization of the Thesis**

Chapter 1 gives an introduction of the relevant research background, statement of problems as well as the objectives and scope of this research.

Chapter 2 presents comprehensive reviews of previous works on crack identification, edge detection and Neural Network models. The Neural Network models are reviewed with respect to their categories in terms of level of detail and data collection methodology.

Chapter 3 shows the methodology of crack identification and crack classification. It includes description of the mechanism behind the Sobel Edge Filter design for crack identification. This chapter also describes methodology for the classification paradigm incorporating Convolutional Neural Network.

Chapter 4 presents a detailed description of data collection methodology and extensive analysis over the collected data. Pictorial data for calibrating the classification model with crack data using CNN Architecture is presented here.

Chapter 5 includes the calibration and validation of the crack classification model. Sensitivity analysis of different parameters have been done and a set of optimum parameters is obtained.

Chapter 6 includes a discussion on the obtained result. The results of the calibrated model is compared with another State-of-Art image classification technique in terms of several measures of performance.

Chapter 7 concludes the thesis with recommendations and future research direction.

## **CHAPTER 2**

### **LITERATURE REVIEW**

---

#### **2.1 Introduction**

Pavements are primary elements of a transportation structure. A nation's highway network is supposed to be capable enough to maximize trade, industrial and social benefits. In order to ensure the required pavement performance, periodic road health monitoring surveys are indispensable to collect information about pavement conditions. Conventionally, road inspection surveys have been performed manually. However, with the ever-increasing growth of road networks and the amount of traffic they handle, the time-consuming visual analysis techniques are not preferred any more. In addition, manual highway surveys also suffer heavily from the associated subjectivity of human decision making. In contrast, automated pavement surveying systems, when designed and validated appropriately, are able to be very fast, accurate and remove the subjectivity involved.

Cracks are generally produced by the partial or entire fractures of the pavement surface. There is an extensive variety of road distresses starting from single crack (longitudinal and traverse crack) to interrelated crack patterns spread over the entire pavement surface (block, crescent-shaped and crocodile cracks). The load-spreading and water-resistant capacity of the road is lost due to these cracks which also speed up the deterioration process of pavement surface. If road distresses are kept untreated, then the consequences become more severe and these cracks transform into potholes, deform the road and sometimes even produce differential settlement of road. Both 2D and 3D imaging techniques have been used for distress measurements. Usually, 2D imaging can be performed by simpler hardware, hence is generally preferred. But 3D imaging becomes irreplaceable in certain situations, e.g. depth measurements. The automated detection of road distresses, the subject of this research, is a topical issue with numerous publications in recent years.

#### **2.2 Crack Identification**

The first steps for a road maintenance assessment is distress labeling and quantification of pavement in term of type, severity, and extension. This phase sometimes become very

difficult because of a lack of standardization in the distress definition that could lead to the inconsistency of the classification.

In literature, there are several distress identifications catalogues by different researchers and organizations (Shahin, 1994; ASTM, 2018), which almost adopt the same identification and evaluation criteria. One of the most well-known and diffused references is the ASTM D6433-16 (Coenen and Golroo, 2017), which provides distress criteria identification and classification for both flexible and rigid pavement. Another American reference that is commonly used is the manual for the Long-Term Pavement Performance Program by the US Department of Transportation (Chambon and Moliard, 2011), which aims to collect pavement performance data in the United States and Canada. This classification is important for its widely experienced and international orientation and integrity of classification, but no numerical threshold for distress severity classification is provided (Chaussees, 1998). The European background in distress identification and management is limited to some isolated cases. Based only on the surface condition evaluation, a guideline for pavement distress identification have been developed by the French Institute of Science and Technology for Transport (GEC, 2003) and the Swiss Association of Road and Transport Professionals (Tech Transport Institute, 2015), while only recently Ireland has included the assessment of road pavement (Instruction for Road Maintenance Planning, 1988). In Italy, the standard procedures for distress identification are few and limited to guidelines of the National Research Council and isolated in case of application of any road officers (EC Policy Orientation on Road Safety, 2010). The common aim of all the above-mentioned Distress Catalogues, is to provide a common set of criteria to evaluate the pavement condition and define management strategies. In fact, almost of the methods to perform a Pavement Management System (PMS) are based on the development of pavement condition indices to express the structural and operational performance, by combining different distress type expressed in terms of severity and extension, such as Pavement Condition Index (PCI) (Chambon et al., 2011; Mahler et al., 1991; Barnich and Van Droogenbroeck, 2009).

To perform a preventive maintenance approach for a whole road network, the availability of detailed information about actual road conditions is required. This can be obtained only through an accurate distress identification and classification. On the other side to

reach high level information, high investments in technologies and qualified staff are necessary. In recent years, inspired by the challenging goal of the European Union (EU) in Road Safety (Chan et al., 2010), many researchers and public agencies have used their energies and resources in investigating the importance of road condition in the incidental phenomena (Loprencipe and Pantuso, 2017; Shah et al., 2013; Tighe et al., 2000; Bennett et al., 2006). Moreover, due to the lack of economic resources, many countries find difficulties to investigate in sophisticated PMS and expensive extensive surveys. From these considerations, the importance of a pavement classification tool emerges. A defined set of information acquired by different surveys (Loprencipe and Pantuso, 2017; Loprencipe et al., 2017; Bennett et al., 2017) can provide a safety and comfort-oriented tool.

As a guideline for the distress classification, the most well-known and appreciated Distress Catalogues have been considered (Shahin, 1994; Coenen, 2017) to identify the most frequent distresses in flexible road pavement tightly coupled with the comfort and safety of the vehicles. The contribution of comfort or safety reduction has been considered for each type of distress, according to its severity and extension. According to the literature (Shahin, 1994; ASTM, 2018; Coenen, 2017), the distress types have been grouped into four families as described below:

- i. Cracking.
- ii. Visco-plastic deformation
- iii. Surface defects.
- iv. Miscellaneous distresses.

In order to facilitate their identification and to evaluate their contribution to comfort and safety level for road users.

### **2.2.1 Cracking group**

The pavement cracking group includes several kinds of distress, such as fatigue cracking, block cracking, edge cracking, longitudinal and transverse cracking, and reflection cracking. Most are related to climatic causes, load, and repetitive traffic load. Moreover, they have different effects on the global safety and comfort level of the road: for example, fatigue cracking advises the end of lifecycle for the pavement.

### **2.2.2 Visco-plastic deformation group**

All the deformations involving both superficial and bottom layers are considered as the visco-plastic deformation group. Corrugation, and lane/shoulder drop off generally correlate with horizontal and vertical displacement of the top layer of the pavement structure, while bumps and sags, rutting, depression, potholes, and swell may affect the whole structure of the pavement. The main reason of visco-plastic distresses is the load, followed by traffic and climatic effects.

### **2.2.3 Surface defect group**

The surface defect group encompasses bleeding, polish aggregate and raveling. When bleeding occurs, an excess of bituminous binder is present in the top layer surface, while in case of raveling an inadequate asphalt binder can cause the dislodging of aggregate. In the case of Polish aggregate, the top layer surface becomes smooth because of the aggregated exposition to the traffic, producing a reduction in adhesion. Causes of surface defects are related to bituminous materials characteristics and to the traffic effect.

### **2.2.4 Miscellaneous distress**

The Miscellaneous distress includes patching due to utility, cut patching, railroad crossing, catch basins and manholes cover. They are frequent in the urban area and are easily recognizable by their peculiar shape.

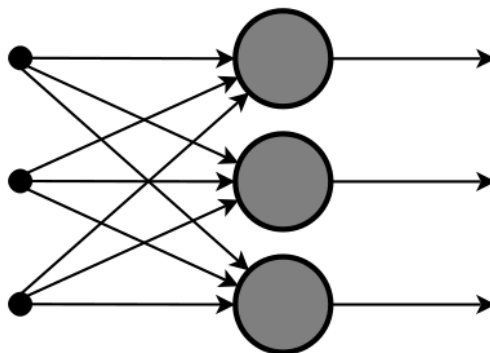
### **2.2.5 Automated devices in distress detection**

An integral solution by means of a single vehicle is sought for determining the pavement condition, location and severances of distresses which consists of different detection measuring methods and supporting equipment. More than a decade ago, this has been discussed by Bennett et al. (2006), but the technology has not developed that way. For data collection, synchronization and analysis additional equipment such as Global positioning system (GPS), frame grabber, data storage and Distance Measurement Instrument (DMI) have been used (Zhang et al., 2014).

## 2.3 Neural Networks

Artificial intelligence technique that mimics the operation of the human brain (nerves and neurons), and comprises of densely interconnected computer processors working simultaneously (in parallel). A key feature of neural networks is that they are programmed to 'learn' by sifting data repeatedly, looking for relationships to build mathematical models, and automatically correcting these models to refine them continuously. Also called neural net. Artificial neural networks are computational models which work similar to the functioning of a human nervous system. There are several kinds of artificial neural networks. These types of networks are implemented based on the mathematical operations and a set of parameters required to determine the output. Some of the types that are most used in today's machine learning (ML) applications are given below.

This neural network is one of the simplest forms of ANN, where the data or the input travels in one direction. The data passes through the input nodes and exit on the output nodes. This neural network may or may not have the hidden layers. In simple words, it has a front propagated wave and no back propagation by using a classifying activation function usually. Figure 2.1 shows a Single layer feed forward network.



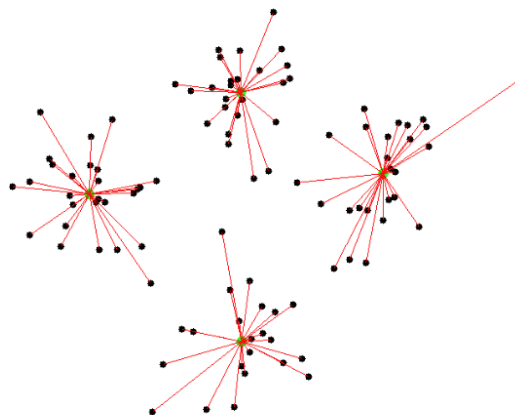
**Figure 2.1: Simple Diagram of Feedforward Neural Network**

Here, the sum of the products of inputs and weights are calculated and fed to the output. The output is considered if it is above a certain value i.e threshold (usually 0) and the neuron fires with an activated output (usually 1) and if it does not fire, the deactivated value is emitted (usually -1). Application of Feed forward neural networks are found in computer vision and speech recognition where classifying the target classes are



complicated. These kinds of Neural Networks are responsive to noisy data and easy to maintain.

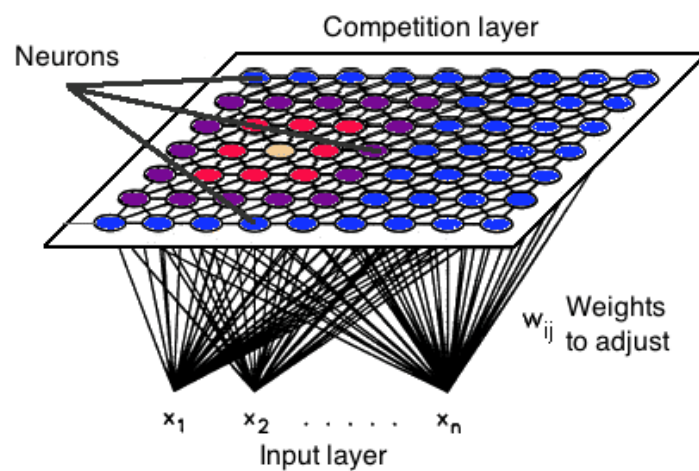
Radial basic functions consider the distance of a point with respect to the center. RBF functions have two layers, first where the features are combined with the Radial Basis Function in the inner layer and then the output of these features are taken into consideration while computing the same output in the next time-step which is basically a memory. Figure 2.2 is a diagram which represents the distance calculating from the center to a point in the plane similar to a radius of the circle. Here, the distance measure used in euclidean, other distance measures can also be used. The model depends on the maximum reach or the radius of the circle in classifying the points into different categories. If the point is in or around the radius, the likelihood of the new point begins classified into that class is high. There can be a transition while changing from one region to another and this can be controlled by the beta function. This neural network has been applied in Power Restoration Systems. Power systems have increased in size and complexity. Both factors increase the risk of major power outages. After a blackout, power needs to be restored as quickly and reliably as possible.



**Figure 2.2: Simple Diagram of Radial Basis Function Neural Network**

The objective of a Kohonen map is to input vectors of arbitrary dimension to discrete map comprised of neurons. The map needs to be trained to create its own organization of the training data. It comprises of either one or two dimensions. When training the map, the location of the neuron remains constant but the weights differ depending on the value.

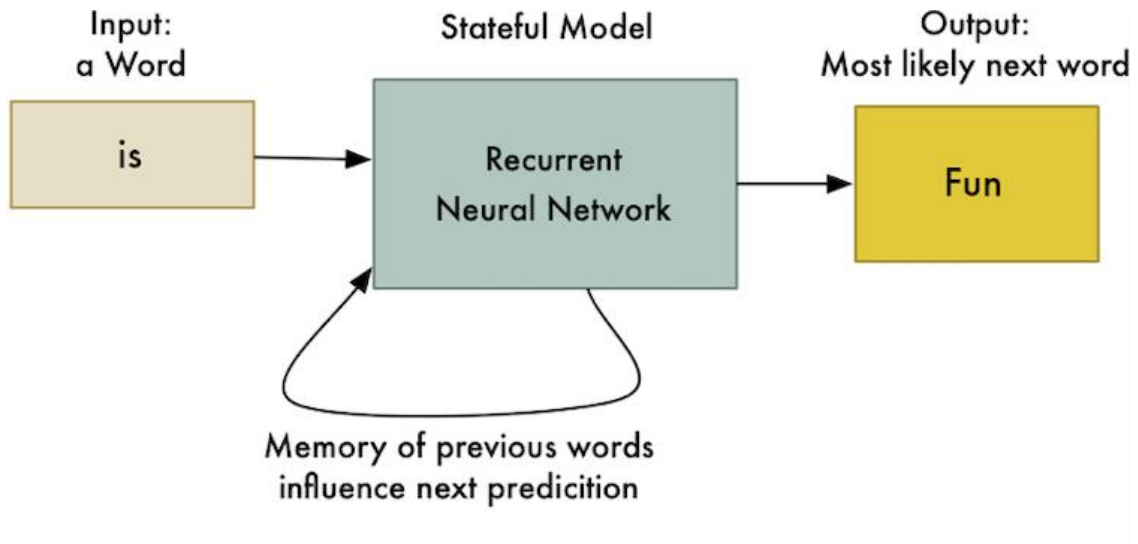
This self-organization process has different parts, in the first phase every neuron value is initialized with a small weight and the input vector. In the second phase, the neuron closest to the point is the ‘winning neuron’ and the neurons connected to the winning neuron will also move towards the point like in the graphic below (Figure 2.2). The distance between the point and the neurons is calculated by the euclidean distance, the neuron with the least distance wins. Through the iterations, all the points are clustered and each neuron represents each kind of cluster. This is the gist behind the organization of Kohonen Neural Network.



**Figure 2.3: Simple Diagram of Kohonen Self-Organizing Neural Network**

Kohonen Neural Network is used to recognize patterns in the data. Its application can be found in medical analysis to cluster data into different categories. Kohonen map was able to classify patients having glomerular or tubular with a high accuracy. The Recurrent Neural Network works on the principle of saving the output of a layer and feeding this back to the input to help in predicting the outcome of the layer. Here, the first layer (Figure 2.4) is formed similar to the feed forward neural network with the product of the sum of the weights and the features. The recurrent neural network process starts once this is computed; this means that from one time step to the next each neuron will remember some information it had in the previous time-step. This makes each neuron act like a memory cell in performing computations. In this process, it is needed to let the neural network to work on the front propagation and remember what information it needs for later use. Here, if the prediction is wrong the learning rate or error correction is used to

make small changes so that it will gradually work towards making the right prediction during the back propagation. The application of Recurrent Neural Networks can be found in text to speech (TTS) conversion models.



**Figure 2.4: Simple Diagram of Recurrent Neural Network**

Many types of artificial neural networks (ANNs), including the probabilistic neural network (NN) (Ahmadlou and Adeli, 2010), have been developed and adapted to research and industrial fields, but convolutional neural networks (CNNs) have been highlighted in image recognition, which are inspired by the visual cortex of animals (Ciresan et al., 2011). CNNs can effectively capture the grid-like topology of images, unlike the standard NNs, and they require fewer computations due to the sparsely connected neurons and the pooling process. Moreover, CNNs are capable of differentiating a large number of classes (Krizhevsky et al., 2012). These aspects make CNNs an efficient image recognition method (Simard et al., 2012; LeCun, 2015). The previous issue of CNNs was the need for a vast amount of labeled data, which came with a high-computational cost, but this issue was overcome through the use of well-annotated databases (ImageNet, no date; CIFAR-10 and CIFAR-100 data set, no date; MNIST Database, no date) and parallel computations using graphic processing units (Steinkrau et al., 2005). Owing to this excellent performance, a study for detecting railway defects using a CNN was later proposed (Soukup et al., 2014). Rail surfaces are homogenous, and the images are collected under controlled conditions. This cannot be considered the same as detecting concrete surface defects due to non-homogenous surface.

Therefore, a carefully configured deep architecture and abundant data set, taken under extensively varying conditions, is essential for dealing with the true variety of real-world problems. There has been a number of research in detecting cracks automatically on concrete surfaces using deep neural networks to reduce human conducted on-site inspections. Some are in basic levels others focusing on increasing accuracy. In 2017 Cha et al. proposed a vision-based method using a deep architecture of Convolutional Neural Networks (CNNs) for detecting concrete cracks without calculating the defect features. Which allowed the architecture to detect cracks in concrete surfaces in the extensively varying real-world situations (e.g., lighting and shadow changes). The CNN has three convolutional layers, 3 pooling layers, uses Rectified Linear Unit (ReLU) at the end of the hidden layers and uses max pooling. To classify input data into binary output the CNN architecture uses softmax function. The overall architecture ends up detecting cracks with 98% accuracy. Feng et al. (2017) proposed a deep Active Learning (AL) framework. Addressing the efficient training and deployment of an automatic defect detection system. A deep residual network (ResNet) is designed as the classifier for detection and classification of defects in an input image patch. In their paper, they prepared a labelled dataset, then fed it to an existing deep neural architecture, designed a suitable cost function for optimization and tuned hyper parameters for training to achieve a best performance. The model is able to detect cracks, deposit and water leakage on concrete surface with an accuracy of 87.5%.

Salman et al. (2013) proposed an approach to automatically distinguish cracks in digital images based on the Gabor filtering. Multi- directional crack detection can be achieved by high potential Gabor filter. The Gabor filter is a highly potential technique for multidirectional crack detection. The image analysis of the Gabor filter function was directly related to the manual visual perception. Once filtering was completed, the cracks aligned to different directions are detected. They have a detection precision of 95% for their proposed methodology.

Iyer et al. (2005) have designed a three-step method for the crack detection from the high contrast images. The proposed method detects the crack like pattern in the noisy environment using curvature evaluation and mathematical morphology technique. It was based on mathematical morphology and curvature evaluation that detects crack-like patterns in a noisy environment. In their study, segmentation is done defining the crack

like pattern with respect to a precise geometric model. Linear filtering was performed after cross curvature evaluation to distinguish them from analogous background pattern. They have identified the irregularity sequentially by the geometry-based recognition crack features. Nguyen et al. (2014) have proposed a method based on the edge detection of concrete cracks from noisy 2D images of concrete surfaces. They have observed the cracks as tree-like topology. Then based on the PSCEF non-crack objects were removed. After the separation, thresholding filter, and morphological thinning algorithm have been used to binarize the image for the crack center line estimation. Then the center line was fitted by cubic splines. They have linked the edge points to form the desired continuous crack edge. From the crack edge, the surface of the crack was attained.

Yang et al. (2015) have proposed an image analysis method to capture thin cracks and minimize the requirement for pen marking in reinforced concrete structural tests. They have used the studies like crack depth prediction, change in detection without image registration, crack pattern recognition based on artificial neural networks, applications to micro-cracks of rocks , and efficient sub-pixel width measurement. Stereo triangulation method was the adopted technique based on cylinder formula approximation and image rectification. Once they have the rectified output, the surface of the observed regions can be unfolded and presented in a plane image for following displacement and deformation analysis. From which the crack detection was analyzed.

Wang et al. (2010) have proposed a system for the image based crack detection and to characterize the crack based upon their effectiveness. They have categorized the present image-based crack detection into four categories. They are an integrated algorithm morphological approach, percolation approach and practical technique. A shading correction was done using integrated algorithm. The unclear crack prediction was detected using percolation method. The crack detection was done using morphological approach for the micro crack detection with the practical method providing high-performance feature extraction with more than 95% of accuracy.

## **2.4 Edge Detection**

Different methods are used to detect edges in image processing among these is Roberts Cross Algorithms. Robert processes a photograph into a line drawing, transform the line drawing into a three-dimensional representation and finally display the three-dimensional structure with all the hidden lines removed, from any point of view (Sparr, 2002). The

Roberts (1965) cross algorithm performs a 2-D spatial gradient convolution on the image. The main idea is to bring out the horizontal and vertical edges individually and then to put them together for the resulting edge detection. The two filters highlight areas of high spatial frequency, which tend to define the edge of an object in an image. The two filters are designed with the intention of bringing out the diagonal edges within the image. The  $G_x$  image will enunciate diagonals that run from the top-left to the bottom-right whereas the  $G_y$  image will bring out edges that run top right to bottom-left. The two individual images  $G_x$  and  $G_y$  are combined using the approximation equation  $G = |G_x| + |G_y|$ . The Canny edge detection operator was developed by John F. Canny in 1986 and uses a multistage algorithm to detect a wide range of edges in images. In addition, Canny edge detector is a complex optimal edge detector which takes significantly longer time in result computations. The image is firstly run through a Gaussian blur to get rid of the noise. When the algorithm is applied, the angle and magnitude is obtained which is used to determine portions of the edges to retain. There are two threshold cut-off points where any value in the image below the first threshold is dropped to zero and values above the second threshold is raised to one.

Canny (1986) considered the mathematical problem of deriving an optimal smoothing filter given the criteria of detection, localization and minimizing multiple responses to a single edge. He showed that the optimal filter given these assumptions is a sum of four exponential terms. He also showed that this filter can be well approximated by first-order derivatives of Gaussians. Canny also introduced the notion of non-maximum suppression, which means that given the pre-smoothing filters, edge points are defined as points where the gradient magnitude assumes a local maximum in the gradient direction. Another algorithm used is the Susan edge detector. This edge detection algorithm follows the usual method of taking an image and using a predetermined window centered on each pixel in the image applying a locally acting set of rules to give an edge response (Vincent et al., 2006). The response is then processed to give the output as a set of edges. The Susan edge filter has been implemented using circular masks (kernel) to give isotropic responses with approximations used either with constant weighting within it or with Gaussian weighting. The usual radius is 3.4 pixels, giving a mask of 37 pixels, and the smallest mask considered is the traditional 3'3 mask. The 37 pixels circular mask used in all feature detection experiments is placed at each point in the image and for each point

the brightness of each pixel within the mask is compared with that of nucleus. The comparison equation is,

$$C(\vec{r}, \vec{r}_0) = \begin{cases} 1, & \text{if } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0 & \text{if } |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases} \quad (2.1)$$

where  $\vec{r}$  is the position of the nucleus in the dimensional image,  $\vec{r}_0$  is the position of any other point within the mask,  $I(\vec{r})$  is the brightness of any pixel,  $t$  is the brightness in difference threshold and  $C$  is the output of the comparison. This comparison is done for each pixel within the mask where total  $n$  of the outputs ( $c$ ) is given as

$$n(\vec{r}_0) = \sum_{\vec{r}} C(\vec{r}, \vec{r}_0) \quad (2.2)$$

## 2.5 Object Detection

Image processing is important in modern data storage and data transmission especially in progressive transmission of images, video coding (teleconferencing), digital libraries, and image database, remote sensing. It has to do with manipulation of images done by algorithm to produce desired images (Milan et al., 2002). Digital Signal Processing (DSP) improve the quality of images taken under extremely unfavorable conditions in several ways: brightness and contrast adjustment, edge detection, noise reduction, focus adjustment, motion blur reduction etc. (Gonzalez and Woods, 2002). The advantage is that image processing allows much wider range of algorithms to be applied to the input data in order to avoid problems such as the build-up of noise and signal distortion during processing (Vincent et al., 2006). Many of the techniques of digital image processing were developed in the 1960's at the Jet Propulsion Laboratory, Massachusetts Institute of Technology (MIT), Bell laboratory and few other places. But the cost of processing was fairly high with the computing equipment of that era.

With the fast computers and signal processors available in the 2000's, digital image processing became the most common form of image processing and is general used because it is not only the most versatile method but also the cheapest. The process allows the use of much more complex algorithms for image processing and hence can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means (Michael, 2003). Thus, images are stored on the computers as collection of bits representing pixel or points forming the picture elements (Vincent et al., 2006). Firstly, images are a measure of parameter over space,

while most signals are measures of parameter over time. Secondly, they contain a great deal of information (Guthe and Strasser, 2004); image processing is any form of information processing for which the input is an image, such as frames of video; the output is not necessarily an image, but can be for instance be a set of features of the image (Yuval, 1996). Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. The process involves the enhancement or manipulation of an image which resulted in another image, removal of redundant data and the transformation of a 2-D pixel array into a statically uncorrelated data set (Priotr, 2004). Since images contain lots of redundant data, scholars have discovered that the most important information lies in it edges (Canny, 1986). Edges being the local property of a pixel and its immediate neighborhood, characterizes boundary (Chang-Huang, 2002). They correspond to object boundaries, changes in surface orientation and describe defeat by a small margin. Edges typically correspond to points in the image where the gray value changes significantly from one pixel to the next. Edges represents region in the image with strong intensity contrast; representing an image by its edges has the fundamental advantage that the amount of data is reduced significantly while retaining most of image's vital information with high frequencies (Keren et al., 2001). Thus, detecting Edges help in extracting useful information characteristics of the image where there are abrupt changes (Folorunso et al., 2007). Edge detection is a process of locating an edge of an image. Detection of edges in an image is a very important step towards understanding image features. Edges consist of meaningful features and contained significant information. It's reduced significantly the amount of the image size and filters out information that may be regarded as less relevant, preserving the important structural properties of an image. Most images contain some number of redundancies that can sometimes be removed when edges are detected and replaced, when it is reconstructed (Osuna et al., 1997). Eliminating the redundancy could be done through edge detection. When image edges are detected, every kind of redundancy present in the image is removed (Sparr, 2002). The purpose of detecting sharp changes in image brightness is to capture important events. Applying an edge detector to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. The image quality reflects significant information in the output edge and the size of the image is reduced. This in turn explains further that edge detection is one of the ways of solving the problem of high volume of space images



occupy in the computer memory. The problems of storage, transmission over the Internet and bandwidth could be solved when edges are detected. Since edges often occur at image locations representing object boundaries, edge detection is extensively used in image segmentation when images are divided into areas corresponding to different objects.

## **2.6 Feature Extraction**

Labeling of connected components in a binary image is one of the most fundamental operations in pattern analysis (recognition), computer (robot) vision, and machine intelligence (Ronsen and Denjiver, 1984; Gonzalez and Woods, 1992; Rosenfeld and Kak, 1982). By use of the labeling operation, a binary image is transformed into a symbolic image in which all pixels belonging to a connected component are assigned a unique label. Labeling is required whenever a computer or a system needs to recognize objects (connected components) in binary images; in other words, labeling is required in almost all image-based applications such as fingerprint identification, character recognition, automated inspection, target recognition, face identification, medical image analysis, and computer-aided diagnosis (Suzuki et al., 2003; Suzuki et al., 2008; Suzuki et al., 2005). Especially in real-time applications such as traffic-jam detection, automated surveillance, and target tracking, faster labeling algorithms are always demanded. Many algorithms have been proposed for addressing this issue, because the improvement of the efficiency of labeling is critical in many applications. For ordinary computer architectures such as the Von Neumann architecture and two-dimensional images, there are mainly the following four classes of algorithms:

(1) Multi-scan algorithms. Algorithms (Haralick, 1981; Hashizume et al., 1990) scan an image in the forward and backward raster directions alternately to propagate label equivalences until no label changes.

(2) Two-scan algorithms. Algorithms (Rosenfeld and Pfalts, 1966; Rosenfeld, 1970; Lumia et al., 1983; Lumia, 1983; Shirai, 1987; Gotoh et al., 1987; Komeichi et al., 1988; Haralick and Shapiro, 1992; Naoi, 1995; Otoo and Suzuki, 2008; He et al., 2008) complete labeling in two scans: during the first scan, they assign provisional labels to object pixels, and record label equivalences. Label equivalences are resolved during or

after the first scan. During the second scan, all equivalent labels are replaced by their representative label.

(3) Hybrid algorithms. The algorithm (Suzuki et al., 2003) is a hybrid between multi-scan algorithms and two-scan algorithms. Like multi-scan algorithms, the hybrid algorithm scans an image in the forward and backward raster directions alternately. During the scans, as in two-scan algorithms, a one-dimensional table is used for recording and resolving label equivalences. Experimental results demonstrated that four is the upper limit on the number of scans (Suzuki et al., 2003; Hu et al., 2005).

(4) Tracing-type algorithms. These algorithms (Rosenfeld; 1970; Hu et al., 2005; Ballard, 1982; Chang et al., 2004) avoid analysis of label equivalences by tracing the contours of objects (connected components) or by use of an iterative recursion operation. Such algorithms had been considered to be efficient only for simple images, but not for complicated images, until Chang's contour-tracing algorithm (Chang, 2004) was proposed. There are other labeling algorithms for special image representations and special computer architectures. Algorithms (Samet, 1981; Gargantini, 1982; Tamminen et al., 1984; Samet, 1985; Samet, 1986; Samet, 1988; Hecquard and Acharya, 1991; Dillencourt et al., 1992) were developed for the images represented by hierarchical tree structures (Srihari, 1980; Jackins and Tanimoto, 1980; Samet, 1984), i.e., n-array trees such as 'bintree', 'quadtree', 'octree', etc. The efficiency of such algorithms may be better than that of other conventional ones, but in the worst case, it is the same as that of conventional two-scan algorithms. On the other hand, parallel algorithms (Hirschberg et al., 1979; Nassimi and Sahani, 1980; Schiloach and Vishkin, 1982; Tucker, 1986; Manohar and Ramapriyan, 1989; Alnuweiri and Prasanna, 1992; Olariu et al., 1993; Choudhary and Thakur; 1996; Bhattacharya, 1996) were developed for parallel machine models such as a mesh connected massively parallel processor or systolic array processors. This paper presents a new two-scan algorithm for labeling of connected components in binary images. We propose an efficient strategy for assigning provisional labels to object pixels and checking label equivalence by case analysis, and we use the equivalent-label sets and a representative label table for solving label equivalences (He et al., 2008; He et al., 2007). Experimental results showed that our algorithm is efficient for connected-component labeling. The rest of this paper is organized as follows: we review the method for assigning provisional labels to object pixels and checking label

equivalence in conventional raster-scan labeling algorithms in the next section, and we introduce our algorithm in Section 3. In Section 4, we show experimental results, and in Section 5, we present a discussion. We give our concluding remarks in Section 6. The preliminary version of this paper was presented at the 2007 IEEE International Conference on Image Processing (ICIP 2007) (He et al., 2007). However, the efficient strategy for assigning provisional labels for object pixels by case analysis given in Section 3, the correctness of our algorithm given in Section 5.1, the complexity analysis, and the efficiency analysis of the proposed algorithm given in Sections 5.2 and 5.3, respectively, the comparison with other conventional labeling algorithm given in Sections 5.4, 5.5, 5.6 and 5.7, and the method for generating consecutive labels for connected components given in Section 5.8 are new.

For an  $N \times N$  binary image, we use  $b(x, y)$  to denote the pixel value at  $(x, y)$  in the image, where  $1 \leq x, y \leq N$ , and  $VO$  for the value of object pixels and  $VB$  for that of background pixels. We assume that  $VO$  and  $VB$  are larger than the value of any provisional label, and  $VO < VB$ .

For convenience, we consider only the case for eight-connected connectivity in this paper, because our algorithm can easily be extended to that for labeling with four-connected connectivity.

By using the mask (Rosenfeld, 1982) shown in Fig. 2.5, all conventional pixel-based raster-scan algorithms except the one proposed in Ref. (He et al., 2007) scan a given image in the raster scan direction once (the first scan) to process pixels one by one. If the current pixel  $b(x, y)$  is a background pixel, nothing needs to be done. On the other hand, if there is no object pixel in the mask other than the current pixel, it is assigned a new label. Otherwise, it is assigned the minimal label in the mask, and all different labels (if any) in the mask are recorded to be equivalent labels.

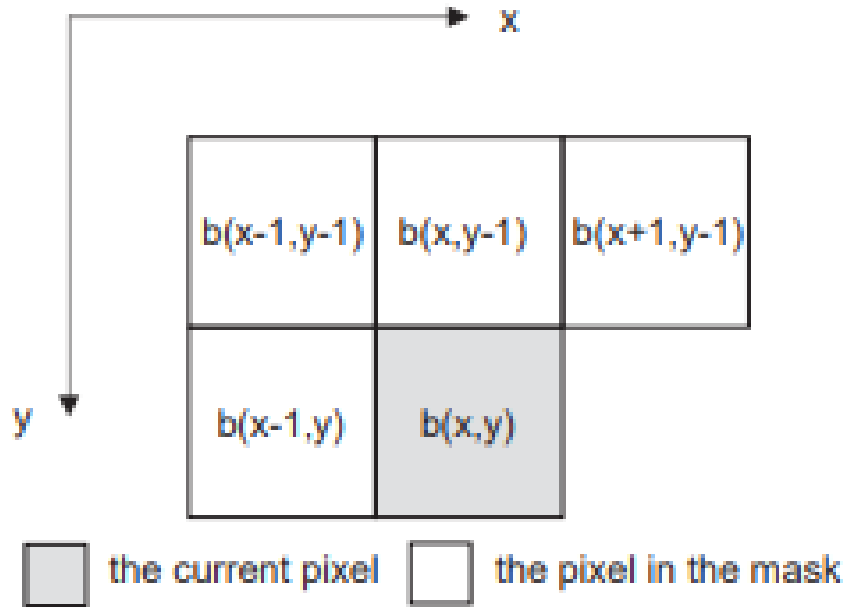
By the above first scan, the connected component illustrated in Fig. 2.6(a), for example, is provisionally labeled as the one shown in Fig. 2.6(b), where label 1 and label 4, label 2 and label 5, label 2 and label 3, and label 3 and label 4 are recorded to be equivalent labels.

There are several methods for recording and resolving label equivalences. One uses an  $L \times L$  two-dimensional array table (Shirai, 1987; Otoo and Suzuki, 2009) to record label equivalences, where  $L$  is the number of provisional labels assigned to an image.<sup>2</sup> Each element  $X_{ij}$  of the table is initialized to 0. If provisional labels  $u$  and  $v$  are found to be equivalent labels, then the element  $X_{uv}$  is set to 1. After the first scan, for each provisional label  $m$  from 0 to  $L$ , it analyzes the table to find all  $m$ 's equivalent labels, and  $m$  is used as their representative label.

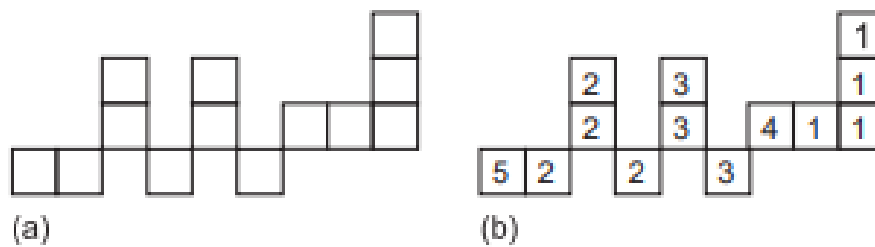
Another method is an application of the so-called union-find algorithm (Wu et al., 2004; Najafabadi et al., 2015; LeCun et al., 2015; Albelwi and Mahmood, 2017; Gopalakrishnan et al., 2017). It records label equivalences by use of union find trees. When two provisional labels are found to be equivalent, the two union-find trees corresponding to the two labels are connected together. Thus, after the first scan, all equivalent provisional labels will be combined in the same union-find tree, and the label on the root is used as their representative label.

The third method was proposed in Refs. (He, 2008; He, 2007). By this method, all equivalent labels are combined in a set (called an equivalent label set), and the smallest label among them is used as their representative label, whose relation is recorded in the so-called representative label table. When two provisional labels are found to be equivalent, the two equivalent label set corresponding to the two labels are combined together, and the representative label table is updated at the same time. Thus, after the first scan, all equivalent provisional labels will be combined in the equivalent label set and hold the same representative label.

After label equivalences are resolved, the second scan is executed by replacement of all equivalent labels with their representative label.



**Figure 2.5: Mask for Eight-Connected Connectivity**



**Figure 2.6: Illustration of Provisional Labeling by the First Scan. (a) Connected-Component Example. (b) Provisional Labels Assigned after the First Scan**

### 2.7 Summary

This chapter gives an overview of the recent studies regarding crack identification, neural networks, edge detection, object detection and feature extraction methodology. In literature, there are several distress identification catalogues by different researchers and organizations, which almost adopt the same identification and evaluation criteria. One of the most well-known and diffused references is the ASTM D6433-16 (ASTM, 2018), which provides distress criteria identification and classification for both flexible and rigid pavement. Different methods are used to detect edges in image processing among these is Roberts Cross Algorithms. Robert processes a photograph into a line drawing, transform the line drawing into a three-dimensional representation and finally display the three-dimensional structure with all the hidden lines removed, from any point of view.

Among many types of artificial neural networks (ANNs), convolutional neural networks (CNNs) have been highlighted in image recognition, which are inspired by the visual cortex of animals. CNNs can effectively capture the grid-like topology of images that can easily differentiate a large number of classes, unlike the standard NNs, and they require fewer computations due to the sparsely connected neurons and the pooling process. Moreover, CNNs are capable of differentiating a large number of classes.

## CHAPTER 3

### METHODOLOGY

---

#### 3.1 Introduction

Current pavement condition inspections are conducted by visual inspections, heavily related on the human factors, which makes them expensive and not affordable for continuous network-scale pavement inspections. Therefore, in the last few years, several research activities have been conducted to design and develop new algorithms and methods for detecting, localizing and classifying road damages in an automatic or semi-automatic way. Amongst the others, several effective applications for automatic pavement distress detection have been conducted, exploiting Image Processing techniques (LeCun et al., 2015; Steinkrau et al., 2005; Soukup and Huber-Mörk, 2014) and Machine Learning (ML) algorithms (Cha et al., 2017; Feng et al., 2017). The techniques mentioned first are applied for image segmentation procedures, which are usually implemented to identify the selected items of interest within a chosen dataset of images (Salman et al., 2013). The second ones, are exploited to classify the selected items to one of several possible classes (Iyer and Sinha; 2005) . In the context of pavement damage detection, the image segmentation process is often made more difficult by critical factors, such as lighting and shadows effects, frequently shown in road images. Furthermore, the different accuracy and level of detail required to identify different features of pavement distresses, still adds complexity to this operation (Nguyen et al., 2014). For this purpose, in order to overcome these occurrences, several techniques based on thresholding (Yang et al., 2015) and edge detection (Wang and Huang, 2010) have been proposed in the literature.

Concerning the classification process, several ML techniques have been proposed in the literature, based on different types of features, such as Scale-Invariant Feature Transform (SIFT) (Milan, 2002), Histogram of Oriented Gradients (HOG) (Gonzalez, 2002) and classification algorithms, such as Support Vector Machine (SVM) (Baker and Nayar, 1996) and Bayesian Classifier (Michael, 2003). More recently, the application of Deep Learning techniques (Vincent et al., 2009) has provided significant development in the field of Artificial Intelligence (Guthe and Strasser, 2004) and, more specifically, in Image Recognition (Yuval, 1996). In this context, several studies have demonstrated that Deep Learning algorithms can provide significant advantages in the recognition and

classification of pavement distresses (Priotr, 2004). In (Canny, 1986), for instance, the authors present a system in which the transfer learning technique (Chang-Huang, 2002) is applied to a deep Convolutional Neural Network (CNN) to identify road damages within the ImageNet dataset. In the literature (Keren et al., 2001), the authors apply the same technique to a Region Based Convolutional Neural Network (R-CNN) for the same issue, but in more complex scenarios, whilst in (Folorunso et al., 2007) an architecture based on multiple Faster R-CNNs is optimized to achieve the best performance in pavement distress detection. In (Osuna et al., 1997), the authors propose a modified version of the YOLO (You Look Only Once) network specifically designed to identify and classify a consistent number of pavement distresses. In (Sparr, 2002), Majidifard et al. propose a hybrid model based on a YOLO v2 network to classify road damages and a U-net network to diagnose its severity. The experimental results are obtained on a labelled pavement dataset (Roberts, 1965) created from images extracted from Google Street View and annotated using nine different classes of distress.

### 3.2 Sobel Filter Design

Most edge detection methods work on the assumption that the edge occurs due to discontinuity in the intensity function or a very steep intensity gradient in the image. Using this assumption, if one takes the derivative of the intensity value across the image and find points, where the derivative is maximum, then the edge could be located. The gradient is a vector, whose components measure how rapid pixel value are changing with distance in the x and y direction. Thus, the components of the gradient may be found using the following approximation:

$$\frac{\delta f(x, y)}{\delta x} = \Delta x = \frac{f(x + dx, y) - f(x, y)}{dx} \quad (3.1)$$

$$\frac{\delta f(x, y)}{\delta y} = \Delta y = \frac{f(x, y + dy) - f(x, y)}{dy} \quad (3.2)$$

where  $dx$  and  $dy$  measure distance along the x and y directions respectively. In discrete images, one can consider  $dx$  and  $dy$  in terms of numbers of pixel between two points.  $dx = dy = 1$  (pixel spacing) is the point at which pixel coordinates are  $(i, j)$  thus,

$$\Delta x = f(i + 1, j) - f(i, j) \quad (3.3)$$

$$\Delta y = f(i, j + 1) - f(i, j) \quad (3.4)$$

To detect the presence of a gradient discontinuity, one could calculate the change in the



gradient at (i, j). This can be done by finding the following magnitude measure

$$M = \sqrt{(\Delta x^2 + \Delta y^2)} \quad (3.5)$$

and the gradient direction  $\theta$  is given by

$$\theta = \tan^{-1} \frac{\Delta x}{\Delta y} \quad (3.6)$$

### 3.2.1 Method of the filter design

There are many methods of detecting edges; most different methods may be grouped into these two categories:

- i. **Gradient:** The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. For example, Roberts, Prewitt, Sobel where detected features have very sharp edges.
- ii. **Laplacian:** The Laplacian method searches for zero crossings in the second derivative of the image to find edges e.g., Marr-Hildreth, Laplacian of Gaussian etc. An edge has one dimensional shape of a ramp and calculating the derivative of the image can highlight its location. Edges may be viewpoint dependent: these are edges that may change as the viewpoint changes and typically reflect the geometry of the scene which in turn reflects the properties of the viewed objects such as surface markings and surface shape. A typical edge might be the border between a block of red color and a block of yellow, in contrast. However, what happens when one looks at the pixels of that image is that all visible portion of one edge is compacted. The Sobel operator is an example of the gradient method. The Sobel operator is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function (Sobel and Feldman, 1968). The different operators in eq. (3.7) and (3.6) correspond to convolving the image with the following mark.

$$\Delta x = \begin{vmatrix} -1 & 1 \\ 0 & 0 \end{vmatrix} \quad (3.7)$$

$$\Delta y = \begin{vmatrix} -1 & 0 \\ 1 & 0 \end{vmatrix} \quad (3.8)$$

When this is done, then:

- i. The top left-hand corner of the appropriate mask is super-imposed over each pixel of the image in turn,
- ii. A value is calculated for  $\Delta x$  or  $\Delta y$  by using the mask coefficients in a weighted sum of the value of pixels  $(i, j)$  and its neighbors,
- iii. These masks are referred to as convolution masks or sometimes convolution kernels. Instead of finding approximate gradient components along the  $x$  and  $y$  directions, approximation of the gradient components could be done along directions at  $45^\circ$  and  $135^\circ$  to the axes respectively. In this case

$$\Delta x = f(i + 1, j + 1) - f(i, j) \quad (3.9)$$

$$\Delta y = f(i, j + 1) - f(i + 1, j) \quad (3.10)$$

This form of operator is known as the Roberts edge operator and was one of the first set of operators used to detect edges in images (Robert, 1965). The corresponding convolution masks are given by:

$$\Delta_1 = \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix} \quad (3.11)$$

$$\Delta_2 = \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \quad (3.12)$$

An advantage of using a larger mask size is that the errors due to the effects of noise are reduced by local averaging within the neighborhood of the mask. An advantage of using a mask of odd size is that the operators are centered and can therefore provide an estimate that is based on a center pixel  $(i, j)$ . One important edge operator of this type is the Sobel edge operator. The Sobel edge operator masks are given as :

$$\Delta x = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix}, \quad (3.13)$$

$$\Delta y = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \quad (3.14)$$

The operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The result therefore shows how "abruptly" or "smoothly" the image changes at that point and therefore how likely it is that part of the image represents an edge, as well as how that the edge is likely to be oriented. In practice, the magnitude (likelihood of an edge) calculation is more reliable and easier to interpret than the direction calculation. Mathematically, the gradient of a two-variable function (the image intensity function) at

each image point is a 2D vector with the components given by the derivatives in the horizontal and vertical directions. At each image point, the gradient vector points to the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. This implies that the result of the Sobel operator at any image point which is in a region of constant image intensity is a zero vector and at a point on an edge is a vector which points across the edge, from darker to brighter values. The algorithm for developing the Sobel model for edge detection is given below.

### 3.2.2 Pseudo-codes for sobel edge detection method

*Input:* A Sample Image

*Output:* Detected Edges

**Step 1:** Accept the input image

**Step 2:** Apply mask  $G_x$ ,  $G_y$  to the input image The  $G_x$  image will enunciate diagonals that run from the top-left to the bottom-right where as the  $G_y$  image will bring out edges that run top-right to bottom-left. The two individual images  $G_x$  and  $G_y$  are combined using the approximation equation  $G = G_x + G_y$

**Step 3:** Apply Sobel edge detection algorithm and the gradient

**Step 4:** Masks manipulation of  $G_x$ ,  $G_y$  separately on the input image

**Step 5:** Results combined to find the absolute magnitude of the gradient

$$|G| = \sqrt{(G_x^2 + G_y^2)} \quad (3.15)$$

**Step 6:** the absolute magnitude is the output edges

### 3.3 Classification Paradigm: Convolutional Neural Network

Deep learning is a machine learning method based on neural networks that applied multiple layers of processing information enabling computers to automatically extract features from data with multiple levels of abstraction; in each transition of layers, representation at one-layer transforms into representation at a higher abstract level in the next layer (Najafabadi et al., 2015). The distinctive property of deep learning is that layers of the feature are learned automatically from data (Le Cun, 2015).

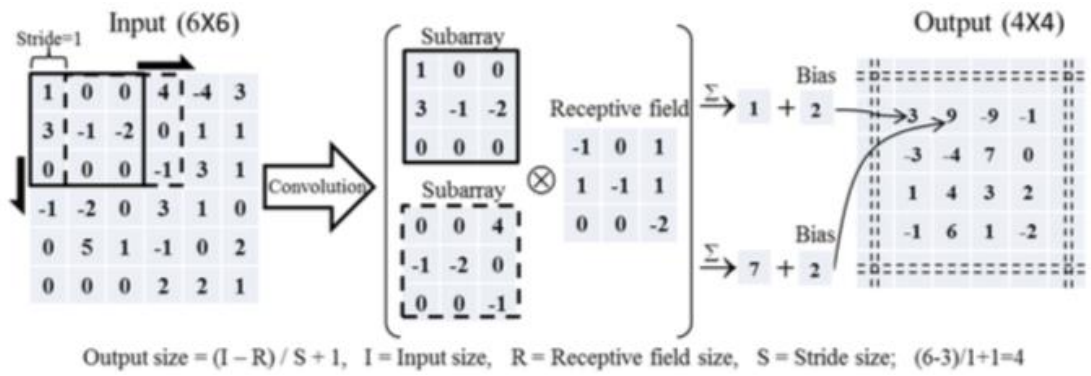
Deep learning can be used for supervised, unsupervised, and semi-supervised learning and has a widespread application in computer vision, analysis of big data, speech

recognition, object detection, handwriting recognition, image classification, audio processing, robotics, self-driving car, drug discovery.

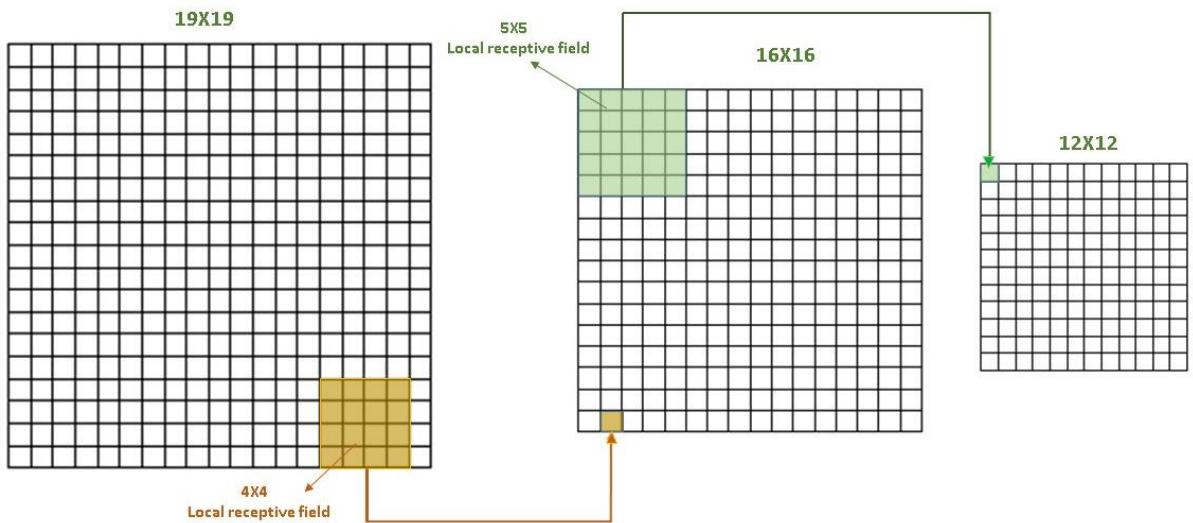
CNN is a special form of deep neural networks that was designed to process data that have multiple arrays and grid-like topology (Albelwi and Mahmood, 2017). CNNs can be used on 3D (video), 2D (image), and 1D (text or audio) input data to perform one of the mentioned deep learning applications (Baker and Nayar, 1996). In recent years, a large number of researches were conducted in pavement management systems to automatic feature extraction and distress detection by applying CNN on the pavement image as input data (Gopalakrishnan et al., 2017).

### **3.3.1 Convolutional neural network**

The structure of CNNs consisted of three main substructures, which include: convolutional layers, pooling layers, fully connected layers. A convolution layer performs the following three operations throughout an input array as shown in Figure 2.13. First, it performs element-by-element multiplications (i.e., dot product) between a subarray of an input array and a receptive field. The receptive field is also often called the filter, or kernel. The initial weight values of a receptive field are typically randomly generated. Those of bias can be set in many ways in accordance with networks' configurations (Krizhevsky et al., 2012). Both values are tuned in training using a stochastic gradient descent (SGD) algorithm. The size of a subarray is always equal to a receptive field, but a receptive field is always smaller than the input array. Second, the multiplied values are summed, and bias is added to the summed values. Figure 3.1 (a) shows the convolutions of the subarrays (solid and dashed windows) with an input array and a receptive field. One of the advantages of the convolution is that it reduces input data size, which reduces computational cost. An additional hyper parameter of the layer is the stride. The stride defines how many of the receptive field's columns and rows (pixels) slide at a time across the input array's width and height. A larger stride size leads to fewer receptive field applications and a smaller output size, which also reduces computational cost, though it may also lose features of the input data. The output size of a convolution layer is calculated by the equation shown in Figure 3.1 (a).



(a)



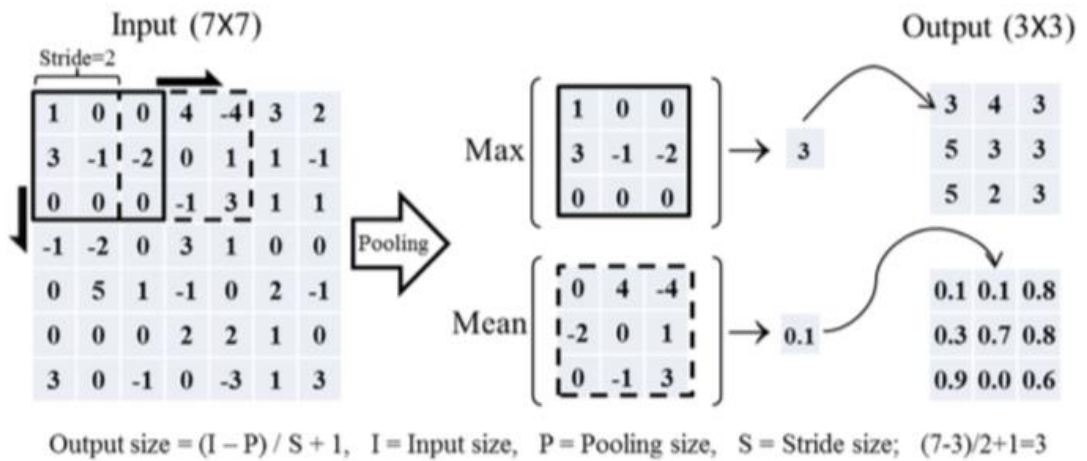
(b)

**Figure 3.1: Convolution Process to Create Feature Map**

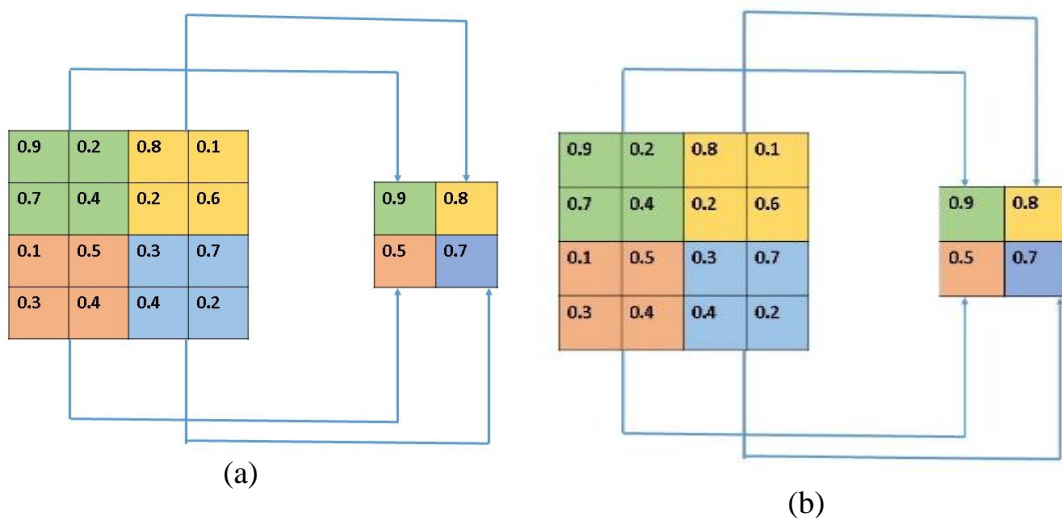
Convolutional layers are also made from several feature maps, and each unit of feature maps is made from convolving a small region in input data which is called the local receptive field. As can be seen in Fig. 3.1 (b), a new feature map is created by sliding a local receptive field over the input. The convolution can be used in various kinds of data such as image, text. For example, in the image, an area of pixels is convolved, and in the text, a group of characters or words are convolved. Unlike the standard neural network, each neuron in the layers is not connected to all the nodes (neurons) in the previous layer but is just connected to nodes in a special region known as the local receptive field (LeCun, 2015).

Another key aspect of the CNNs is a pooling layer, which reduces the spatial size of an input array. This process is often defined as down sampling. There are two different

pooling options. Max pooling takes the max values from an input array's subarrays, whereas mean pooling takes the mean values. Figure 3.2 shows the pooling method with a stride of two, where the pooling layer output size is calculated by the equation in the figure. Owing to the stride size being larger than the convolution example in Figure 3.2, the output size is further reduced to  $3 \times 3$ . Max pooling performance in image data sets is better than that of mean pooling.



**Figure 3.2: Illustration of CNN Architecture**

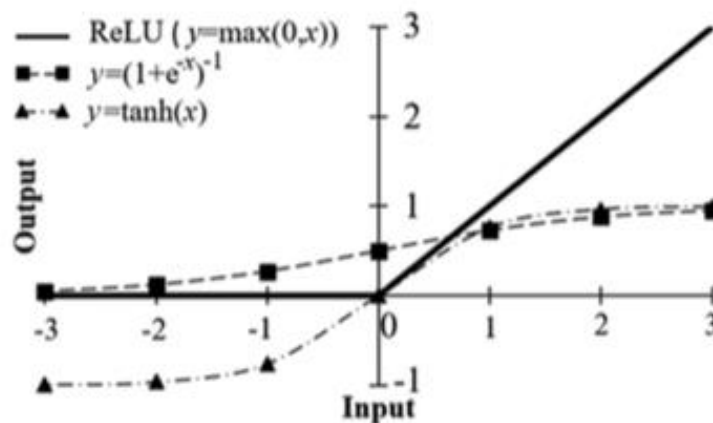


**Figure 3.3: Pooling Layer**

Pooling layers are commonly used immediately after convolutional layers. These layers were generated to simplify the information and reduce the scale of feature maps. In other words, pooling layers make a condensed feature map from each feature map in convolutional layers. In some references, these layers are called the subsampling layer. Pooling operation can be performed in various types such as geometric average, harmonic average, maximum pooling (Ioffe and Szegedy, 2015). Max-pooling and average-pooling

are two of the most prevalent processes for pooling that have been presented in Fig. 3.2. The pooling layers are necessary to reduce the computational time and overfitting issues in the CNN (LeCun et al., 2015).

The most typical way to give nonlinearity in the standard ANN is using sigmoidal functions, such as  $y = \tanh(x)$ , but saturating nonlinearities slows computations (Nair and Hinton, 2010). Recently, the ReLU was introduced as a nonlinear activation function. Figure 3.3 depicts the several examples of nonlinear functions. Briefly, while other nonlinear functions are bounded to output values (e.g., positive and negative ones and zeros), the ReLU has no bounded outputs except for its negative input values. Intuitively, the gradients of the ReLU are always zeros and ones. These features facilitate much faster computations than those using sigmoidal functions and achieve better accuracies.



**Figure 3.4: Non-Linear Activation Functions**

Overfitting has been a long-standing issue in the field of machine learning. This is a phenomenon where a network classifies a training data set effectively but fails to provide satisfactory validation and testing results. To address this issue, dropout layers are used (Srivastava et al., 2014). Training a network with a large amount of neurons often results in overfitting due to complex coadaptation. The main idea of dropout is to randomly disconnect the connections between neurons of connected layers with a certain dropout rate. Accordingly, a network can generalize training examples much more efficiently by reducing this coadaptation. A well-known trick, taking the average values of a training data set (i.e., whitening), has often been used to shorten network training time (Ioffe and Szegedy, 2015). However, the distribution of layer's input shifts by

passing through layers, which is defined as internal covariate shift, and this has been pointed out as being the major culprit of slow training speed. A Batch Normalization was proposed to adapt the similar effect of whitening on layers (Bengio et al., 2016). As a result, this technique facilitates high-learning rate and leads to much faster network convergence.

To classify input data, it is necessary to have a layer for predicting classes, which is usually located at the last layer of the CNN architecture. The most prominent method to date is using the *softmax* function given by the equation given below, which is expressed as the probabilistic expression  $p(y(i) = n | x(i); W)$  for the  $i$ th training example out of  $m$  number of training examples, the  $j$ th class out of  $n$  number of classes, and weights  $W$ , where  $Wn, T, x(i)$  are inputs of the *softmax* layer. The sum of the right-hand side for the  $i$ th input always returns as 1, as the function always normalizes the distribution. In other words, this equation returns probabilities of each input's individual classes.

$$P(y^{(i)} = n | x^{(i)}; W) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; W) \\ p(y^{(i)} = 2 | x^{(i)}; W) \\ \vdots \\ p(y^{(i)} = n | x^{(i)}; W) \end{bmatrix} = \frac{1}{\sum_{i=1}^n e^{W_j^T x^{(i)}}} \begin{bmatrix} e^{W_1^T x^{(i)}} \\ e^{W_2^T x^{(i)}} \\ \vdots \\ e^{W_n^T x^{(i)}} \end{bmatrix} \quad (3.16)$$

As the initial values of  $W$  are randomly assigned during training, the predicted and actual classes do not usually coincide. To calculate the amount of deviations between the predicted and actual classes, the *softmax* loss function is defined by an equation.

$$L = \frac{1}{m} \left[ \sum_{i=1}^m \sum_{j=1}^n 1\{y^{(i)} = j\} \log \frac{e^{W_j^T x^{(i)}}}{\sum_{i=1}^n e^{W_j^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{j=1}^n e^{W_j^2} \quad (3.17)$$

The new index  $L$  is introduced to indicate that  $\sum n l = \exp(W T x(i))$  is independent of  $\sum n j = 11\{. \}$ . The term  $1\{y(i) = j\}$  is the logical expression that always returns either zero or ones. In other words, if a predicted class of the  $i$ th input is true for  $j$  class, the term returns ones, returning zeros otherwise. The last hyper-parameter  $\lambda$  in the equation is a regularization (i.e., weight decay) parameter to penalize large weights, which is also a well-known trick for preventing overfitting (Bottou, 2012; Aszemi and Dominic, 2019). To narrow the deviations, an algorithm that updates receptive field weights is necessary for obtaining the expected results (i.e., predicting true classes). This process is considered



for CNN training. There are several known methods, but SGD using back-propagation is considered the most efficient and simplest way to minimize the deviations (Ioffe and Szegedy, 2015). The standard gradient descent algorithm performs updating  $W$  on an entire training data set, but the SGD algorithm performs it on single or several training samples. To accelerate the training speed, the momentum algorithm (Bottou, 2012) is also often used in SGD. The overall updating process is as follows. First, the gradient  $\nabla W$  of a loss function is calculated with respect to  $W$ .

Second, the hyper-parameters of momentum  $\varepsilon$  and learning rate  $\alpha$  are introduced to update ( $\leftarrow$ ) velocity  $v$ , where momentum is defined as mass times velocity in physics, but with unit mass being what is considered in SGD. Finally, the weights are updated. A network can be tuned by repeating the explained process several times until  $W_j \leftarrow W_j + v$  converges. The superscript  $(i)$  indicates the  $i$ th training sample, where the range of  $i$  is dependent on a mini-batch size, which defines how many training samples out of the whole data set are used. For example, if 100 images are given as the training data set and 10 images are assigned as the mini-batch size, this network updates weights 10 times; each complete update out of the whole data is called an epoch.

$$\Delta_w L(W; x^{(i)}, y^{(i)}) = \frac{1}{m} \sum_{i=1}^m [x^{(i)} \{1(y^{(i)} = j) - p(y^{(i)} = j | x^{(i)}; W)\}] + \lambda W_j \quad (3.18)$$

Fully connected layers are the final layers in the CNN structure that can be one or more layers and placed after a sequence of convolution and pooling layers. This part of CNN comprises the composite and aggregates of the most important information from all procedures of CNN. Consequently, these layers provide the feature vector for the input data, which can be used for some machine learning tasks such as classification, prediction (Albelwi and Mahmood, 2017). The last layer of fully connected layers is known as softmax classifier and determines the probability of each class label over  $N$  classes (LeCun and Bengio, 2015).

Designing the CNN structure is a big challenge because there are many hyperparameters that have significant influence on the efficiency of CNNs such as depth (which includes the number of convolutional, pooling, and fully-connected layers), the number of filters, stride (step-size that the local receptive field must be moved), pooling types, locations, and sizes, and the number of units in fully-connected layers. Finding the proper

hyperparameters combination needs rigorous analysis. Computational intelligence for modeling of asphalt pavement surface distress (Manohar and Ramapriyan, 1989) knowledge and is often performed as a trial-and-error process. Recently, this challenge has been raised as an optimization problem.

Generally, there are two methods for applying CNN models that include: training from scratch and performing transfer learning by use of pre-trained models. If the first method (training from scratch) was applied for training a CNN model, it would be necessary to define the number of layers and filters and use massive amounts of data which are a time-consuming procedure.

In the other method (transfer learning), one of the pre-trained CNN models is being used that was trained on the source domain (big image data set). This means that in transfer learning, the ability of pre-trained models to learn the predictive function helps to train the new target domain (new image dataset) instead of training from scratch. Transfer learning is a much faster and easier method for applying deep learning, and in this method, it is not necessary to understand the structure and combinations of network layers. As can be seen, it is possible to create a pavement distress detector and classifier model by using transfer learning techniques, and a proper dataset of pavement distresses image.

### **3.4 Summary**

CNN is a special form of deep neural networks that was designed to process data that have multiple arrays and grid-like topology. CNNs can be used on 3D (video), 2D (image), and 1D (text or audio) input data to perform one of the mentioned deep learning applications. The structure of CNNs consisted of three main substructures, which include: convolutional layers, pooling layers, fully connected layers. Designing the CNN structure is a big challenge because there are many hyperparameters that have significant influence on the efficiency of CNNs such as depth (which includes the number of convolutional, pooling, and fully-connected layers), the number of filters, stride (step-size that the local receptive field must be moved), pooling types, locations, and sizes, and the number of units in fully-connected layers.

## **CHAPTER 4**

### **DATA COLLECTION AND ANALYSIS**

---

#### **4.1 Introduction**

Data collection and data analysis are considered the major subjects for any ideal pavement management system. The majority of the necessary information for pavement administration is provided by effective pavement surveys. For any vital pavement project, the following are considered the most important requirements: a quantified condition of networks, more precise and attainable information, forecasting the maintenance and rehabilitation requirements, setting the rehabilitation and maintenance priorities, tracking interpretation treatments, prediction of the pavement evaluation, and assigning funding. Therefore, it is important to acquire precise pavement condition data, in an effective and secure way, to ensure a credible analysis and interpretation system.

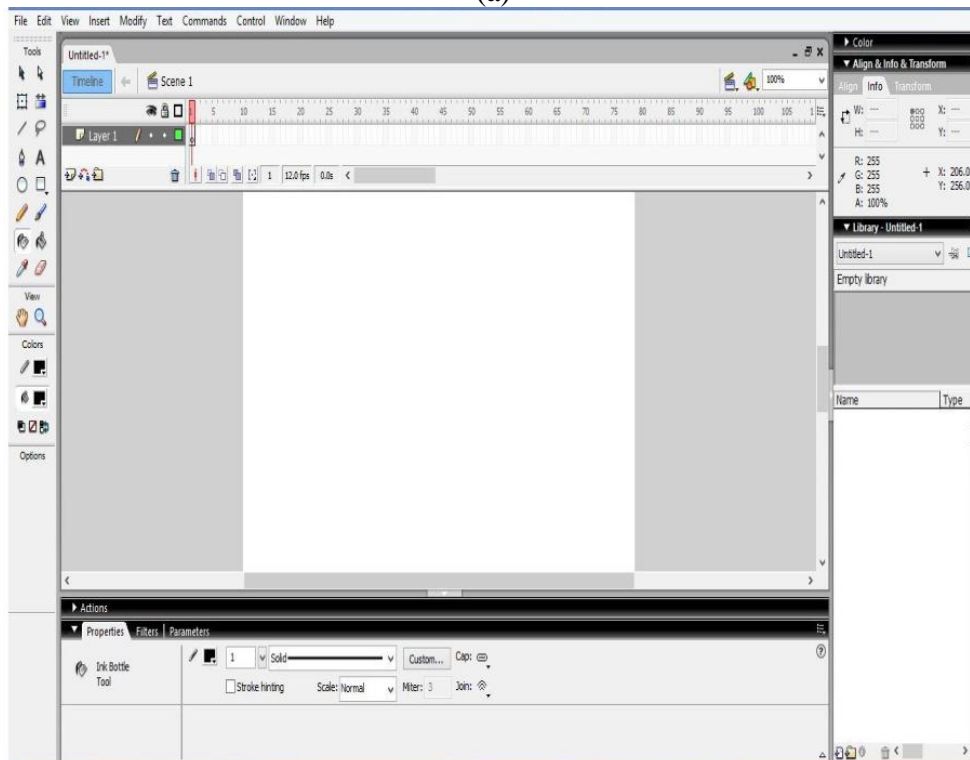
#### **4.2 Data Collection and Dataset Preparation**

##### **4.2.1 Synthetic data**

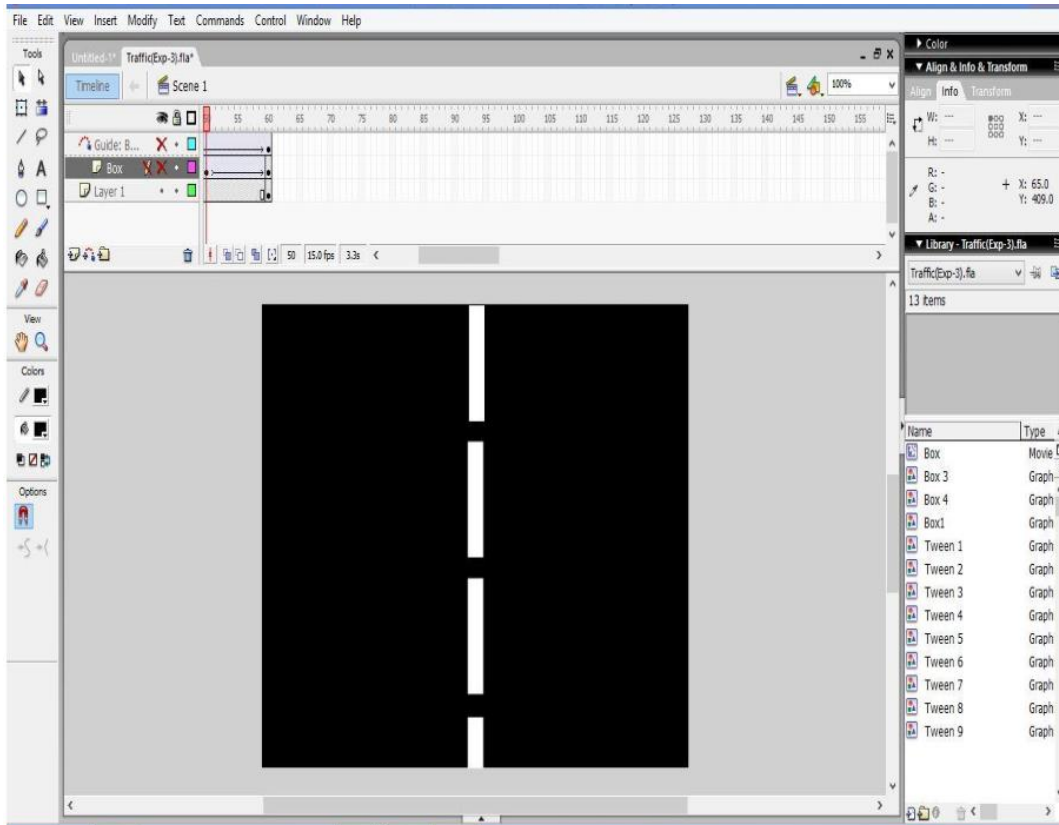
The synthetic video sequence is generated considering illumination variation, and camera displacement analogous to urban condition. The construction of synthetic video includes two steps: (1) frame production and (2) ground truth extraction. The major step in synthetic video construction is the frame production. The frame must consist of: (1) background (2) object. The background is compared with the pavement of the road and the object is compared with the distress on the road. The object shall be named as road distress object in these sections. For producing artificial video, various software is available. However, in this research work, Macromedia MX has been used. The software mostly uses the basic drawing concepts of which makes it flexible and user friendly. At the same time, it supports object importing from the Microsoft or Autodesk applications. For producing a frame, the steps of generating synthetic video using Macromedia MX is given in Figure 4.1 with proper illustration.



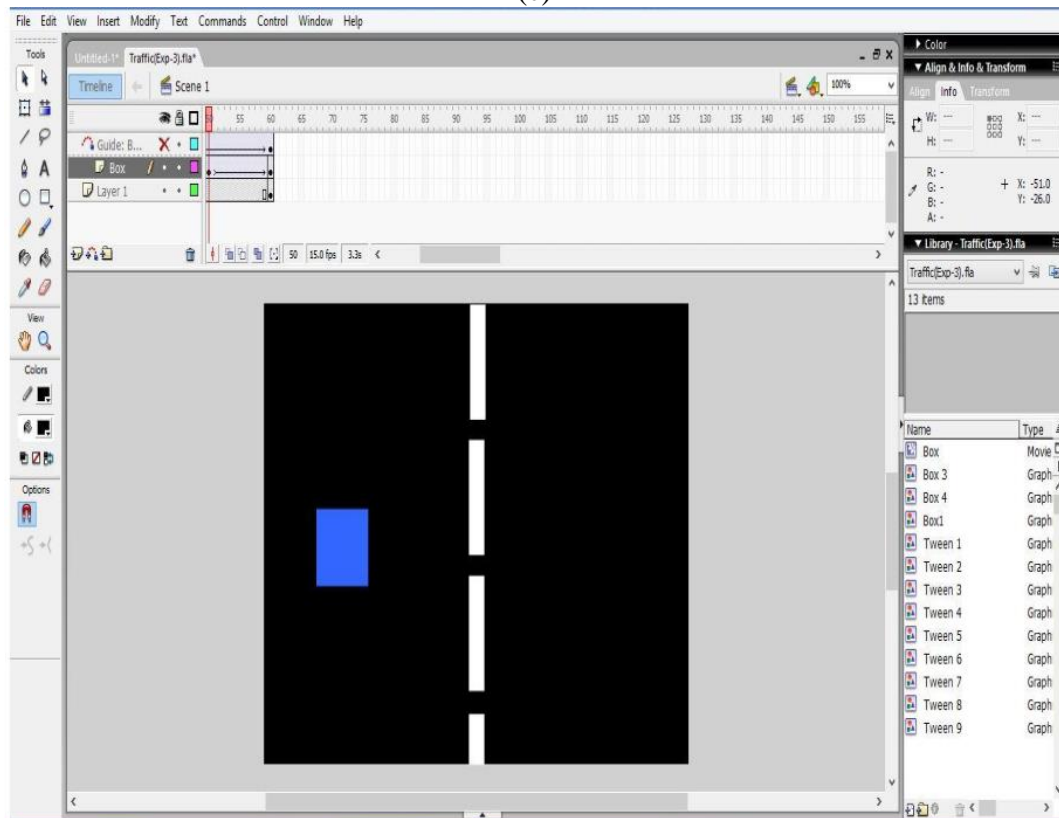
(a)



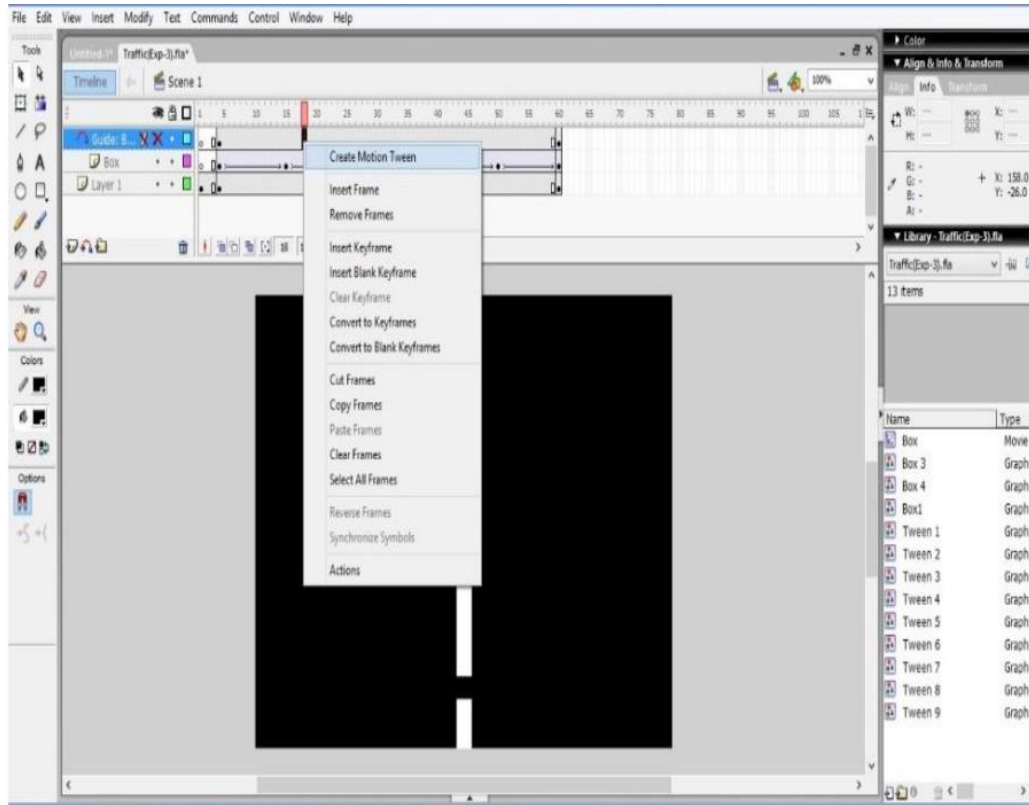
(b)



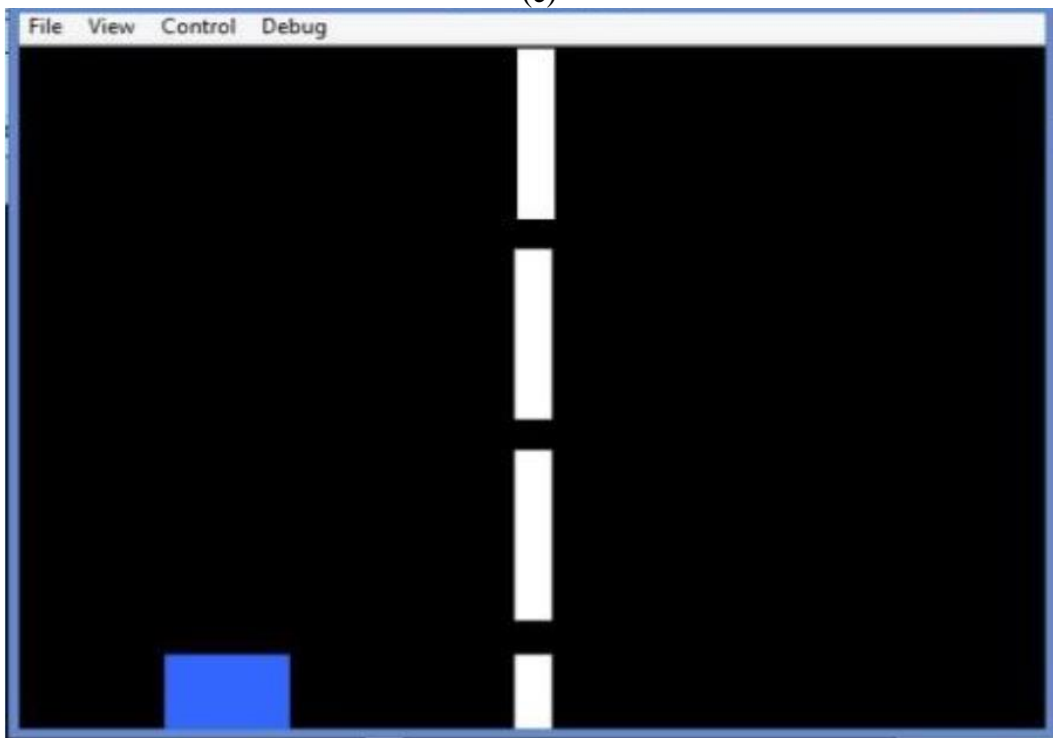
(c)



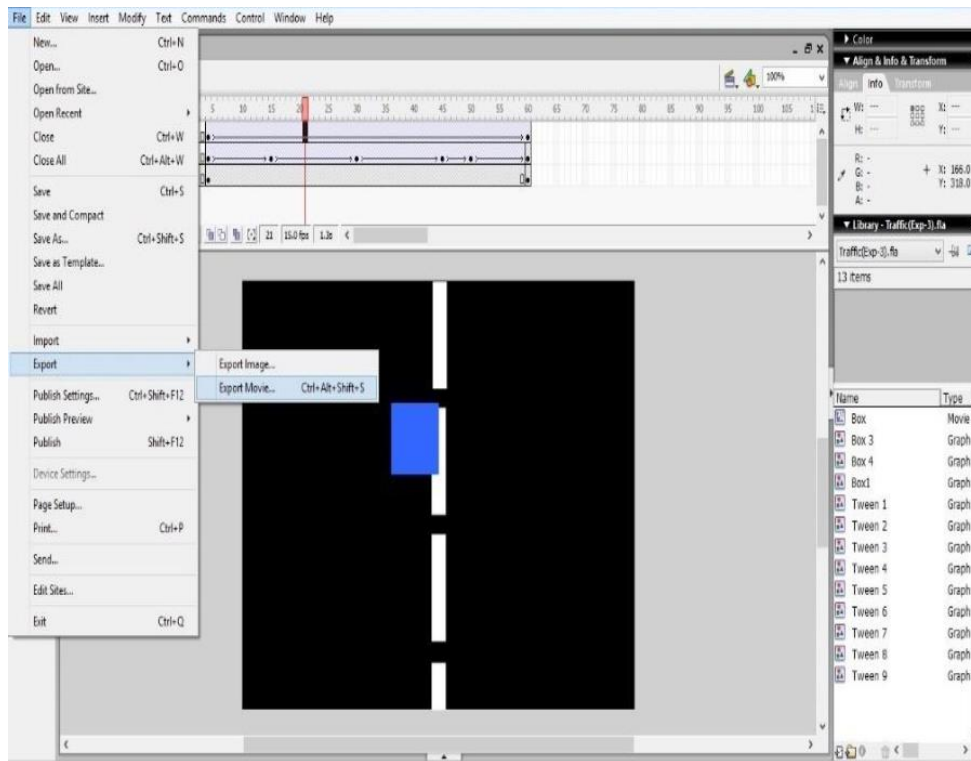
(d)



(e)



(f)



(g)

**Figure 4.1: Process of Generating Synthetic Data**

## 4.2.2 Real-time image data collection

For classifying cracks into binary crack or non-crack, a standardized dataset was used. The dataset contains different types of crack image. The data is collected at the different street and roads in the Dhaka region by using a smart phone as the data sensor. The dataset is divided into two as negative and positive crack images for distress classification. Total 52000 images has been used in this research with 99 x 99 pixels with RGB channels (Fig 3.3). The dataset is generated from 4000 high-resolution images (varies from 3264 ×2448 pixel to 2816x2112 pixel) with the method proposed by Zhang et al (2014). High-resolution images have variance in terms of surface finish and illumination conditions.

### 4.2.2.1 Image resizing

Images from dataset varies from 2816x2112 pixels to 3264 ×2448 pixels before cropping. This variation in resolution cannot be adopted by the training algorithm, rather a constant size pixel is needed for all the images. Here in this study, all the input images are resized into 99x99 pixels considering time and resource constraints. Such size not only makes

the images uniform but also provided higher training speed with adequate image details for the training algorithm.

#### **4.2.2.2 Image augmentation**

In this study image data were augmented to increase image numbers in order to increase accuracy. In this study, to achieve a good compromise between computational cost and accuracy of the detection results (Instruction for Road Maintenance Planning, 1988; EC Policy Orientations on Road Safety, 2018), each sample is a 3-channel (RGB) 99×99-pixel image patch generated by the image data augmentation procedure described in the following steps:

1. A patch whose center is within  $f = 5$  pixels of the crack centroid is regarded as a positive patch; otherwise it is considered as a negative patch.
2. To reduce the similarity between training samples, the overlap of two positive patches  $P1$  and  $P2$ , expressed as  $O = \text{area}(P1 \cap P2) / \text{area}(P1 \cup P2)$ , should be kept at a low level. In this study, we choose the distance between the centers of two adjacent patches to be  $d = 0.75w$ , where  $w$  is the width of a patch. For the negative patches, two adjacent patches should have no overlap.
3. Given a patch center  $c$ , each candidate patch is rotated around  $c$  by a random angle  $\alpha \in [0^\circ, 360^\circ]$ . This plays an important role to increase the number of crack samples because crack patches only consist of a small proportion of the collected images.

#### **4.2.2.3 Data organization**

Out of the generated samples from the above steps, 26000 samples are used as total dataset, 18200 samples are used as for training and the other 7800 samples are used as the testing purpose. From the generated data, the images are put into two folders naming ‘training set’ and ‘Testing set’ for testing and validation purpose. In each folder, there are 11 subfolders containing 9 classes of road distress image and remaining one is non-crack image. All the images in the folder are labeled with the type and a number to give them a serial. In total 11 classes of images were distinguished and stored in separate hard-disk

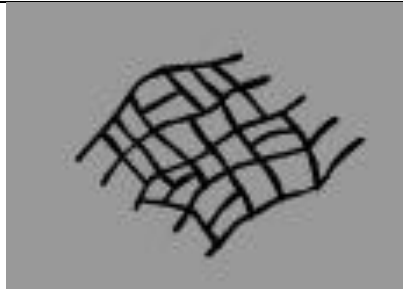
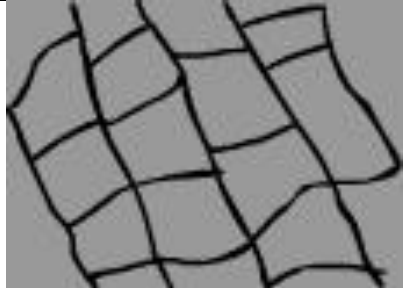



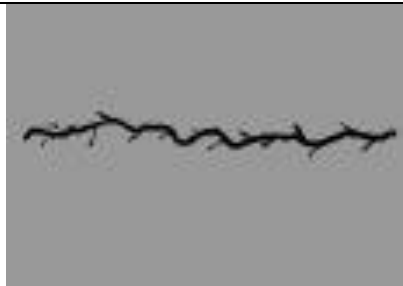




drive folder. Afterwards, these images are included in a dataset named ‘**RoadCrack2019**’ and compressed in a .zip folder to ensure shareability.

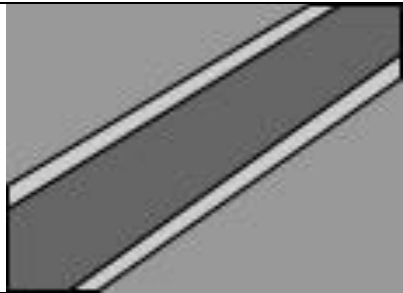
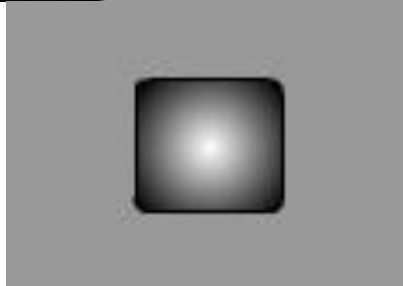

### 4.3 Data Analysis

In this section, the distance between the patch center and the crack center is determined. The percent mean overlapping among the synthetic and real-time image has also been calculated. Table 4.1 and Table 4.2 shows the summary of the analysis on the synthetic image and the real-time captured image.


**Table 4-1: Analysis Result of the Synthetic Data**





Serial	Name of the Distress	Illustration	Number of Images	Distance, f	Percent Mean Overlapping, O
1	Fatigue		500	6.4038	55.1699
2	Block Cracking		500	3.2655	36.6024
3	Corrugation and Shoving		500	6.2187	70.9343





Serial	Name of the Distress	Illustration	Number of Images	Distance, f	Percent Mean Overlapping, O
4	Longitudinal Crack		500	6.1742	65.6452
5	Patching		500	5.7427	48.9096
6	Polished Aggregate		500	5.0681	41.0718
7	Potholes		500	6.1895	52.8668
8	Ravelling		500	6.4778	60.2066



Serial	Name of the Distress	Illustration	Number of Images	Distance, f	Percent Mean Overlapping, O
9	Rutting		500	5.0988	46.7264
10	Depression		500	6.7119	62.4985
11	No Crack		500	5.8017	47.6345

**Table 4-2: Analysis Result of the Real-Time Data**

Serial	Name of the Distress	Illustration	Number of Images	Distance, f	Percent Mean Overlapping, O
1	Fatigue		1580	0.7207	1.4963

Serial	Name of the Distress	Illustration	Number of Images	Distance, f	Percent Mean Overlapping, O
2	Block Cracking		1792	2.7426	1.9012
3	Corrugation and Shoving		1641	1.7107	1.7935
4	Depression		1606	0.6840	0.7040
5	Join Reflection crack		1568	1.2204	0.5679

Serial	Name of the Distress	Illustration	Number of Images	Distance, f	Percent Mean Overlapping, O
6	Longitudinal Crack		1617	3.7870	1.3222
7	Patching		1566	2.1245	2.0669
8	Polished Aggregate		1549	0.7034	2.0764
9	Potholes		1656	2.5087	1.2191

Serial	Name of the Distress	Illustration	Number of Images	Distance, f	Percent Mean Overlapping, O
10	Rutting		1236	0.9840	1.7040
11	No Crack		12000	N/A	N/A

#### 4.4 Summary

In this chapter, data collection, data augmentation, and data analysis which are three major important factors of the preprocessing task has been included. Again, dynamic image dataset divided into two parts: (a) Synthetic Data; and (b) Real-time Data. An artificial video using Macromedia MX has been used as synthetic data. After image augmentation about 500 pictures from each synthetic image class are generated. In case of real-time data set with more than 3500 pavement static image of size  $3264 \times 2448$  are collected at the different street and roads in the Dhaka region by using a camera as the data sensor. Using these images augmentation technique is performed and the data set turned into 26000 samples.

# CHAPTER 5

## OPTIMIZATION OF THE MODEL PARAMETERS

---

### 5.1 Introduction

Optimizing hyperparameters in Convolutional Neural Network (CNN) is a tedious problem for many researchers and practitioners. To get hyperparameters with better performance, experts are required to configure a set of hyperparameter choices manually. The best results of this manual configuration are thereafter modeled and implemented in CNN.

### 5.2 Optimization Process

Building CNN requires a set of configurations which is external to the data and manually tune by the machine learning researcher. The variable of the network structure and the network trained of CNN are known as hyperparameters (Chambon and Moliard, 2011). Finding a set of hyperparameters that gives an accurate model in a reasonable time is also part of the hyperparameter optimization problem.

#### 5.2.1 Parameters of the model

An actual machine learner will ask the machine to perform this exploration and configure the optimal model architecture automatically. The variable in the configuration can be called hyperparameters which it is external to the model, and the value cannot be estimated from the data. Hyperparameters can be divided into two types:

a) Hyperparameter that determines the network structure such as:

1. Kernel Size –the size of the filter.
2. Kernel Type–values of the actual filter (e.g., edge detection, sharpen).
3. Stride–the rate at which the kernel pass over the input image.
4. Padding–add layers of 0s to make sure the kernel pass over the edge of the image.
5. Hidden layer–layers between input and output layers.
6. Activation functions–allow the model to learn nonlinear prediction boundaries.

b) Hyperparameter that determines the network trained such as:

1. Learning rate–regulates on the update of the weight at the end of each batch.

2. Momentum—regulates the value to let the previous update influence the current weight update.
3. A number of epochs—the iterations of the entire training dataset to the network during training.
4. Batch size—the number of patterns shown to the network before the weights are updated.

### 5.2.2 Parameter optimization using synthetic data

The main purpose of using synthetic data is to get the preliminary idea of the initial values and scales of the hyper parameter. The use of synthetic data reduces consumption of time resources by reducing the calibration time. Thus, the synthetic data has been used in the first place to identify the initial values of the hyper parameter. The candidate parameter values of the hyperparameters are chosen to be Max Epoch = {1,3,5,7,9}, Training Data = (60%, 65%, 70%, 75%, 80%), Learning Rate = {1E-5, 2E-05, 3E-05, 4E-05, 5E-05}. Number of Filters = {10,20,30,40,50}. In total, the training has been iterated for 625 times. And after finishing the iteration, the following optimum value has been found:

**Table 5-1: Preliminary Calibrated Parameters Values Using Synthetic Data**

<b>Calibrated parameter values from Synthetic Data</b>	<b>Training Accuracy</b>	<b>Testing Accuracy</b>
Max Epoch = 7 Training Data = 70% Learning Rate = 2E-05 Number of Filters = 30	99.98%	98.68%

It can be seen that the training accuracy has reached to 99.98% after iterating 625 times. It should be noted that each of this iteration took around 20 minutes to complete each iteration. In total 9 days were required to complete the 625 iterations.



### 5.2.3 Parameter optimization using real time data

Using the initially identified parameter value from the previous sub-section, the actual calibrated parameter value is determined in this sub-section. Figure 5.1 shows the relation between the batch accuracy of the training algorithm and the classification accuracy of the trained model.

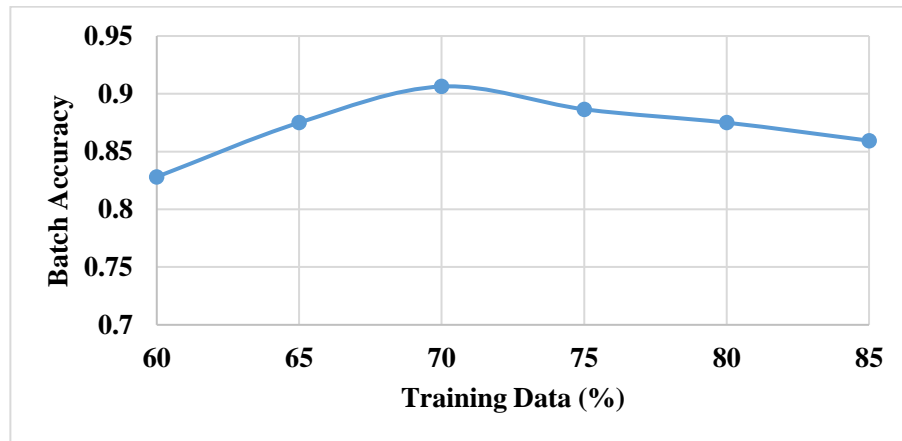


Figure 5.1: Accuracy vs Training data (%)

From the Figure 5.2 it is observed that batch accuracy is increased with classification accuracy. The graph is also showing classification accuracy vs batch accuracy for different learning rates. When batch accuracy is close to 100% the classification accuracy goes to 60-65% (parameters are still uncalibrated).

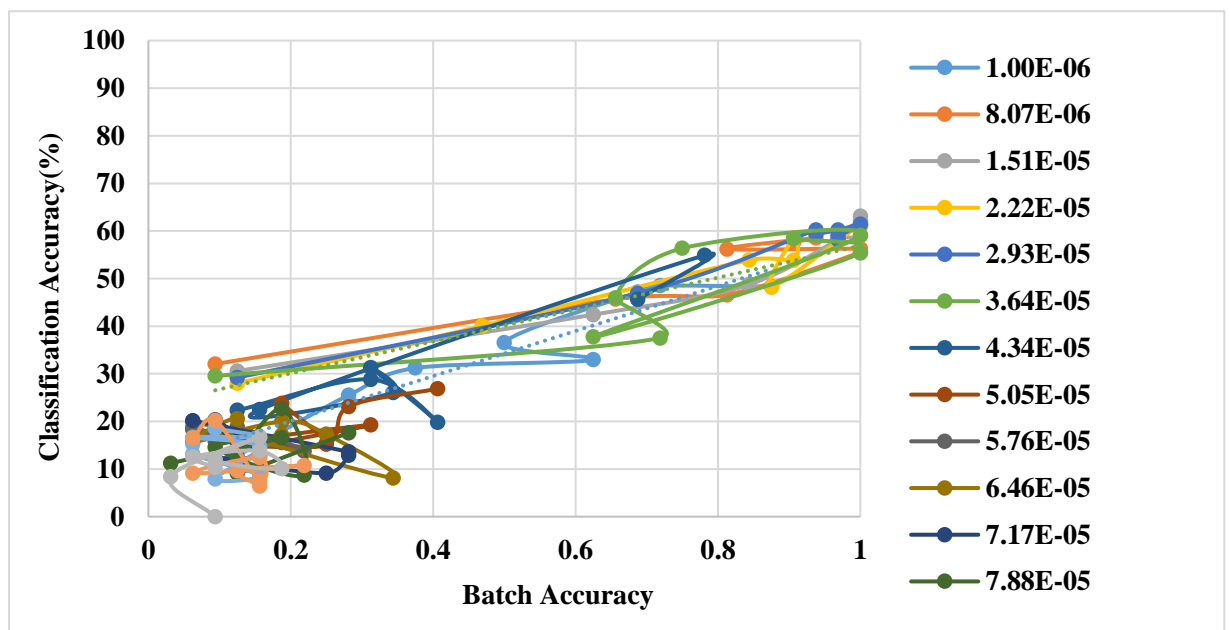
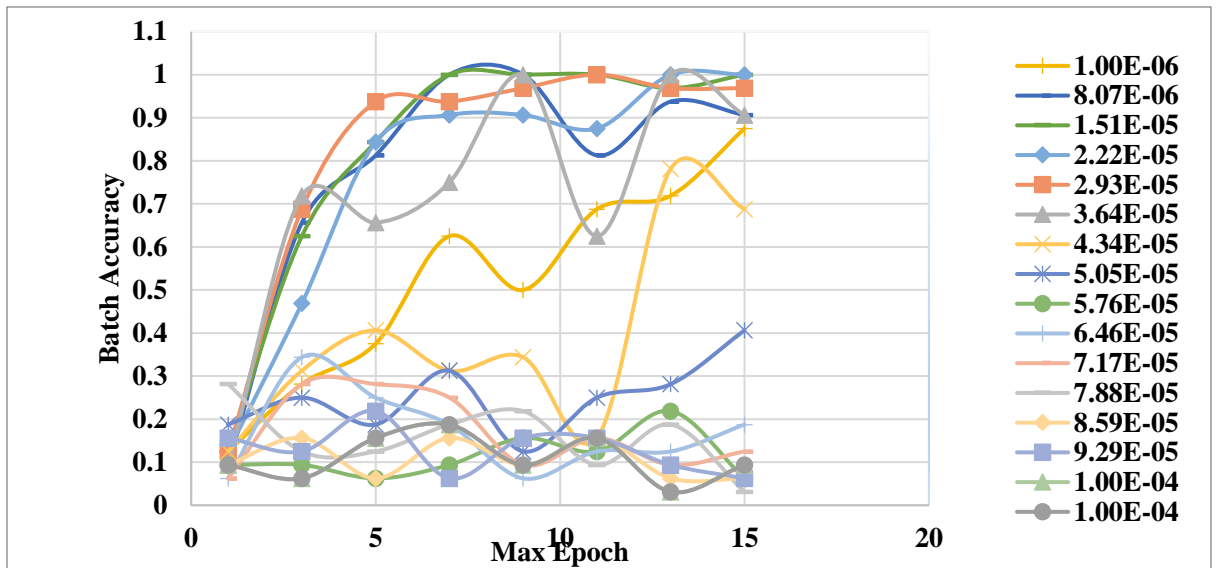


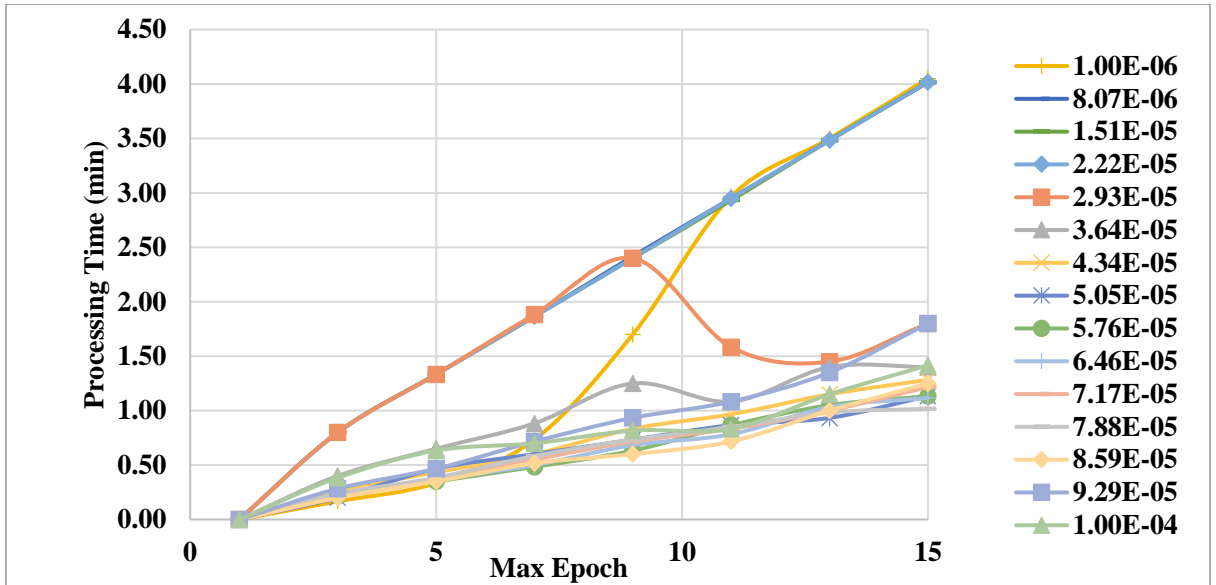
Figure 5.2: Classification Accuracy Vs Batch Accuracy For Different Learning Rates

From the figure 5.3, it is observed for certain learning rate ( $3.64 \times 10^{-5}$ ) that batch accuracy is increased with maximum epoch. But, for the learning rate below this certain value, the model is under fitting for max epoch vs batch accuracy. Similarly, for the learning rate above the certain value the model is overfitting. So, for learning rate  $3.64 \times 10^{-5}$  the model shows the best performance as after this point the model become unstable.



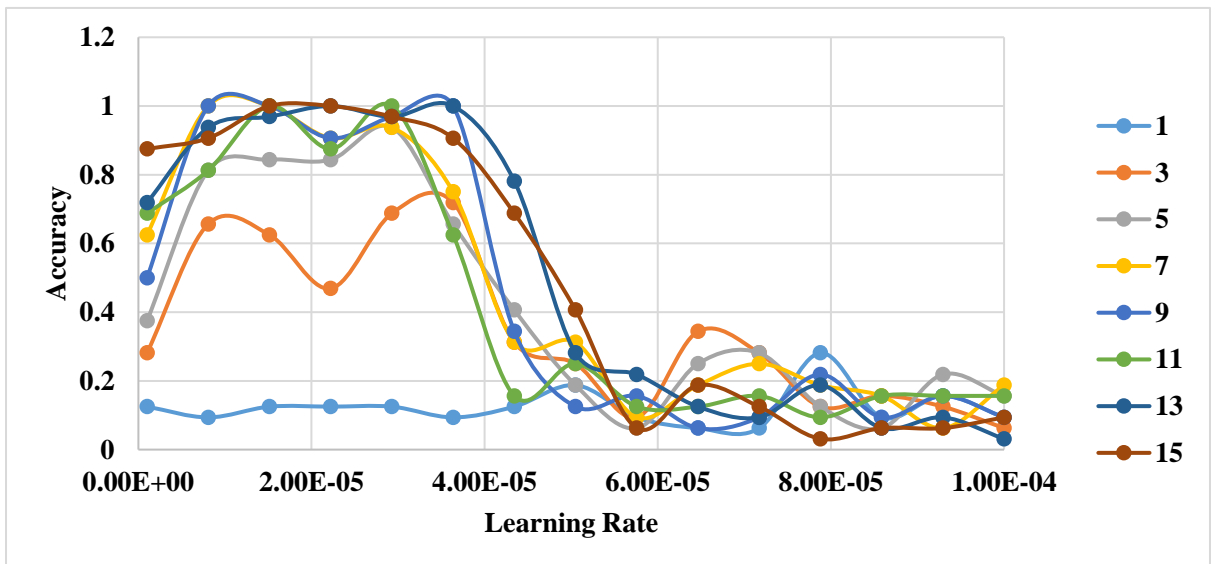
**Figure 5.3: Max Epoch vs Batch Accuracy for Different Learning Rates**

Figure 5.4 shows processing time vs max epoch for different learning rates. It is observed that processing time increases with max epoch. It is also observed that processing time is decreased with increasing learning rate. Though processing time is decreased with increasing learning time, there will be a risk of overfitting due to increase of learning rate. Because with increasing learning rate, there is a probability to propagate the model to local solution rather than global solution.



**Figure 5.4: Processing Time Vs Max Epoch for Different Learning Rates**

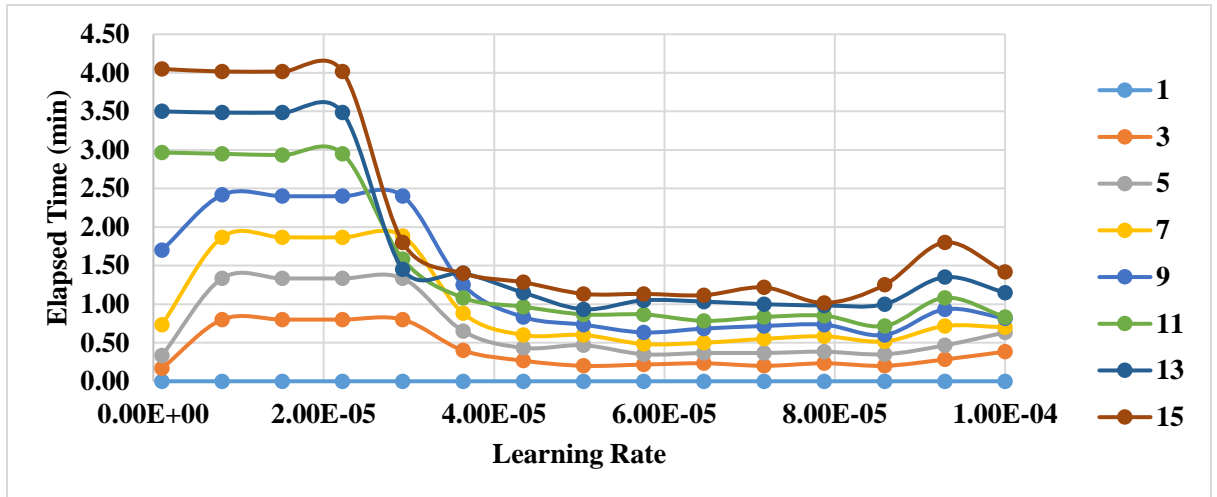
The figure 5.5 shows accuracy vs learning rate for different max epoch. For a certain range of the value of learning rate ( $<3.9 \times 10^{-5}$ ), the model shows the benefit of increasing max epoch and the accuracy rises with the increasing learning rate and max epoch number. However, beyond that limit of learning rate, accuracy does not increase with the increasing learning rate and max epoch number.



**Figure 5.5: Accuracy vs Learning Rate for Different Max Epoch**

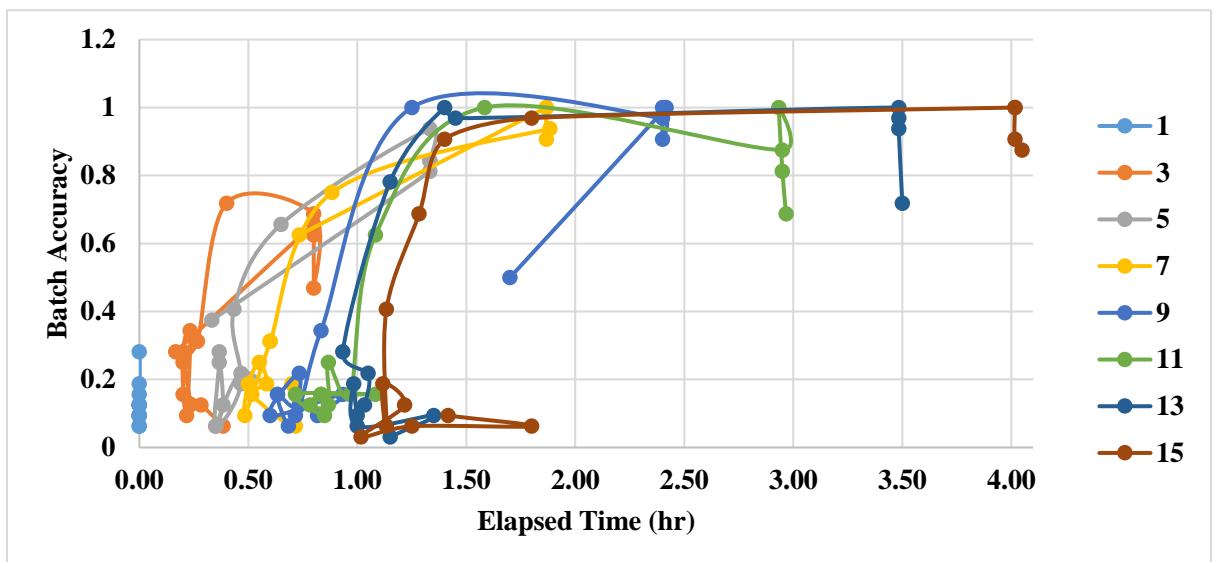
Figure 5.6 shows elapsed time vs learning rate for different epoch number. For a certain range of the value of learning rate ( $<3.9 \times 10^{-5}$ ), the model shows the benefit of increasing

max epoch and the elapsed time rises with the increasing learning rate and max epoch number. However, beyond that limit of learning rate, elapsed time does not increase with the increasing learning rate and max epoch number.



**Figure 5.6: Elapsed time vs Learning rate for Different Epoch Numbers**

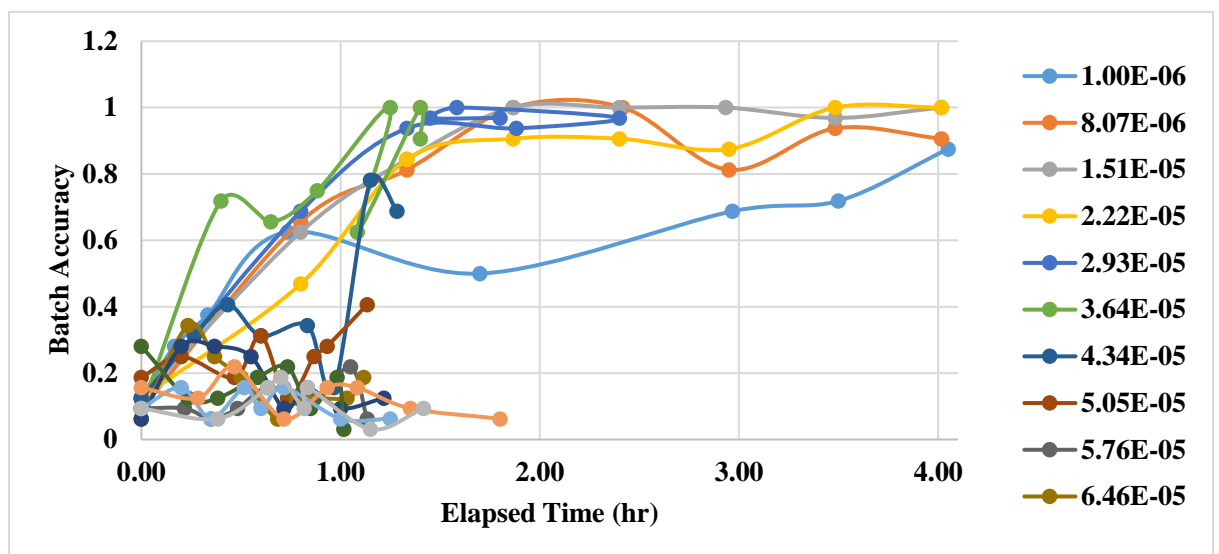
Figure 5.7 shows batch accuracy vs elapsed time for different epoch number. For epoch 9, the model reaches in 100% accuracy within the shortest possible time.



**Figure 5.7: Batch Accuracy Vs Elapsed Time for Different Epoch Numbers**

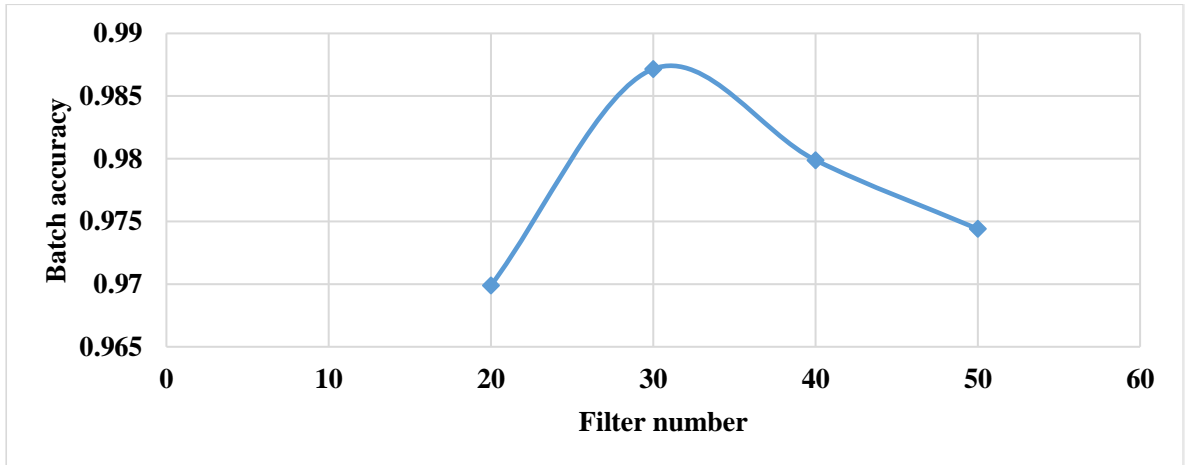
Figure 5.8 shows batch accuracy vs elapsed time for different learning rates. For higher learning rate ( $> 4.34 \text{ E-}05$ ), the model does not reach the 100% accuracy. For learning rate  $3.46 \text{ E-}05$ , the model reaches 100% accuracy within the shortest possible time.

However, this accuracy does not sustain for the time being. For learning rate  $2.93 \text{ E-}05$ , the model reaches towards solution within 2nd lowest possible time, but the solution sustains over the time being. From the above Analysis, it is observed that with increasing learning rate accuracy decreases and processing time also decreases. At the same time with increasing epoch processing time increases, and accuracy also increases. However, after a certain value of epoch with increasing epoch accuracy does not have significant change. From this analysis the optimum value for learning rate and max epoch has been determined as  $3.46 \text{ E-}05$  and 9 respectively where maximum accuracy can be obtained with lowest possible time.



**Figure 5.8: Batch Accuracy vs Elapsed Time for Different Learning Rates**

The Figure 5.8 shows Accuracy vs Filter number with respect to determined optimum learning rate and maximum epoch. The other hyper parameters are kept optimum as found before. It is observed that for filter number 30 maximum accuracy can be obtained for optimum learning rate and max epoch value.



**Figure 5.9: Accuracy vs Filter Number for Optimum Epoch and Learning Rate**

The Figure 5.9 shows accuracy vs training data. From the figure it is observed that for 70% training dataset maximum accuracy has been obtained.

From 5.2.2 it is noted that with increasing learning rate, accuracy decreases, and processing time also decreases. At the same time with increasing epoch processing time increases and accuracy also increases. However, after a certain value of epoch with increasing epoch accuracy does not have significant change. In this regard, the maximum learning rate and minimum epoch where accuracy is maximum will be the optimum value. From this analysis the optimum value for learning rate and max epoch has been determined as  $3.46 \text{ E-}05$  and 9 respectively. For filter number 30 and 70% training dataset maximum accuracy can be obtained.

**Table 5-2: Optimized Parameters of the Model for both Real-Time data**

Calibrated parameter values for real-time data	Training Accuracy	Testing Accuracy
Max Epoch = 9 Training Data = 70% Learning Rate = $3.64\text{E-}05$ Number of Filters = 30	99.22%	97.68%

### 5.3 Summary

In this chapter, calibration and validation of all the models using different data set are presented. Calibration of the CNN model is done using graphical method. At first, the

preliminary calibration has been done using the synthetic data to obtain initial starting point of the parameter values. Afterwards, the in-depth calibration is done using the real-time images. From the analysis it has been found that if the training data is taken 70% of the total data, initial learning rate is kept as  $3.64E-05$ , number of filters is used as 30 and the max epoch is taken as 10, the model yields the maximum training accuracy of 99.22% with a testing/validation accuracy of 97.68%.

## CHAPTER 6

### RESULTS AND DISCUSSION

---

#### 6.1 Introduction

The calibration of the selected models involved the use of a training dataset and a validation dataset. At the stage of this research, the parameters used for the training are set as default ones according to the authors of the models (Mario and Maltoni, 1997; Vincent et al., 2006). As a result of the implemented model, the recognition stage implies the detection of the damage location in the image and the recognition of the damage type. Additionally, in order to further improve the precision of the model and reduce the inaccuracies (i.e., false-positive or errors in the damage type recognition), the fine-tuning of the model hyperparameters is suggested for future developments.

#### 6.2 Road Distress Identification Result

This chapter includes a detailed analysis over the compiled data. Initially the correlation among the variables is revealed. Afterwards, the data has been fitted into newly proposed model and the performance of the models have been compared. Ultimately, compares the accuracy of the baseline method and CNN, with CNN achieving 99.22% accuracy, which is significantly higher than that for the baseline method.

**Table 6-1: Sample Optimization Output of a CNN Train Process**

Epoch	Iteration	Mini-batch Accuracy	Mini-batch Loss
1	1	10.16%	11.0402
1	50	58.59%	2.0749
2	100	73.44%	0.9859
2	150	82.81%	0.5868
3	200	86.72%	0.4832
3	250	89.06%	0.4182
4	300	87.50%	0.4885
5	350	94.53%	0.2808
5	400	93.75%	0.2021
6	450	96.88%	0.1932
6	500	97.66%	0.0818
7	550	98.44%	0.1027
8	600	98.44%	0.0735
8	650	99.22%	0.0537
9	700	99.22%	0.037



<b>Epoch</b>	<b>Iteration</b>	<b>Mini-batch Accuracy</b>	<b>Mini-batch Loss</b>
9	750	99.22%	0.0817

From Table 6-1, it is observed that for iteration 650 maximum accuracy 99.22% has been achieved and after certain iteration 650, maximum accuracy has been reached the saturation limit.

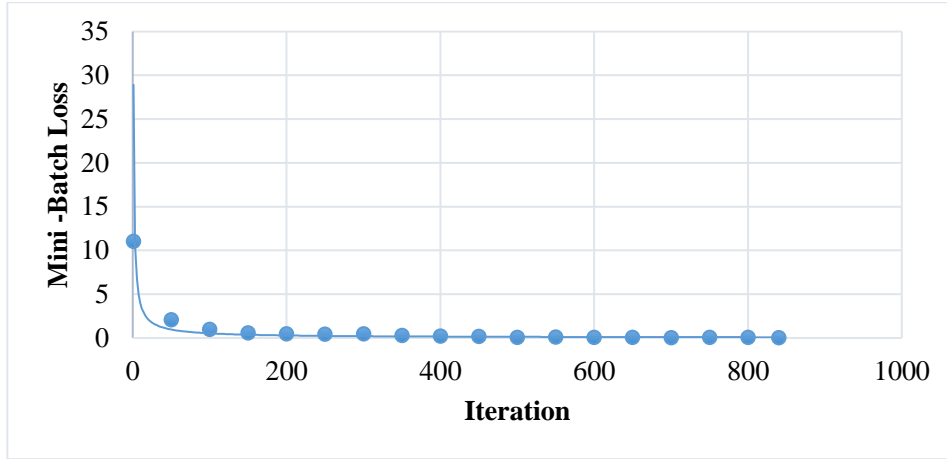
The trained models were evaluated using some known metrics, including the Jaccard Index which is useful for defining the percentage of overlap between the label and the prediction of the neural network, and the threshold score applied to exclude those predictions with a probability lower than a selected value of threshold. These parameters determine two acceptance thresholds for the results produced by the neural networks. Moreover, further evaluation metrics, such as Precision and Recall, were computed from the confusion matrix. In addition, the harmonic average between Precision and Recall, known as the F1-Score, was determined. The confusion matrix shows the performance of a classifier given some truth values/instances that is made up at least of four components for each label, which exemplifies three different instances. Generally, an algorithm labels a pixel as either positive or negative in a binary decision problem, where ‘positive’ and ‘negative’ respectively represent foreground pixel and background pixel. For a given frame in a video sequence, a comparison can be drawn between the resultant image and the ground truth image. A pixel is denoted as white when it is a part of an object in the foreground, and black when it actually belongs to the background. To quantify the classification performance with respect to ground-truth, four basic measures are used, such as, (1) true positives (TP): correctly classified foreground pixels; (2) true negatives (TN): correctly classified background pixels; (3) false positives (FP): incorrectly classified foreground pixels; and (4) false negatives (FN): incorrectly classified background pixels. In addition, Precision-Recall curves are good performance indicator providing an optimistic appraisal of the classifier’s performance when there is a significant skewness in the class distribution (Milan, 2002). Other measures for fitness quantification, in the context of background subtraction techniques, were proposed in the literature (Michael, 2003), such as, Percentage of Correct Classification (PCC) and F-measure.

$$\text{Precision (PR)} = \frac{TP}{TP + FP} \quad (6.1)$$

$$\text{Recall (RE)} = \frac{TP}{TP + FN} \quad (6.2)$$

$$PCC = \frac{TP + TN}{TP + FN + FP + TN} \quad (6.3)$$

$$F = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6.4)$$



**Figure 6.1: Change in Mini-Batch Loss with Iteration**

Figure 6.1 illustrates the learning curves recorded during the training phase for the calibrated parameters. The value shown on the vertical axis is the Total Loss, that is, the sum of the loss values useful for the training of the deep neural networks.

### 6.3 Comparison between CNN and the Baseline Method for the Classification Task

Recently several other algorithms are used for image classification e.g., Support Vector Machine (SVM), Artificial Neural Network (ANN), Adaptive Neuro Fuzzy Inference (ANFIS). However, among these algorithms SVM performs best (Mahler et al., 2011; Soukup and Huber-Mö rk, 2014). Thus, SVM has been chosen as baseline method. For the experiment using the baseline method, we randomly selected the same number (26575) of crack images and non-crack images. We computed the average result over 10 trials. Table 3 compares the accuracy of the baseline method namely Support Vector Machine (SVM). and CNN, with CNN achieving 97.68% accuracy, which is significantly higher than that for the baseline method. Figure 5 shows some examples of images incorrectly detected by CNN. These images would seem to be hard to detect, even by human observation.

**Table 6-2: Confusion Matrix for Support Vector Machine (SVM)**

	Fatigue	Block Cracking	Corrugation and shoving	Depression	Joint Reflection cracking	Longitudinal cracking	Patching	Polished Aggregate	Potholes	Rutting	No Crack
Fatigue	0.6	0.05	0.07	0	0	0.06	0.02	0.07	0.07	0.02	0.04
Block Cracking	0	0.99	0	0	0	0	0	0	0	0.01	0.01
Corrugation and shoving	0.03	0	0.7	0	0	0.01	0.03	0.15	0.05	0.03	0.04
Depression	0	0	0	1	0	0	0	0	0	0	0
Joint Reflection cracking	0	0	0	0	1	0	0	0	0	0.01	0
Longitudinal cracking	0.02	0.12	0.07	0	0.01	0.53	0.1	0.03	0.05	0.03	0.07
Patching	0	0.04	0.04	0.01	0	0.02	0.76	0.01	0.1	0.02	0.02
Polished Aggregate	0	0	0.01	0	0	0.01	0	0.98	0.01	0.01	0
Potholes	0.11	0.04	0.04	0.02	0.01	0.05	0.04	0.02	0.67	0.05	0.01
Rutting	0.03	0.02	0.05	0.01	0	0.01	0.06	0.01	0.07	0.75	0.01
No Crack	0	0.01	0.01	0.01	0	0.01	0.03	0.01	0.01	0.03	0.89

**Table 6-3: Confusion Matrix for CNN**

	Fatigue	Block Cracking	Corrugation and shoving	Depression	Joint Reflection cracking	Longitudinal cracking	Patching	Polished Aggregate	Potholes	Rutting	No Crack
Fatigue	0.86	0.00	0.03	0.00	0.00	0.05	0.00	0.01	0.03	0.00	0.01
Block Cracking	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Corrugation and shoving	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Depression	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Joint Reflection cracking	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Longitudinal cracking	0.00	0.00	0.01	0.00	0.00	0.98	0.01	0.00	0.00	0.00	0.01
Patching	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00

	Fatigue	Block Cracking	Corrugation and shoving	Depression	Joint Reflection cracking	Longitudinal cracking	Patching	Polished Aggregate	Potholes	Rutting	No Crack
Polished Aggregate	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Potholes	0.02	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.96	0.00	0.00
Rutting	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
No Crack	0.01	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.96

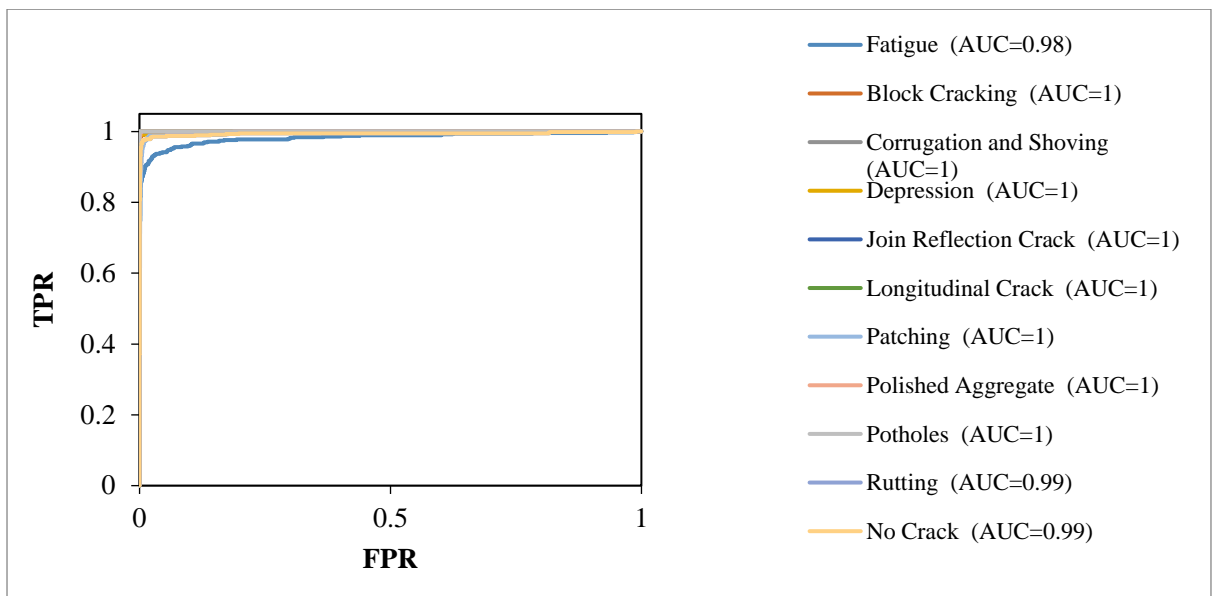
\*Red cells show the class of high probability. Blue cells show confusion probabilities exceeds 10%.

Table 6.2 shows the confusion matrix for the crack and non-crack image using SVM (Support vector machines) method and CNN (Convolutional Neural Network) method. From the confusion matrix, it is observed that CNN gives more accuracy than SVM method. In case of SVM method, the model obtained 100% in two cases, more than 90% accuracy in 2 cases, more than 80% accuracy in one case and more than 50% accuracy in 5 cases. On the other hand, In case of CNN method, the model obtained 100% in 5 cases, more than 90% accuracy in 4 cases, more than 80% accuracy in one case. From the above-mentioned analysis, it can be finalized that CNN gives better and higher accuracy than SVM.

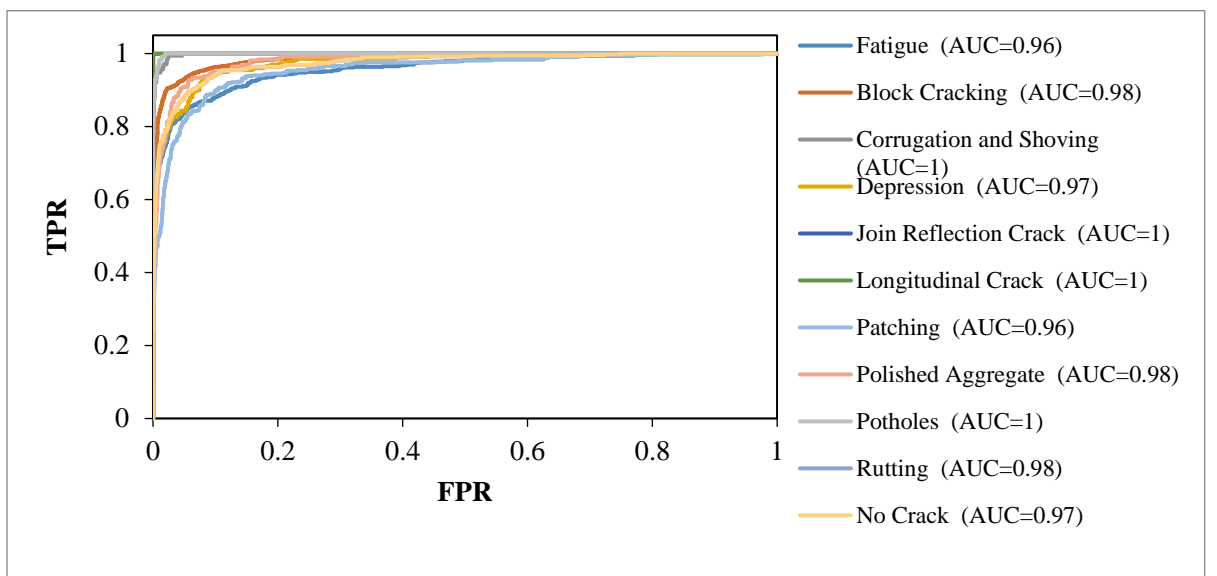
**Table 6-4: Comparison of Accuracy between CNN and SVM Method for Crack Detection and Crack Classification**

Crack Detection Accuracy			Crack Classification Accuracy	
Measure of performance	SVM	CNN	SVM	CNN
Precision	0.91562	0.9687	80.30%	97.68%
Recall	0.97441	0.9978		
PCC (Percent of Correct Classification)	0.94166	0.9813		

From table 6.4 the value of precision as a measure of performance for crack detection accuracy in SVM method is 91.56% on the other hand, in CNN method it is 96.87%. Similarly, the value of Recall and PCC (percent of correct classification) as a measure of performance for crack detection accuracy is significantly higher for CNN than SVM. From above explanation, it can be finalized that in case of crack detection CNN provides better performance than SVM method. In case of classification accuracy, CNN provides 97.68% and SVM provides 80.32% accuracy which represents that CNN method perform better than SVM method.

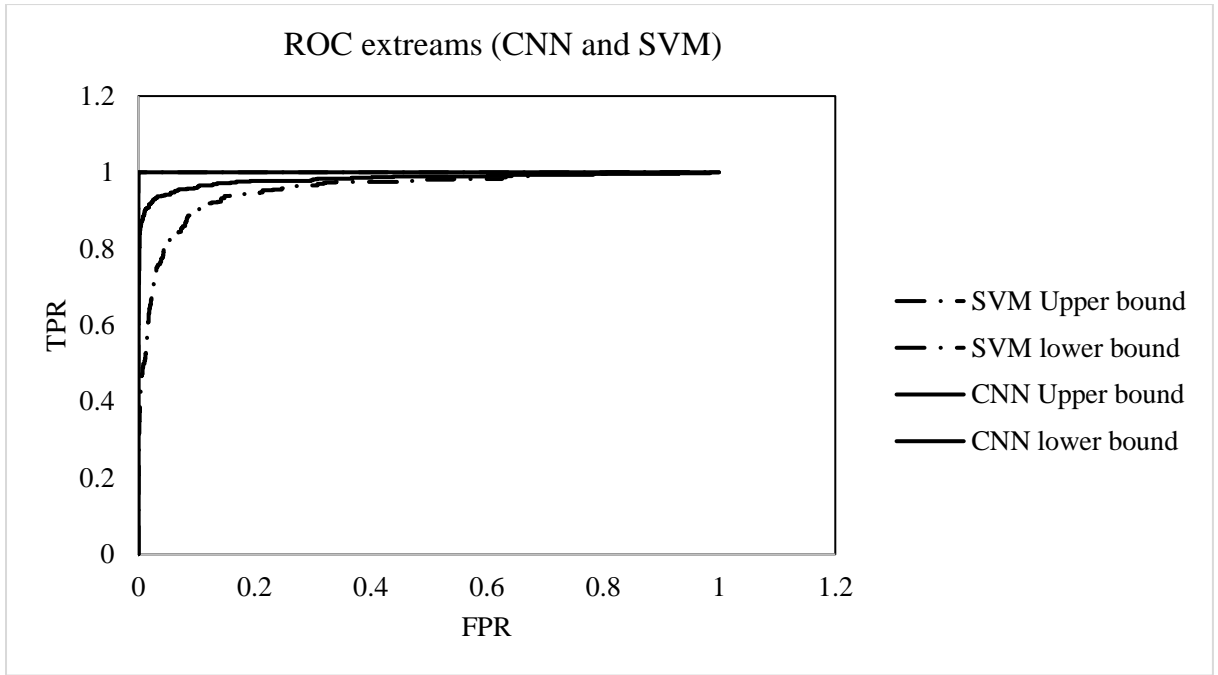


(a)

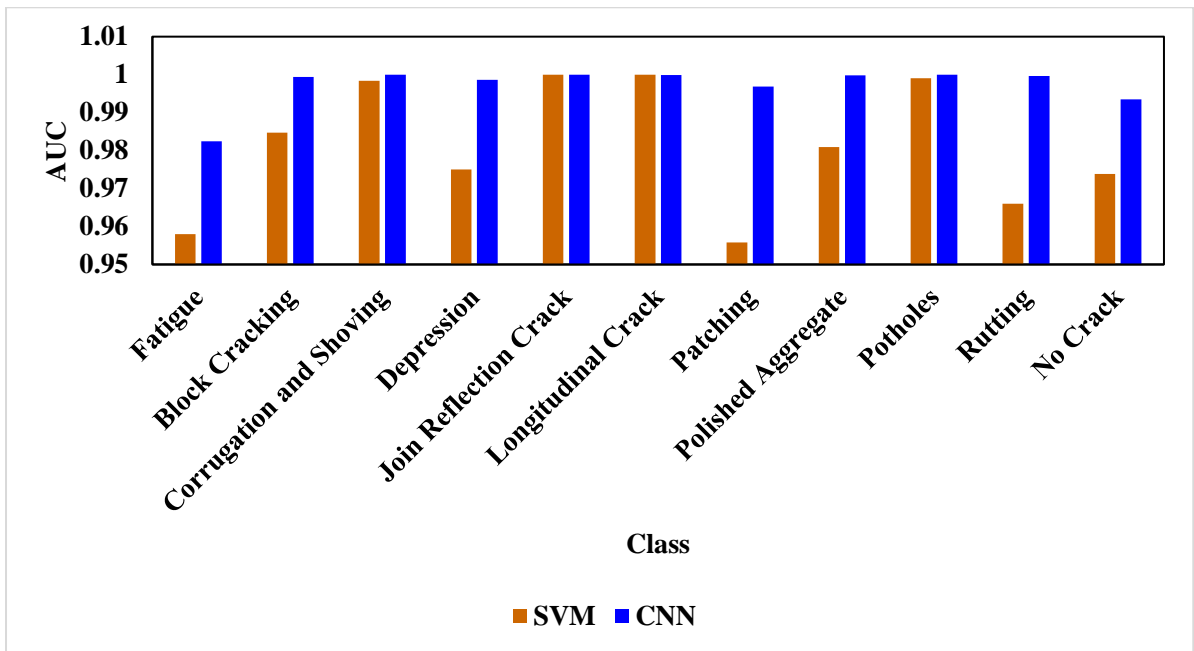


(b)

**Figure 6.2: ROC Curve for (a) CNN model (b) and SVM model**



(a)



(b)

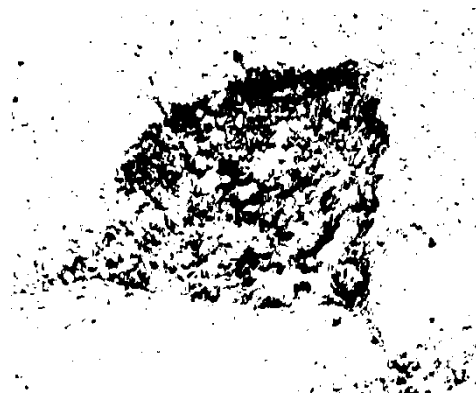
**Figure 6.3: Comparison between SVM and CNN Method with respect to ROC and AUC as a Measure of Performance**

From the ROC (Receiver Operating Characteristic) curve (Figure 6.2), it is observed that for SVM method the value of AUC (Area Under the Curve) for most of road distresses is lower than 1. On the other hand, for CNN method, the value of AUC (Area Under

Curve) for all type of cracks is 1 (Figure 6.3) except fatigue failure, which represent the better performance of CNN in classification of road distress.

#### **6.4 Field Testing and Pavement Scoring**

Field test has been conducted from Abul Hotel (23.7547° N, 90.4154° E) to Abdullahpur (24.3654° N, 91.1641° E) of the Dhaka region for crack detection and crack classification. The distance from Abul Hotel to Abdullahpur has been divided into two segments: (i) Abul Hotel to Kuril flyover; and (ii) Kuril flyover to Abdullahpur. The sample video that has been collected from the field. Afterwards, Sobel edge detection performed on the video to convert the RGB images into binary image. After the binary conversion, image morphological operation has been done to remove unnecessary noises and fill the void within the images. Ultimately, blob analysis is done to extract the crack features (i.e., area, centroid, orientation etc.) by creating a bounding box.



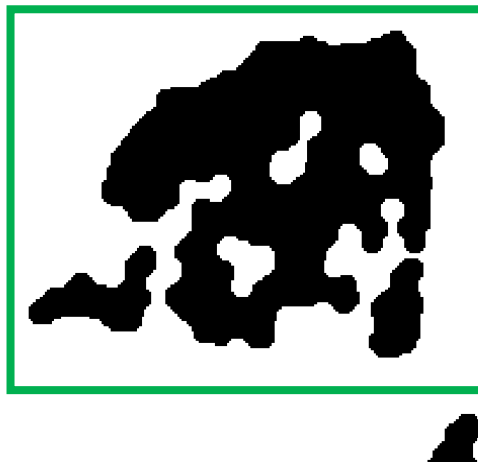
(a)



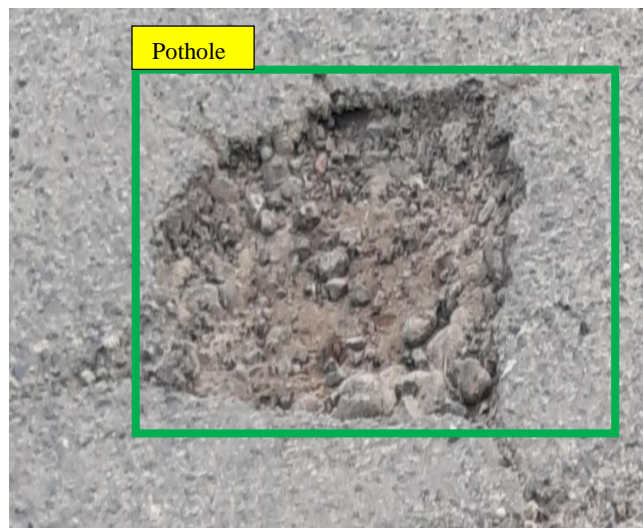
(b)



(c)



(d)



(e)

**Figure 6.4: Crack Detection from the Collected Video Sequence**



The above-mentioned procedure has been run through the calibrated and validated CNN model for crack detection and crack classification. The result has been shown in Table 6.4 and Table 6.5. The blob analysis extracts the features (orientation, length, density, displacement, location) of the crack. An equivalent patch size of 99×99 pixel was considered and the number of crack patches required to cover the whole distress region were determined. The total number of crack patches is directly related to the pavement health score. An increased number of total crack patches indicates deteriorated pavement health. A reduced number of crack patches indicates improved pavement health. An analysis of total number of crack patch before and after conducting a pavement treatment can give quantitative result of the improved pavement health.

Two types of road segment has been considered for field testing; (i) Urban City Road (Khilgaon to Kuril), (ii) National Highway N3 (Kuril road to Abdullahpur). It is observed from the Table-1 and Table-2 that urban city road has lower number of crack patches than National highway. As construction work of BRT and MRT is ongoing in Kuril to Abdullahpur road segment, number of cracks is higher.

**Table 6-5: Crack Classification of Abul Hotel to Kuril Road (Urban City Road)**

Type of the crack	Equivalent number of patches
Bleeding	3
Patching	57
Polished Aggregate	11
Raveling	11
Rutting	7
Stripping	5
Corrugation and Shoving	1
Alligator Crack	24
Block Cracking	0
Longitudinal Cracking	5
Slippage Cracking	5
Transverse Cracking	4
Pot Hole	135
Rutting	38
Total	306

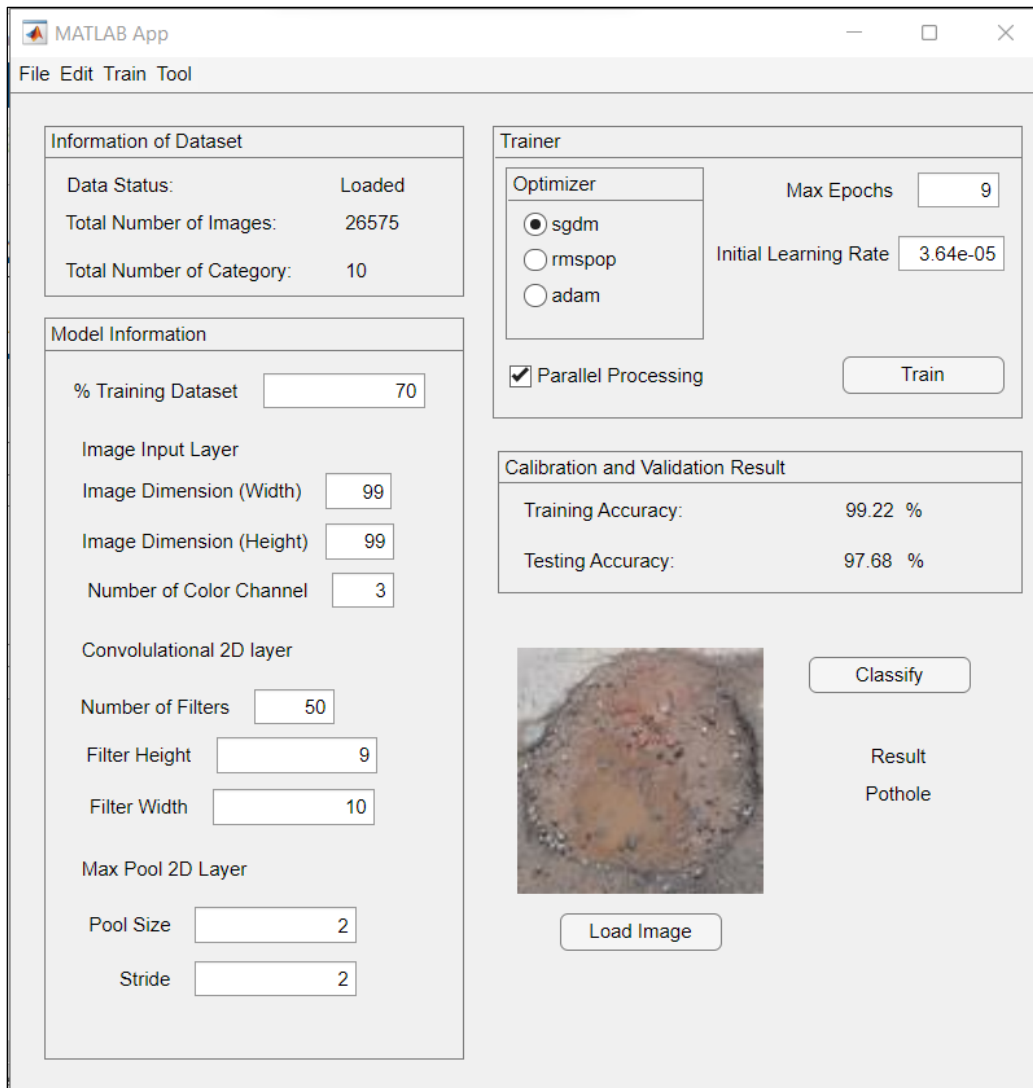
**Table 6-6: Crack Classification of Kuril Road to Abdullahpur (National Highway)**

Type of the crack	Equivalent number of patches
Bleeding	14

<b>Type of the crack</b>	<b>Equivalent number of patches</b>
Patching	135
Polished Aggregate	58
Raveling	112
Rutting	91
Stripping	32
Corrugation and Shoving	71
Alligator Crack	29
Block Cracking	10
Longitudinal Cracking	18
Slippage Cracking	15
Transverse Cracking	29
Pot Hole	194
Rutting	50
Total	858

### **6.5 Development of Tool for Practice**

A tool is developed to help practitioners for implementing the road distress classification paradigm named Road Distress Training and Classification (RoadDisTrac). The tool is designed in MATLAB runtime environment. The Tool uses a Graphical User Interface (GUI) to interact with user for taking commands. The tool requires 4 GB of minimum ram with processor clock speed greater than 2.6 GHz. If multiple core in the processor is available the tool will impart more faster calculation. Additionally, installing a Graphics Processor Unit (GPU) may reduce the runtime more drastically. The tool requires MATLAB RUNTIME ENVIROMENT to be installed in the machine to operate properly. This tool mainly facilitates training CNN models and classify any input image. The tool takes all the parameters using interacting the dialogue box and passes these parameters to the model.



**Figure 6.5: Interface of RoadDisTrac**

## 6.6 Summary

In this chapter, with the optimized value of different hyper-parameters the accuracy of crack detection for CNN method has been determined. From the confusion matrix, it is observed that CNN gives more accuracy than SVM method. In case of SVM method, the model obtained 100% in two cases, more than 90% accuracy in 2 cases, more than 80% accuracy in one case and more than 50% accuracy in 5 cases. On the other hand, In case of CNN method, the model obtained 100% in 5 cases, more than 90% accuracy in 4 cases, more than 80% accuracy in one case. From the above-mentioned analysis, it can be finalized that CNN gives better and higher accuracy than SVM. Field testing shows that the model is capable of classifying road distresses on movement incorporating Sobel edge detection.

## CHAPTER 7

### CONCLUSION AND RECOMMENDATIONS

---

#### 7.1 Concluding Remarks

This study investigated the capability of Convolutional Neural Network (CNN) algorithms to develop and test an innovative health-monitoring methodology for road pavements. For this purpose, in this study, several images of road distress in the Dhaka region were processed and encouraging results were obtained. Furthermore, this study aims at low-cost instrumentation with higher accuracy for detecting and classifying road distress using CNN. More comprehensive health monitoring and the assessment of the condition of road pavement. Overall, this thesis contributes to proving that CNN is a promising method for monitoring deformation phenomena and the identification and classification of damages in road pavements and, more in general, in transport infrastructures exposed to various natural hazards. This research paves the integration between remote-sensing data and non-destructive information collected on-site to improve and optimize the road maintenance system of the transport agency. The implementation of the model is to provide a technologically enhanced and reliable methodology for identification and classifying road distress which to be more rapidly processed and conclusively actioned by asset owners and management agencies, giving crucial information that could be implemented for the prioritization of maintenance activities within Pavements Management Systems (PMSs). The main key points from this research are summarized below.

- (1) From the analysis it has been found that if the training data is taken 70% of total data, initial learning rate is kept  $3.64E-05$ , number of filters is used as 30 and the max epoch is taken as 10 the models yields maximum training accuracy of 99.22% with a testing/validation accuracy of 97.68%.
- (2) From the confusion matrix, it is observed that CNN gives more accuracy than SVM method. In case of SVM method, the model obtained 100% in two cases, more than 90% accuracy in 2 cases, more than 80% accuracy in one case and more than 50% accuracy in 5 cases. On the other hand, In case of CNN method, the model obtained 100% in 5 cases, more than 90% accuracy in 4 cases, more than 80% accuracy in one case.

- (3) Field testing shows that the model is capable of classifying road distresses on movement incorporating Sobel edge detection.
- (4) The accuracy of the calibrated CNN model in crack identification is 98.78% which is compared with base line method (94.17%). In case of Classification of road distress CNN model (97.68%) perform better than base line method SVM (80.31%).

## **7.2 Recommendations for Future Research**

Since the CNN model for crack identification and classification has been studied for more than a decade in the developed world, research on this topic in Bangladesh as well as in other south-east Asian countries, is extremely scarce and challenging. This is mainly due to the complexity of data collection and processing and the absence of a proper mathematical framework. Even though the current study tried to focus some effort on this sector, it can't be viewed as a complete understanding of the highly complex behavior involved in road distress identification and classification. Although the model performed well and attained a well goodness of fit, there are some limitations of this study:

- (1) The main limitation of the developed CNN model in this study is not capable of capturing the background dynamics in night time pavement images.
- (2) The Patch size considered in the study to train the developed CNN architecture is  $99 \times 99$ . The CNN model can take input image size of  $99 \times 99$  only. If any other size of image is provided as input in the model will cause abnormal behavior of the program.
- (3) The CNN model developed using pavement images having background pixels consisting of only pavement. Pavement images with foreign objects (i.e. tree, sky, vehicle, pedestrian) have not been considered in this study.

Further research is to explore other forms of crack classification and identification of deep learning models with greater accuracy. In this section, some recommendations are provided for future research following the studies carried out in this dissertation. These are listed below,

- (1) The challenge with night time images can be investigated using more sophisticated camera with infrared (IR) facility. The CNN architecture needs to be redesigned to cope up with the IR images; which can lead towards road distress identification even in night time.
- (2) Presence of foreign objects (i.e. manhole cover, tree, vehicle, sky) in captured pavement image makes identification and detection task more complex. The CNN structure needs to be redesigned to cope up with the images with foreign objects. Moreover, sophisticated edge detection methods (i.e. Prewitt, Canny, fuzzy) needs to be adopted to cope up with presence of the foreign objects.
- (3) Field dimension of the road distress (i.e. width, length, depth, perimeter, area) can be obtained if the camera calibration is done. The camera calibration may include the correction for the perspective view of the camera and the estimation of the corresponding magnification factor.

## REFERENCES

---

- Ahmadlou, M. & Adeli, H. (2010), Enhanced probabilistic neural network with local decision circles: a robust classifier, *Integrated Computer-Aided Engineering*, 17(3), 197– 210.
- Albelwi, S., & Mahmood, A. (2017). A framework for designing the architectures of deep convolutional neural networks. *Entropy*, 19(6), 242.
- Albelwi, S., & Mahmood, A. (2017). A framework for designing the architectures of deep convolutional neural networks. *Entropy*, 19(6), 242.
- Alnuweiri, H. M., & Prasanna, V. K. (1992). Parallel architectures and algorithms for image component labeling. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 14(10), 1014-1034.
- ASTM D6433-18. (2018). Standard Practice for Roads and Parking Lots Pavement Condition Index Surveys; ASTM International: West Conshohocken, PA, USA.
- Aszemi, N. M., & Dominic, P. D. D. (2019). Hyperparameter optimization in convolutional neural network using genetic algorithms. *International Journal of Advanced Computer Science and Applications*, 10(6).
- Baker, S., & Nayar, S. K. (1996, June). Pattern rejection. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 544-549). IEEE.
- Barnich, O., & Van Droogenbroeck, M. (2009, April). ViBe: a powerful random technique to estimate the background in video sequences. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on* (pp. 945-948). IEEE.
- Bennett, C.R.; De Solminihac, H.; Chamorro, A. (2006). *Data Collection Technologies for Road Management*; World Bank: Washington, DC, USA.
- Bhattacharya, P. (1996). Connected component labeling for binary images on a reconfigurable mesh architecture. *Journal of systems architecture*, 42(4), 309-313.
- Bottou, L. (2012), Stochastic gradient descent tricks, in G. Montavon, G. B. Orr, and K.-R. Müller (eds.), *Neural Networks: Tricks of the Trade*, 2nd edn., Springer, Berlin Heidelberg, pp. 421–36.
- Canny, J. F. (1986). A computational approach to edge detection. *IEEE Trans Pattern Analysis and Machine Intelligence*, 8(6), 679-698.
- Canny, J. F. (1986). A computational approach to edge detection. *IEEE Trans Pattern Analysis and Machine Intelligence*, 8(6), 679-698.
- Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361-378.

- Chambon, S. (2011, March). Detection of points of interest for geodesic contours: Application on road images for crack detection. In International Joint Conference on Computer Vision Theory and Applications, VISAPP (p. sp).
- Chambon, S., & Moliard, J. M. (2011). Automatic road pavement assessment with image processing: review and comparison. *International Journal of Geophysics*.
- Chan, C.Y.; Huang, B.; Yan, X.; Richards, S. (2010). Investigating effects of asphalt pavement conditions on traffic accidents in tennessee based on the pavement management system (PMS). *J. Adv. Transp.*, 44, 150–161. [CrossRef].
- Chang, F., Chen, C. J., & Lu, C. J. (2004). A linear-time component-labeling algorithm using contour tracing technique. *computer vision and image understanding*, 93(2), 206-220.
- Chang-Huang, C. (2002). Edge detection based on class ratio. 152, sec.3, Peishen Rd., Shenkeng, Taipei, 22202, Taiwan, R.O.C.
- Choudhary, A., & Thakur, R. (1994). Connected component labeling on coarse grain parallel computers: an experimental study. *Journal of Parallel and Distributed Computing*, 20(1), 78-83.
- Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011, June). Flexible, high performance convolutional neural networks for image classification. In Twenty-second international joint conference on artificial intelligence.
- Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L. & Schmidhuber, J. (2011), Flexible, high performance convolutional neural networks for image classification, in *Proceedings of International Joint Conference on Artificial Intelligence, Barcelona, Spain*, pp. 1234–42.
- Coenen, T.B.J.; Golroo, A. (2017). A review on automated pavement distress detection methods. *Cogent Eng.*, 4, 1374822. [CrossRef]
- Dillencourt, M. B., Samet, H., & Tamminen, M. (1992). A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM (JACM)*, 39(2), 253-280.
- Eom, H., & Choi, H. (2018). Alpha-pooling for convolutional neural networks. arXiv preprint arXiv:1811.03436.
- European Commission. EC Policy Orientations on Road Safety 2011–2020. 2010. Available online: <https://goo.gl/ndXFPV> (accessed on 2 June 2018).
- Feng, Chen and Liu, Ming-Yu and Kao, Chieh-Chi and Lee, Teng-Yok. (2017), Deep Active Learning for Civil Infrastructure Defect Detection and Classification. 298-306. 10.1061/9780784480823.036.
- Folorunso O., Vincent, O. R., & Dansu, B. M. (2007). Image edge detection: A knowledge management technique for visual scene analysis. *Information Management and Computer Security*, 15(1), 23-32.
- Gargantini, I., & Tabakman, Z. (1982). Separation of connected component using linear quad-and oct-trees. In *Proc. 12th Conf. Numerical Mathematics and Computation (Vol. 37, pp. 257-276)*. University of Manitoba Winnipeg.



- Gestion de L'entretien des Chaussées (GEC). (2003). Relevé d'état et Appréciation Valeur D'indice. The Swiss Association of Road and Transportation Experts (VSS) SN-640925B.
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing*. 2nd edn Prentice Hall. New Jersey, 567-612.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning-An MIT Press book*.
- 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.
- Gotoh, T., Ohta, Y., Yoshida, M., & Shirai, Y. (1987, October). Component labeling algorithm for video rate processing. In *Advances in Image Processing* (Vol. 804, pp. 217-224). SPIE. M. Komeichi, Y. Ohta, T. Gotoh, T. Mima, M. Yoshida, Video-rate labeling processor, in: *Proceedings of the SPIE, Image Processing II*, vol. 1027, September 1988, pp. 69-76.
- Guthe, S., & Strasser, W. (2004). Advanced techniques for high-quality multi-resolution volume rendering. *Computers & Graphics*, 28(1), 25-78.
- Haralick, R. M. (1981). Some neighborhood operators. In *Real-Time Parallel Computing* (pp. 11-35). Springer, Boston, MA.
- Haralick, R. M., & Shapiro, L. G. (1992). *Computer and robot vision* (Vol. 1, pp. 28-48). Reading: Addison-wesley.
- Hashizume, A., Suzuki, R., Yokouchi, H., Horiuchi, H., & Yamamoto, S. (1990). An algorithm of automated RBC classification and its evaluation. *Bio Medical Engineering*, 28(1), 25-32.
- He, L., Chao, Y., & Suzuki, K. (2007, September). A linear-time two-scan labeling algorithm. In *2007 IEEE International Conference on Image Processing* (Vol. 5, pp. V-241). IEEE.
- He, L., Chao, Y., & Suzuki, K. (2008). A run-based two-scan labeling algorithm. *IEEE transactions on image processing*, 17(5), 749-756.
- Hecquard, J., & Acharya, R. (1991). Connected component labeling with linear octree. *Pattern recognition*, 24(6), 515-531.
- Hirschberg, D. S., Chandra, A. K., & Sarwate, D. V. (1979). Computing connected components on parallel computers. *Communications of the ACM*, 22(8), 461-464.
- Hu, Q., Qian, G., & Nowinski, W. L. (2005). Fast connected-component labelling in three-dimensional binary images based on iterative recursion. *Computer Vision and Image Understanding*, 99(3), 414-434.
- Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456).

- Istruzioni Sulla Pianificazione Della Manutenzione Stradale (Instruction for Road Maintenance Planning) (1988). CNR BU N.125/88.
- Iyer, S., & Sinha, S. K. (2005). A robust approach for automatic detection and segmentation of cracks in underground pipeline images. *Image and Vision Computing*, 23(10), 921-933.
- Jackins, C. L., & Tanimoto, S. L. (1980). Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14(3), 249-270.
- Keren, D., Osadchy, M., & Gotsman, C. (2001). Antifaces: A novel, fast method for image detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(7), 747-761.
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, 1097-1105.
- Laboratoire Central des Ponts et Chaussées. (1998). *Catalogue des Dégradations de Surface des Chaussées*; LCPC: Paris, France.
- LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9-48). Springer, Berlin, Heidelberg.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015), Deep learning, *Nature*, 521(7553), 436-44.
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436-444.
- Loprencipe, G.; Pantuso, A. (2017). A specified procedure for distress identification and assessment for urban road surfaces based on pci. *Coatings*, 7, 65. [CrossRef].
- Loprencipe, G.; Pantuso, A.; Di Mascio, P. (2017). Sustainable pavement management system in urban areas considering the vehicle operating costs. *Sustainability*. 9, 453. [CrossRef]
- Lumia, R. (1983). A new three-dimensional connected components algorithm. *Computer Vision, Graphics, and Image Processing*, 23(2), 207-217.
- Lumia, R., Shapiro, L., & Zuniga, O. (1983). A new connected components algorithm for virtual memory computers. *Computer Vision, Graphics, and Image Processing*, 22(2), 287-300.
- 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.
- Manohar, M., & Ramapriyan, H. K. (1989). Connected component labeling of binary images on a mesh connected massively parallel processor. *Computer vision, graphics, and image processing*, 45(2), 133-149.
- Mario, D., & Maltoni, D. (1997). Direct gray-scale minutiae detection in fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1), 27-40.



- Michael, U. (2003). Mathematics properties of the JPEG 2000 wavelet filters. *IEEE Transactions on Image Processing*, 12(9), 080-1090.
- Milan, S., Vaclav, H., & Roger, B. (2002). *Image processing analysis and machine vision*. London: Chapman and Hall, 255-280.
- Miller, J.S.; Bellinger, W.Y. (2003). *Distress Identification Manual for the Long Term Pavement Performance Program*, 4th ed.; Federal Highway Administration: Washington, DC, USA.
- Nair, V. & Hinton, G. E. (2010), Rectified linear units improve restricted Boltzmann machines, in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Haifa, Israel, June 21–24, 807–14.
- Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1), 1-21.
- Naoi, S. (1995). High-speed labeling method using adaptive variable window size for character shape feature. In *IEEE Second Asian Conf on Computer Vision (ACCV'95)* (Vol. 1, pp. I-408).
- Nassimi, D., & Sahni, S. (1980). Finding connected components and connected ones on a mesh-connected parallel computer. *SIAM Journal on computing*, 9(4), 744-757.
- Nguyen, H. N., Kam, T. Y., & Cheng, P. Y. (2014). An automatic approach for accurate edge detection of concrete crack utilizing 2D geometric features of crack. *Journal of Signal Processing Systems*, 77(3), 221-240.
- Olariu, S., Schwing, J. L., & Zhang, J. (1993). Fast component labelling and convex hull computation on reconfigurable meshes. *Image and vision computing*, 11(7), 447-455.
- Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: An application to face detection. *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*.
- Priotr, P. (2004). Practical signal decomposition design based on Haar – Wavelet transformation. *Journal of Applied Computer Science*, 2(1), 61-82.
- Roberts, L. G. (1965). Machine perception of three-dimensional solids. In J. T. Tippett (Ed.), *Optical and electro-optical information processing* (Ch. 9, pp. 159-197). Cambridge, Massachusetts: MIT Press.
- Ronsen, C., & Denjiver, P. (1984). *Connected Components in Binary Images: The Detection Problem*. New York, USA: John Wiley & Sons. R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1992.
- Rosenfeld, A. (1970). Connectivity in digital pictures. *Journal of the ACM (JACM)*, 17(1), 146-160.
- Rosenfeld, A. (1982). *AC Kak, Digital Picture Processing*. Ac. Press, New York.
- Rosenfeld, A., & Pfaltz, J. L. (1966). Sequential operations in digital picture processing. *Journal of the ACM (JACM)*, 13(4), 471-494.

- Salman, M., Mathavan, S., Kamal, K., Rahman, M. (2013), Pavement crack detection using the gabor filter, in: Proceedings of 16th International IEEE Annual Conference on Intelligent Transportation Systems, , pp. 2039–2044.
- Samet, H. (1981). Connected component labeling using quadtrees. *Journal of the ACM (JACM)*, 28(3), 487-501.
- Samet, H., & Tamminen, M. (1985). Computing geometric properties of images represented by linear quadtrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2), 229-240.
- Samet, H., & Tamminen, M. (1986, June). An improved approach to connected component labeling of images. In *International Conference on Computer Vision And Pattern Recognition (Vol. 318, p. 312)*.
- Samet, H., & Tamminen, M. (1988). Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE transactions on pattern analysis and machine intelligence*, 10(4), 579-586.
- Shah, Y.U.; Jain, S.S.; Tiwari, D.; Jain, M.K. (2013). Development of overall pavement condition index for urban road network. *Procedia Soc. Behav. Sci.*, 104, 332–341. [CrossRef].
- Shahin, M.Y. (1994). *Pavement Management for Airports, Roads, and Parking Lots*; Chapman & Hall: New York, NY, USA.
- Shirai, Y. (1987). Labeling connected regions, in: *Three-Dimensional Computer Vision*, Springer, Berlin, pp. 86–89.
- Simard, P. Y., Steinkraus, D. & Platt, J. C. (2003), Best practices for convolutional neural networks applied to visual document analysis, in *Proceedings of the 7th International Conference on Document Analysis and Recognition*, Edinburgh, UK, August 3–6, 958–62.
- Soukup, D. & Huber-Mö rk, R. (2014), Convolutional neural networks for steel surface defect detection from photometric stereo images, in *Proceedings of 10th International Symposium on Visual Computing*, Las Vegas, NV, 668–7.
- Sparr, G. (2002). *Image processing and pattern classification for character recognition*. Center for Mathematical Sciences, Lund University, 2, 25-78.
- Srihari, S.N. (1980). Hierarchical representations for serial section images, in: *Proceedings of the 5th International Conference on Pattern Recognition*, pp. 1075–1080.
- 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.
- Steinkrau, D., Simard, P. Y. & Buck, I. (2005), Using GPUs for machine learning algorithms, in *Proceedings of 8th International Conference on Document Analysis and Recognition*, Seoul, Korea, August 29–September 1, 1115–19.
- Suzuki, K., Armato III, S. G., Li, F., Sone, S., & Doi, K. (2003). Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection



- of lung nodules in low-dose computed tomography. *Medical physics*, 30(7), 1602-1617.
- Suzuki, K., Horiba, I., & Sugie, N. (2003). Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1), 1-23.
- Suzuki, K., Li, F., Sone, S., & Doi, K. (2005). Computer-aided diagnostic scheme for distinction between benign and malignant nodules in thoracic low-dose CT by use of massive training artificial neural network. *IEEE Transactions on Medical Imaging*, 24(9), 1138-1150.
- Suzuki, K., Yoshida, H., Näppi, J., Armato III, S. G., & Dachman, A. H. (2008). Mixture of expert 3D massive-training ANNs for reduction of multiple types of false positives in CAD for detection of polyps in CT colonography. *Medical physics*, 35(2), 694-703.
- Tamminen, M., & Samet, H. (1984). Efficient octree conversion by connectivity labeling. *ACM SIGGRAPH Computer Graphics*, 18(3), 43-51.
- Tech Transportation Institute; PMS Pavement Management Services Ltd.; Ireland Department of Transport Tourism & Sport. Development and implementation of a simplified system for assessing the condition of Irish regional and local roads (18–21 May, 2015). In *Proceedings of the 9th International Conference on Managing Pavement Assets*, Virginia Tech Transportation Institute (VTTI), Alexandria, VA, USA
- Tighe, S.; Li, N.; Falls, L.; Haas, R. (2000). Incorporating road safety into pavement management. *Transp. Res. Rec.*, 1699, 1–10. [CrossRef]
- Tucker, L. W. (1986). Labeling connected components on a massively parallel tree machine. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (pp. 124-129).
- Vincent, O. R., Folorunso, O., & Agboola, A. A. A. (2006). Modified algorithm for transform-based image compression. *Journal of Computer Science and its Application*, 13(1), 43-48.
- Vincent, O. R., Folorunso, O., & Agboola, A. A. A. (2006). Modified algorithm for transform-based image compression. *Journal of Computer Science and its Application*, 13(1), 43-48.
- Wang, P., & Huang, H. (2010, October). Comparison analysis on present image-based crack detection methods in concrete structures. In *2010 3rd international congress on image and signal processing* (Vol. 5, pp. 2530-2533). IEEE.
- Witkin, A. P. (1987). Scale-space filtering. In *Readings in Computer Vision* (pp. 329-332). Morgan Kaufmann.
- Wu, K., Otoo, E., & Suzuki, K. (2009). Optimizing two-pass connected-component labeling algorithms. *Pattern Analysis and Applications*, 12(2), 117-135.
- Y. Schiloach, U. Vishkin. (1982). An  $o(\log n)$  parallel connectivity algorithm, *J. Algorithms* 3, 57–67.



- Yang, Y. S., Yang, C. M., & Huang, C. W. (2015). Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis. *Advances in Engineering Software*, 83, 99-108.
- Yuval, F. (1996). *Fractal image compression (theory and application)*. Institute for non-linear Science, University of California, San Diego, USA.
- Zhang, J., Qiu, H., Shamsabadi, S., S.; Birken, R.; Schirner, G. (2014). Wip abstract: System-level integration of mobile multi-moda multi-sensor systems. In *2014 ACM/IEEE 22d International Conference on Cyber-Physical Systems, IICPS 2014*, 227.



## **APPENDIX A-1: FATIGUE CRACKS**



Sample No	Typical Sample Images of Fatigue Cracks
1	 A photograph of a concrete surface showing several dark, irregular cracks. A prominent crack runs diagonally from the top left towards the center. Another crack runs horizontally across the middle. A circular white mark is visible on the right side of the image.
2	 A photograph of a concrete surface showing several dark, irregular cracks. A prominent crack runs horizontally across the top. Another crack runs diagonally from the top left towards the center. A circular white mark is visible in the center of the image.







Sample No	Typical Sample Images of Fatigue Cracks
3	 A photograph of a concrete surface showing a network of dark, irregular fatigue cracks. The cracks are most prominent in the center and right side, forming a complex, interconnected pattern. The concrete is a light gray color with some darker spots.
4	 A photograph of a concrete surface showing a network of dark, irregular fatigue cracks. The cracks are most prominent in the center and right side, forming a complex, interconnected pattern. The concrete is a light gray color with some darker spots.


Sample No	Typical Sample Images of Fatigue Cracks
5	
6	

Sample No	Typical Sample Images of Fatigue Cracks
7	 A photograph showing a network of fatigue cracks on a concrete surface. The cracks are dark and form a complex, interconnected pattern that generally follows a diagonal path from the upper left towards the lower right. The surrounding concrete is a light gray color.
8	 A photograph showing a network of fatigue cracks on a concrete surface, similar to sample 7. The cracks are dark and form a complex, interconnected pattern that generally follows a diagonal path from the upper left towards the lower right. The surrounding concrete is a light gray color.

Sample No	Typical Sample Images of Fatigue Cracks
9	
10	



Sample No	Typical Sample Images of Fatigue Cracks
11	 A grayscale microscopic image of a concrete surface. The surface is covered with a complex network of fine, interconnected cracks. The cracks vary in orientation, with some running roughly parallel to each other and others branching off at various angles. The overall appearance is that of a highly stressed and fractured material.
12	 A grayscale microscopic image of a concrete surface. A single, prominent, and relatively straight crack runs vertically through the center of the frame. There are also several smaller, fainter cracks scattered around the main crack. The surface texture is granular and typical of concrete.



Sample No	Typical Sample Images of Fatigue Cracks
13	
14	


Sample No	Typical Sample Images of Fatigue Cracks
15	

## **APPENDIX A-2: BLOCK CRACKING**



Sample No	Typical Sample Images of Block Cracking	
1		
2		


Sample No	Typical Sample Images of Block Cracking
3	 A close-up photograph of asphalt pavement showing several irregular, interconnected cracks that form a network of polygonal blocks. The cracks are dark and recessed into the surface.
4	 A close-up photograph of asphalt pavement showing several irregular, interconnected cracks that form a network of polygonal blocks. The cracks are dark and recessed into the surface.



Sample No	Typical Sample Images of Block Cracking
5	 <p data-bbox="898 1048 938 1077">(e)</p>



## **APPENDIX A-3: CORRUGATION AND SHOVING**

Table A-3.1:


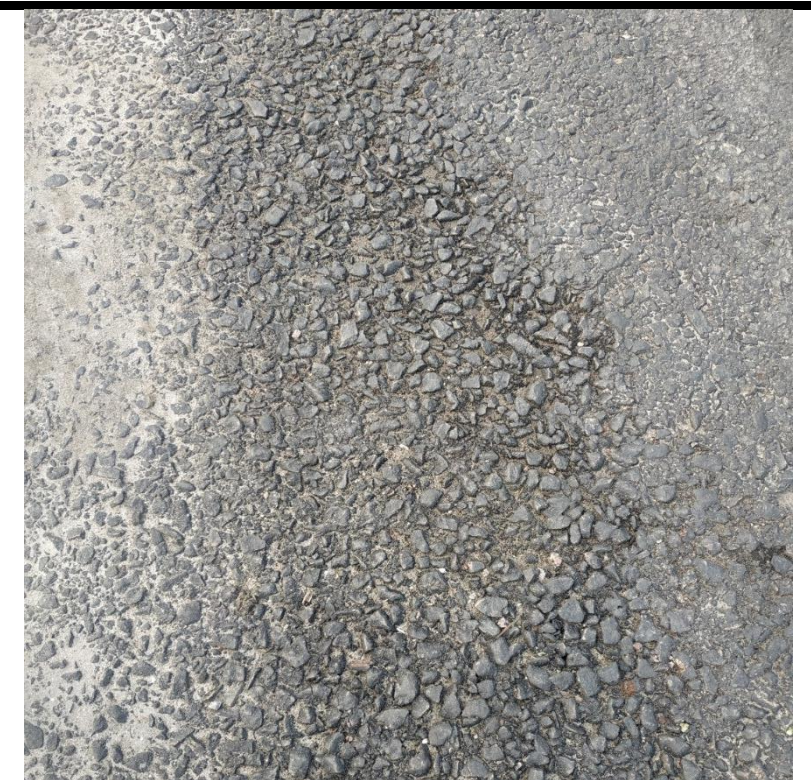
Sample No	Typical Sample Images of Corrugation and Shoving
1	
2	

Sample No	Typical Sample Images of Corrugation and Shoving		
3		 A photograph showing a close-up view of a road surface. The surface is a light brown, granular material, likely asphalt or concrete. It exhibits a distinct longitudinal corrugation or wavy pattern, with a series of parallel ridges and grooves running vertically through the center of the frame.	
4		 A photograph showing a close-up view of a road surface. The surface is a dark gray, granular material, likely asphalt or concrete. It exhibits a significant shoving or pothole defect, characterized by a large, irregularly shaped area of displaced material. The displaced material is lighter in color and appears to be a different aggregate or has been pushed up from below. The surrounding surface shows some minor cracking and wear.	



Sample No	Typical Sample Images of Corrugation and Shoving
5	
6	


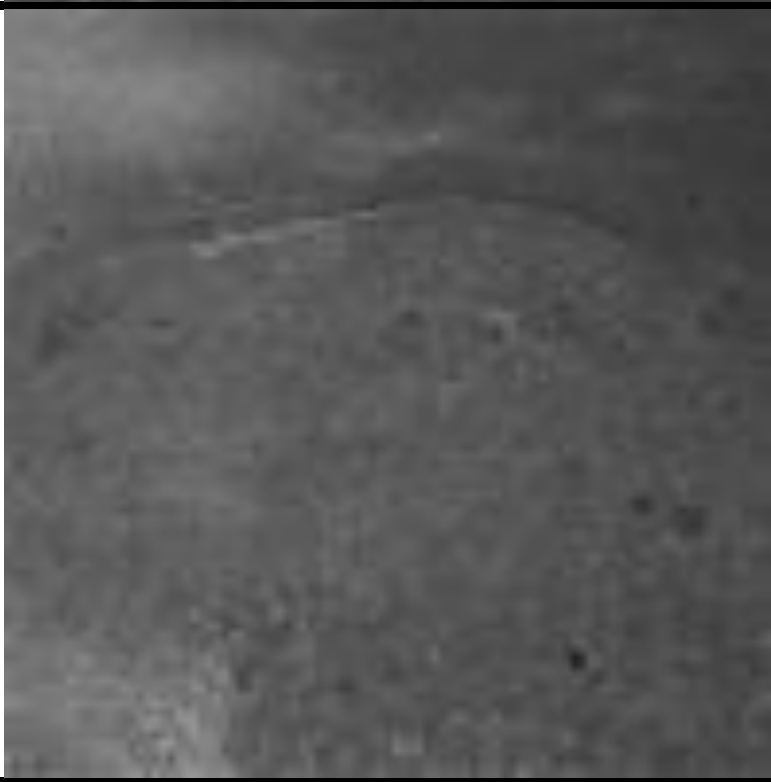
Sample No	Typical Sample Images of Corrugation and Shoving	
7		 A photograph showing a close-up view of an asphalt surface. The surface exhibits significant corrugation and shoving, with a rough, uneven texture. The material appears dark gray to black, with some lighter, sandy areas interspersed. The corrugations are irregular and somewhat parallel to each other, suggesting a directional force or traffic pattern.
8		 A photograph showing a close-up view of an asphalt surface, similar to sample 7 but with more pronounced corrugation and shoving. The surface is highly textured and uneven, with a dark gray to black color. The corrugations are more irregular and pronounced, with some areas appearing more severely damaged or displaced. The overall appearance is one of significant surface distress.



Sample No	Typical Sample Images of Corrugation and Shoving		
9		 A close-up photograph of a road surface showing significant corrugation and shoving. The surface is uneven, with a dense layer of dark, angular aggregate particles. The texture is highly irregular, with many small voids and protrusions, indicating a loss of structural integrity.	
10		 A close-up photograph of a road surface showing significant corrugation and shoving. The surface is uneven, with a dense layer of dark, angular aggregate particles. The texture is highly irregular, with many small voids and protrusions, indicating a loss of structural integrity.	

## **APPENDIX A-4: DEPRESSION**

Sample No	Typical Sample Images of Depression cracks	
1		 A grayscale microscopic image of a concrete surface. A prominent, dark, slightly curved depression crack runs diagonally from the upper left towards the center. The surface texture is granular and uneven.
2		 A grayscale microscopic image of a concrete surface, similar to the first sample. It shows a network of fine, dark depression cracks. One main crack runs horizontally across the middle, with several smaller, branching cracks extending from it.

Sample No	Typical Sample Images of Depression cracks		
3			
4			

Sample No	Typical Sample Images of Depression cracks	
5		



## **APPENDIX A-5: JOINT REFLECTION CRACKING**

Sample No	Typical Sample Images of Joint Reflection Cracking
1	 A photograph showing a concrete surface with a prominent joint reflection crack. The crack runs diagonally from the upper left towards the lower right. The surface is wet and shows some discoloration and aggregate exposure.
2	 A close-up photograph of a concrete surface showing joint reflection cracking. The crack is more pronounced and runs diagonally. There is some debris, including a small leaf, on the surface.



Sample No	Typical Sample Images of Joint Reflection Cracking
3	
4	







Sample No	Typical Sample Images of Joint Reflection Cracking
5	
6	



Sample No	Typical Sample Images of Joint Reflection Cracking
7	 A photograph showing a concrete surface with a prominent horizontal joint reflection crack. The crack runs across the middle of the frame. The concrete is light gray with some reddish-brown spots. There is a small white object and a purple object on the surface above the crack. A black cable is visible on the left side.
8	 A close-up photograph of a concrete surface showing joint reflection cracking. The crack is a jagged, irregular line. There is a small pool of brownish water in the center of the crack. The concrete is light gray with many reddish-brown spots. The background is a darker, more textured surface.



## **APPENDIX A-6: LONGITUDINAL CRACKING**

Sample No	Typical Sample Images of Longitudinal Cracking	
1		
2		

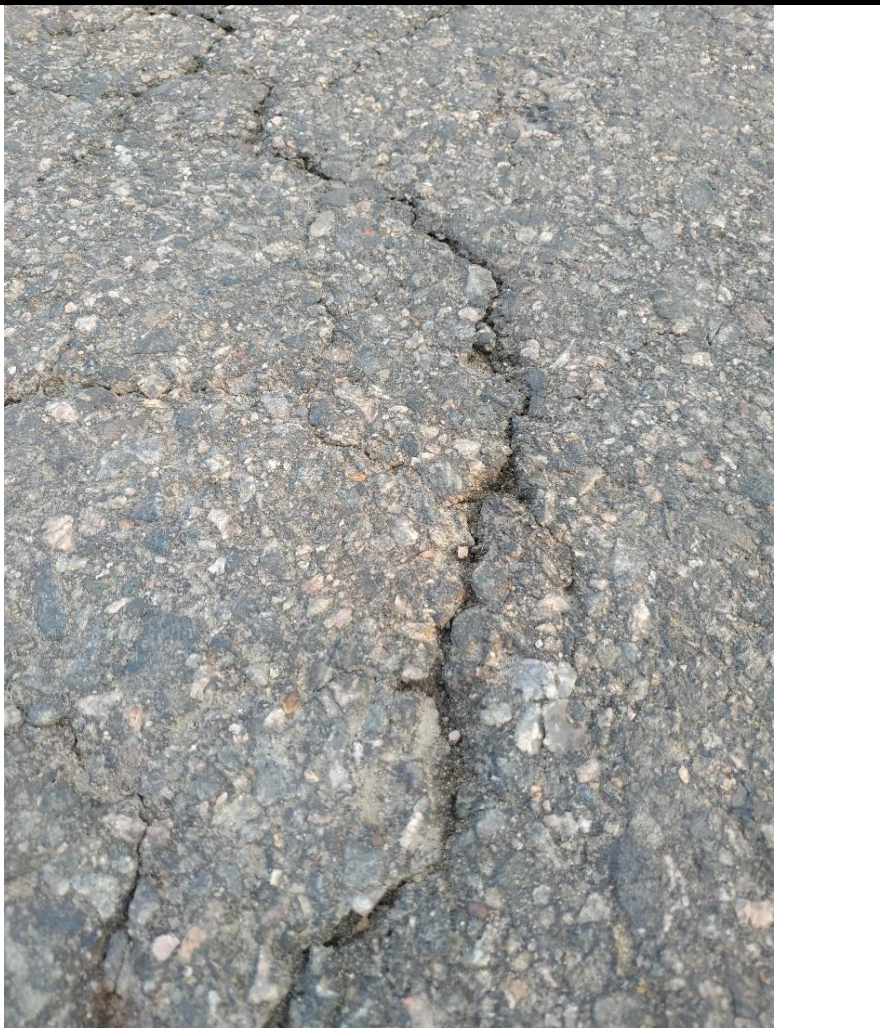

Sample No	Typical Sample Images of Longitudinal Cracking
3	 A photograph of a concrete pavement surface. A prominent, dark, irregular longitudinal crack runs across the middle of the frame. The crack is filled with a dark, possibly oily or water-soaked material. The surrounding concrete is light gray and shows some minor surface wear and small dark spots.
4	 A photograph of an asphalt pavement surface. A longitudinal crack runs horizontally across the middle of the frame. To the left of the crack, there is a distinct, lighter-colored, irregular patch, likely a repair or a different material. The asphalt is dark gray and shows some texture and small aggregate particles.




Sample No	Typical Sample Images of Longitudinal Cracking		
5		 A photograph showing a close-up view of a concrete surface. A prominent, irregular longitudinal crack runs vertically through the center of the frame. The crack is filled with a dark, possibly oily or dirt-filled material. The surrounding concrete has a light gray, textured appearance.	
6		 A photograph showing a close-up view of a concrete surface. A prominent, irregular longitudinal crack runs vertically through the center of the frame. The crack is filled with a dark, possibly oily or dirt-filled material. The surrounding concrete has a light gray, textured appearance.	




Sample No	Typical Sample Images of Longitudinal Cracking
7	 A photograph of a concrete surface showing several longitudinal cracks. The cracks are dark, irregular lines that run roughly parallel to each other across the width of the image. The concrete has a light gray, textured appearance.
8	 A photograph of a concrete surface showing longitudinal cracking. The cracks are dark, irregular lines that run roughly parallel to each other across the width of the image. The concrete has a light gray, textured appearance.




Sample No	Typical Sample Images of Longitudinal Cracking	
9		 A photograph showing a close-up of a smooth, light gray concrete surface. A single, irregular longitudinal crack runs diagonally from the bottom left towards the top right. The crack is filled with a dark, reddish-brown material, likely a repair or sealant.
10		 A photograph showing a close-up of a rough, aggregate concrete surface. The surface is composed of various sized stones and a dark gray matrix. Multiple longitudinal cracks are visible, running vertically and slightly diagonally. The cracks are filled with a dark, reddish-brown material, similar to the one in sample 9.




Sample No	Typical Sample Images of Longitudinal Cracking	
11		
12		



Sample No	Typical Sample Images of Longitudinal Cracking	
13		
14		
15		



Sample No	Typical Sample Images of Longitudinal Cracking	
16		
17		
18		

Sample No	Typical Sample Images of Longitudinal Cracking	
19		
20		
21		



## **APPENDIX A-7: PATCHING**

Sample No	Typical Sample Images of Patching	
1		
2		



Sample No	Typical Sample Images of Patching
3	
4	



Sample No	Typical Sample Images of Patching
5	 A photograph of a concrete surface showing several irregular, blue-tinted patches. The patches vary in size and are scattered across the light-colored concrete. The surface appears slightly textured and possibly weathered.
6	 A photograph of a concrete surface showing a large, dark, irregularly shaped area. This area contains several parallel, dark, textured lines that resemble tire tracks or a similar surface treatment. The surrounding concrete is a uniform light gray color.







Sample No	Typical Sample Images of Patching		
7		 A photograph showing a close-up of a concrete surface. A diagonal crack runs from the top left towards the bottom right. A patching area is visible, appearing as a slightly different texture and color compared to the surrounding concrete.	
8		 A photograph showing a close-up of a concrete surface. The surface is highly textured and appears to be a patching area, with many small, dark, irregular fragments and a rough, uneven appearance.	



**APPENDIX A-8: POLISHED AGGREGATE**


Sample No.	Typical Sample images of Polished Aggregate		
1		 A photograph showing a close-up view of a polished aggregate surface. The surface is composed of a light-colored, fine-grained matrix with numerous small, dark, angular aggregate particles embedded throughout. The overall appearance is smooth and uniform.	
2		 A photograph showing a close-up view of a polished aggregate surface. The surface is composed of a dark, fine-grained matrix with numerous small, light-colored, angular aggregate particles embedded throughout. The overall appearance is smooth and uniform.	

Sample No.	Typical Sample images of Polished Aggregate
3	 A photograph showing a close-up view of a polished aggregate surface. The surface is composed of small, dark-colored aggregate particles (likely crushed stone or gravel) embedded in a dark matrix. A prominent white line runs diagonally across the upper portion of the image. There are also some red markings or stains on the surface, particularly near the bottom left corner.
4	 A photograph showing a close-up view of a polished aggregate surface. The surface is composed of small, dark-colored aggregate particles (likely crushed stone or gravel) embedded in a dark matrix. A blue and white label is visible in the bottom left corner, partially obscured by the aggregate. The label appears to be a technical specification or test result label.



Sample No.	Typical Sample images of Polished Aggregate
5	
6	

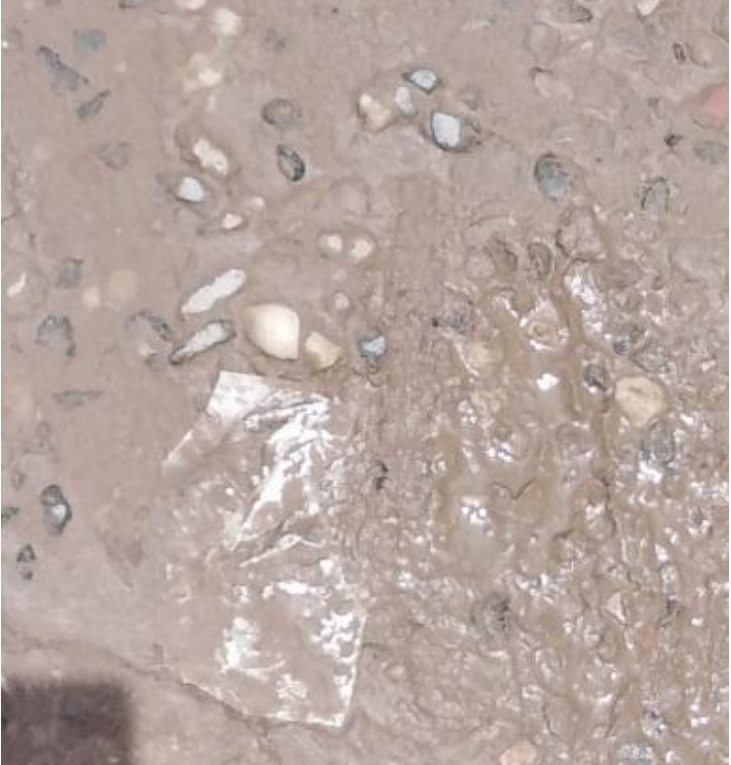
Sample No.	Typical Sample images of Polished Aggregate
7	
8	

Sample No.	Typical Sample images of Polished Aggregate		
9			
10			



Sample No.	Typical Sample images of Polished Aggregate		
11			
12			







Sample No.	Typical Sample images of Polished Aggregate	
13		
14		




Sample No.	Typical Sample images of Polished Aggregate
15	 A microscopic image of a polished aggregate sample. The matrix is a light brown, sandy material. Scattered throughout are various aggregate particles, including small, dark grey, angular particles and larger, more rounded, light-colored particles. A prominent, large, irregularly shaped, light-colored particle is visible in the lower-left quadrant of the image.



## **APPENDIX A-9: POTHOLES**

Sample No	Typical Sample Images of Potholes
1	
2	




Sample No	Typical Sample Images of Potholes	
3		
3		




Sample No	Typical Sample Images of Potholes	
4		
5		




Sample No	Typical Sample Images of Potholes
6	
7	
8	




Sample No	Typical Sample Images of Potholes		
9			
10			









Sample No	Typical Sample Images of Potholes		
11			
12			
13			



Sample No	Typical Sample Images of Potholes	
14		
15		
16		



Sample No	Typical Sample Images of Potholes	
17		 A close-up photograph of a gravel surface. The gravel consists of small, dark grey stones of various sizes. A small, shallow pothole is visible in the center, containing some loose gravel and a small piece of dried leaf.
18		 A close-up photograph of a gravel surface. The gravel consists of small, dark grey stones of various sizes. A small, shallow pothole is visible in the center, containing some loose gravel and a small piece of dried leaf.
19		 A close-up photograph of a gravel surface. The gravel consists of small, dark grey stones of various sizes. A small, shallow pothole is visible in the center, containing some loose gravel and a small piece of dried leaf.

Sample No	Typical Sample Images of Potholes		
20			
21			
22			




Sample No	Typical Sample Images of Potholes	
23		
24		
25		




Sample No	Typical Sample Images of Potholes		
26			
27			
28			



Sample No	Typical Sample Images of Potholes	
29		 A photograph showing a pothole in a gravelly surface. The pothole is a shallow, irregular depression filled with dark brown soil and small, light-colored pebbles. The surrounding surface is composed of a mix of grey and white gravel and dark brown soil.
30		 A photograph showing a pothole in a gravelly surface, similar to sample 29. The pothole is a shallow, irregular depression filled with dark brown soil and small, light-colored pebbles. The surrounding surface is composed of a mix of grey and white gravel and dark brown soil.


Sample No	Typical Sample Images of Potholes		
31			
32			








Sample No	Typical Sample Images of Potholes
33	
34	
35	




Sample No	Typical Sample Images of Potholes	
36		
37		
38		



Sample No	Typical Sample Images of Potholes
39	
40	

Sample No	Typical Sample Images of Potholes
41	




Sample No	Typical Sample Images of Potholes	
42		
43		




Sample No	Typical Sample Images of Potholes	
44		
45		
46		




Sample No	Typical Sample Images of Potholes		
47			
48			
49			




Sample No	Typical Sample Images of Potholes
50	
51	









Sample No	Typical Sample Images of Potholes	
52		
53		
54		



Sample No	Typical Sample Images of Potholes	
55		
56		
57		



Sample No	Typical Sample Images of Potholes	
58		
59		
60		

Sample No	Typical Sample Images of Potholes	
61		
62		
53		



Sample No	Typical Sample Images of Potholes	
64		
65		
66		



Sample No	Typical Sample Images of Potholes	
67		
68		
69		




Sample No	Typical Sample Images of Potholes		
70			
71			

Sample No	Typical Sample Images of Potholes	
72		
73		





Sample No	Typical Sample Images of Potholes	
74		 A close-up photograph showing a surface composed of dark grey gravel and sand. The gravel consists of small, irregularly shaped stones, and the sand is a light brown color. The surface appears slightly uneven and textured.
75		 A photograph of a pothole in a paved surface. The pothole is filled with dark brown, wet mud. The surrounding pavement is a light grey color. The edges of the pothole are irregular and show some signs of wear and debris.



Sample No	Typical Sample Images of Potholes		
76		 A photograph showing a pothole on a road surface. The pothole is filled with a layer of brown, muddy water. The surrounding road surface is composed of grey gravel and sand. The pothole itself is irregular in shape and appears to be a significant depression in the pavement.	
77		 A photograph showing a pothole on a road surface. The pothole is filled with a layer of brown, muddy water. The surrounding road surface is composed of grey gravel and sand. The pothole itself is irregular in shape and appears to be a significant depression in the pavement.	

Sample No	Typical Sample Images of Potholes		
78			
79			
80			

## **APPENDIX A-10: RUTTING**

Sample No.	Typical Sample Images of Rutting	
1		 A close-up photograph of asphalt pavement showing a longitudinal rut. The rut is a deep, narrow groove that runs parallel to the direction of traffic. The surrounding pavement surface is relatively smooth and shows some minor texture.
2		 A photograph of asphalt pavement showing a longitudinal rut. The rut is a deep, narrow groove that runs parallel to the direction of traffic. A white painted line is visible on the right side of the rut. The surrounding pavement surface is relatively smooth and shows some minor texture.

Sample No.	Typical Sample Images of Rutting
3	 A photograph showing a close-up view of asphalt pavement. The surface is dark grey and shows signs of rutting, with a distinct longitudinal groove or depression running through the center of the frame. The texture of the asphalt is visible, showing small aggregate particles.
4	 A photograph showing a close-up view of asphalt pavement. The surface is dark grey and shows signs of rutting, with a distinct longitudinal groove or depression running through the center of the frame. The texture of the asphalt is visible, showing small aggregate particles.



Sample No.	Typical Sample Images of Rutting
5	 A photograph of asphalt pavement showing a distinct longitudinal rut. The rut is a deep, narrow groove that runs parallel to the direction of traffic. The surrounding pavement surface is relatively smooth but shows some minor texture and small debris.
6	 A photograph of asphalt pavement showing a longitudinal rut, similar to sample 5. The rut is clearly visible as a dark, recessed channel in the pavement. The overall appearance is that of a well-used road surface with some rutting.



Sample No.	Typical Sample Images of Rutting		
7		 A close-up photograph of asphalt pavement showing a distinct longitudinal rut. The rut is a shallow groove that runs parallel to the direction of traffic, with a slightly darker, more compacted surface within it. The surrounding pavement has a standard aggregate texture.	
8		 A close-up photograph of asphalt pavement showing a longitudinal rut. The rut is a shallow groove with a slightly darker, more compacted surface. The surrounding pavement has a standard aggregate texture.	



Sample No.	Typical Sample Images of Rutting		
9		 A close-up photograph of asphalt pavement showing a distinct longitudinal rut. The rut is a deep, narrow groove that runs parallel to the direction of traffic. The surface texture is granular and dark grey.	
10		 A photograph of asphalt pavement showing a longitudinal rut. The rut is a deep, narrow groove that runs parallel to the direction of traffic. The surface texture is granular and dark grey.	

Sample No.	Typical Sample Images of Rutting		
11		 A close-up photograph of asphalt pavement showing a distinct rutting defect. The rut is a longitudinal groove that has formed in the surface, with a slightly raised edge on one side. The texture of the asphalt is visible, and the rutting is a result of traffic loading over time.	
12		 A close-up photograph of asphalt pavement showing a rutting defect. The rut is a longitudinal groove that has formed in the surface, with a slightly raised edge on one side. The texture of the asphalt is visible, and the rutting is a result of traffic loading over time.	

Sample No.	Typical Sample Images of Rutting
13	 A close-up photograph of an asphalt road surface. A prominent, deep rut is visible, running diagonally from the bottom left towards the top right. The rut is filled with a darker, smoother material, likely a repair or a different layer of asphalt. A white painted line is visible on the right side of the road, parallel to the rut.
14	 A close-up photograph of an asphalt road surface, very similar to sample 13. It shows a deep rut in the asphalt, with a white painted line on the right side. The rut is filled with a darker material. There is a faint "© 2022 Google" watermark in the center of the image and a "Google" logo at the bottom center.

Sample No.	Typical Sample Images of Rutting	
15		 A photograph showing a close-up view of a road surface. The asphalt is dark and shows signs of wear, including a distinct longitudinal rut or groove running down the center of the frame. The texture of the asphalt is visible, and there is a white dashed line on the right side of the image.
16		 A photograph showing a close-up view of a road surface. The asphalt is dark and shows signs of wear, including a distinct longitudinal rut or groove running down the center of the frame. The texture of the asphalt is visible, and there is a white dashed line on the left side of the image.

Sample No.	Typical Sample Images of Rutting
17	
18	

<b>Sample No.</b>	<b>Typical Sample Images of Rutting</b>
19	