

Imaging-Based Identification of International Roughness Index Using Deep Neural Networks

by

Jiangyu Zeng

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Structural Engineering

Department of Civil and Environmental Engineering
University of Alberta

© Jiangyu Zeng, 2022

Abstract

The International Roughness Index (IRI) is one of the most critical indicators in the field of pavement performance management. Since the introduction of Deep Neural Networks (DNNs), many researchers have created different kinds of DNN models to forecast the IRI. However, none of them employed pavement photos as the direct inputs when training their models, regardless of the substantial development of DNNs in image processing. On the other hand, thanks to the fast products in photography equipment, small and convenient sports action cameras, such as the GoPro Hero series, are able to capture smooth videos at a high framerate with built-in electronic image stabilization systems. Although there exist DNN-based models to detect, classify, and quantify distresses as well as models based on neural networks to identify the IRI using distresses, there is currently no model that directly uses pavement images as the input to estimate the IRI. Therefore, this thesis can be considered as the first attempt to use imaging-based DNN models to estimate IRI. Besides the development of the DNN, the procedures of collecting IRI ground truths and stitching a full-size image of an entire 50-m long pavement segment are also demonstrated. After analyzing the testing results, it is shown that imaging-based DNN models can be a promising method to identify the IRI. In the discussion section, several controlled experiments were performed to learn how the related hyperparameters and data combinations affect the performance. Additionally, the proposed DNN model was compared with other models, and the results indicated that the current model still has room for improvement. Therefore, at the end of this thesis, recommendations and future works are also addressed.

Acknowledgements

I would firstly express my deepest gratitude to my two supervisors, Dr. Mustafa Gül and Dr. Qipei Mei. Without their guidance and support, this thesis could never have been completed. They helped me with my research, taught me how to organize papers and write articles in technical languages. They are the best supervisors in the world, and I would like to continue my Ph.D. education under their supervision.

I would also like to thank my friend Mr. Rahmat Ali, who was a collaborator during my undergraduate study at the University of Manitoba. With his encouragement, I started my journey to do academic research and write technical papers. It is also him who encouraged me and helped me apply for higher education at the University of Alberta.

At last, I wish to thank my parents and my sister for their endless support, especially during the most challenging time of the COVID pandemic.

Table of Contents

1. CHAPTER 1: INTRODUCTION.....	1
1.1. The Structural Condition of Pavements.....	1
1.2. Current Practices for Pavement Assessment.....	2
1.2.1. Pavement Management System.....	2
1.2.2. Pavement Condition Indices.....	2
1.3. Introduction to IRI Measurement.....	4
1.3.1. State of the Practice.....	4
1.3.2. State of the Art.....	7
1.4. Problem Statement and Crowdsensing-Based Infrastructure Monitoring.....	8
1.5. DNN and Transfer Learning.....	9
1.6. The Assumption in This Study.....	10
1.7. Objectives and Scope.....	11
1.8. Organization of the Thesis.....	12
2. CHAPTER 2: LITERATURE REVIEW.....	13
2.1. Past-Based ANNs to Predict IRI.....	13
2.2. Present-Based ANNs to Predict IRI.....	15
2.3. Imaging-Based DNNs in Distress Detection.....	18
3. CHAPTER 3: DATA COLLECTION.....	20
3.1. IRI Ground Truths.....	20

3.2. Collection of Image Data	22
3.2.1. Camera Setup	22
3.2.2. Overview of Image Preparation Procedures	23
3.2.3. Perspective Transformation	24
3.2.4. Verification of the Proposed Perspective Transformation.....	28
3.2.5. Image Stitching	29
3.2.6. Verification of GoPro GPS Accuracy	34
4. CHAPTER 4: DNN MODEL AND TRANSFER LEARNING.....	37
4.1. The Invention of ResNet34.....	37
4.1.1. Overview Architecture of ResNet34.....	37
4.1.2. Convolutional Layers.....	38
4.1.3. Residual Learning	40
4.1.4. FC Layers.....	42
4.2. Modifications of ResNet34	43
4.2.1. Overview Architecture of the DNN after Modification.....	43
4.2.2. Mixed Pooling Layer	43
4.2.3. Linear Transformation	44
4.2.4. Loss Function.....	45
4.2.5. Adam Optimization Algorithm.....	46
4.2.6. Cyclical Learning Rate	48

4.3. Transfer Learning.....	49
5. CHAPTER 5: RESULTS AND DISCUSSIONS	50
5.1. Training and Testing Results	50
5.1.1. Separated Left and Right Wheel Path.....	51
5.1.2. Combining Left and Right Wheel Path Datasets	53
5.1.3. Intuitive Demonstration of Prediction Results.....	54
5.2. Discussions	55
5.2.1. Comparison Between Results of Different Dataset Combinations.....	55
5.2.2. Effects of δ in the Loss Function	56
5.2.3. Effects of Batch Sizes	58
5.2.4. Comparison With the Other Present-Based Models	59
6. CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS	61
6.1. Summary and Conclusions	61
6.2. Recommendations for Future Work.....	62
REFERENCES	64

List of Tables

Table 1.1: IRI trigger thresholds categorized by Annual Average Daily Traffic (AADT) [10]....	3
Table 1.2: Highway pavement condition rating criteria [11].....	4
Table 3.1. Sample of International Roughness Index Data [54].....	20
Table 3.2. Coordinates of the investigated road sections.....	22
Table 3.3. Sample of geographical data recorded by GoPro Hero 7 Black.	29
Table 3.4. Errors of GoPro GPS coordinates.....	35
Table 4.1. Summary of FC layers.	45
Table 5.1. Testing results.....	56
Table 5.2. Results of controlled experiments on δ	57
Table 5.3. Results of controlled experiments on batch size.....	58

List of Figures

Figure 1.1. PSP-7000 survey vehicle [12]	5
Figure 1.2. Quarter-car algorithm (adapted from [13]).....	5
Figure 3.1. Maps of the investigated road sections (modified from [56])......	21
Figure 3.2. GoPro Hero 7 Black mounted at the rear of BMW 328i xDrive.....	23
Figure 3.3. Schematic procedures of the imaging data preparation.....	24
Figure 3.4. Side view of the camera setup (modified from [58]).	25
Figure 3.5. Sample image.	26
Figure 3.6. Perspective transformation.	26
Figure 3.7. Side view in a Cartesian coordinate system.	27
Figure 3.8. Sample of calculated perspective transformation.....	28
Figure 3.9. Scene simulation of a highway lane.	28
Figure 3.10. Comparison between theoretical and measured AB length.....	29
Figure 3.11. Sample calculation of using linear interpolation method get the frame number.....	30
Figure 3.12. Stitching transformed frames to get a completed image for one segment.	32
Figure 3.13. Sample of how a transformed frame is cropped.....	33
Figure 3.14. A sample of the stitched full-size image composed of 200 frames (rotated 90°).....	34
Figure 3.15. Screen shot of the Movable Type Scripts (modified from [59]) [56]).	35
Figure 3.16. Configuration of the camera path in a highway lane (not to scale).....	36
Figure 4.1. Architecture of ResNet34 with the depth of each layer annotated.....	38
Figure 4.2. Compositions of convolutional layers in ResNet34.	39
Figure 4.3 Composition of the plain network.	40
Figure 4.4 Comparison of residual block and plain block.....	41

Figure 4.5. Linear projection of fc-1000.....	42
Figure 4.6. Architecture of the modified ResNet34 [60]......	43
Figure 4.7. Comparison of L2 loss with Huber’s loss in terms of $(IRI_{ground} - IRI_{pred})$	46
Figure 4.8. Loss vs learning rate curve.	49
Figure 5.1. IRI distributions.....	50
Figure 5.2. Training and validation loss of left wheel path and right wheel path separately.	51
Figure 5.3. Relationships between IRI ground truths and predictions of testing datasets.	53
Figure 5.4. Training and validation loss of combined dataset.	54
Figure 5.5. Relationships between ground truths and predictions of the combined testing subset.	54
Figure 5.6. Samples of prediction results.....	55
Figure 5.7. Results of controlled experiments on δ	57
Figure 5.8. Results of controlled experiments on batch size.	59

1. CHAPTER 1: INTRODUCTION

1.1. The Structural Condition of Pavements

Civil infrastructure is the key that binds every citizen closely with the society – from roads to railways, bridges to tunnels, water supply to electricity delivery. Each component plays an essential role in modern people’s daily life. Transportation infrastructure composes a critical part of the civil infrastructure system, ensuring the safe and efficient movement of passengers and freight. Canada, as the second-largest country in total area, has 1.13 million kilometres of roads [1], and road is the dominant transportation mode across the country in terms of both moving passengers and goods [2]. Like all the other infrastructures, road transportation systems require careful and frequent maintenance to keep them in good shapes and conditions. About 40% of the roads across the country are paved, which need more money and labour to monitor and rehabilitate than those unpaved [1].

For a typical pavement, there are three structural layers on top of the compacted/natural soil layer [3]. From bottom to top, these layers are namely subbase layer, base course layer, and surface course layer. The subbase layer is usually of lower quality than the other two layers, and its primary function is to support the base course layer, especially when the subgrade soils are in very poor condition. The base course layer is the layer on top of the subbase layer to support the surface course layer stably and uniformly. Finally, the surface course layer is the layer directly supporting the traffic load. Its function requires this layer can provide both tire friction and resistance to traffic abrasion. Besides that, this layer must be waterproof to protect the underlying layers because it is directly exposed under the climate.

In the Federal Highway Administration (FHWA)’s report [3], failure of pavements is defined as rutting and cracking that exceeds the anticipation during their service life, and failure is a relative term of distress. In their opinions, distress can be caused by 1) overloading that leads to shear failure in the structural layers, 2) surface fatigue and excessive settlement, 3) soils’ volume change in the subgrade layer because of climate/temperature change, and 4) improper drainage. These factors imply that pavement surface distress can reflect the structural performance to a significant extend. On the other hand, a study by Gong et al. [4] reveals that the International Roughness

Index (IRI) is substantially affected by distress. Therefore, IRI could be an essential index to quantitatively measure the distress and thus reflect the pavement's structural condition.

There are two main types of paved pavements based on structural performance, flexible pavements and rigid pavements. In general, flexible pavements have asphalt surface layers, while rigid pavements consist of concrete surface layers. The objectives in this thesis are about flexible asphalt pavements, so all the words “pavement(s)” mean asphalt pavement(s) in the following sections.

1.2. Current Practices for Pavement Assessment

1.2.1. Pavement Management System

Considering the vast pavement networks, a system approach is needed for more efficient management. Therefore, the Pavement Management System (PMS) is developed as a decision support tool for engineers or transportation agencies to manage the pavement networks better. Although PMS is now a comprehensive system that can not only forecast future pavement conditions but also perform economic analysis between alternative maintenance strategies, the initial purpose of PMS was only to help decision-makers evaluate potential pavement preservation treatments concerning current pavement conditions and traffics [5]. Therefore, one of the main tasks of PMS is to determine what type of preservation treatment is needed for a given pavement section. Thus, a criterion must be defined to classify between the candidate treatments. This criterion is known as the roughness of pavement surface, which is a critical factor that dominates the driving experience and transportation safety of the highway systems.

1.2.2. Pavement Condition Indices

Many researchers and engineers dedicated enormous time and efforts to standardize the indicator of pavement roughness throughout the world. They have established some well-known indices, including Ride Index, Present Serviceability Index, and Pavement Condition Index (PCI). However, those indicators are not repeatable by different agencies or even the same agency but at another time [6] [7]. Fortunately, the World Bank established International Roughness Index (IRI) [8] in 1986, and it has become the most commonly used roughness indicator around the world due to its high reproducibility.

The Transportation & Civil Engineering Division of Alberta (Canada) Infrastructure adapted IRI as its only pavement roughness measurement in 1998 [9] and applied it in two main aspects: 1) trigger value for rehabilitation; 2) rating standards.

To improve the pavement management efficiency and serve the public with better transportation infrastructures, the transportation department as a government agency needs to decide which roads need repair, rehabilitation, or just regular maintenance more systematically and effectively. Thus, IRI is employed as a trigger threshold between preventative maintenance and preservative repair. That is to say, preventative maintenance is required if the IRI of a segment is lower than the trigger value, while a preservative repair is considered if it is higher than the trigger value. The trigger values are listed in Table 1.1 concerning varying traffic levels. It is plain to see that the lower the traffic volume is, the higher the IRI trigger value is. In other words, roads of low traffic volumes have higher tolerances to poor pavement conditions and do not require frequent preservative repair, and vice versa.

Table 1.1: IRI trigger thresholds categorized by Annual Average Daily Traffic (AADT) [10].

AADT	IRI trigger threshold (mm/m)
< 400	3
400 - 1500	2.6
1501 - 6000	2.3
6001 - 8000	2.1
> 8000	1.9

To make the highway conditions transparent and clear to the public, the transportation department employs IRI again as a standard to rate highway conditions following the criteria proposed by Jurgens and Chan [11] in 2005. The detailed criteria are shown in Table 1.2. From the table, it can be seen that two roads can be rated differently even they have the same IRI value. For example, if the speed limit of the highway is higher (110 km/h), it must keep a smaller IRI (i.e., a better pavement condition) to get itself rated as the same condition class as the other roads of lower speed limits. That is, the rating system is stricter on roads of higher speed limits.

Table 1.2: Highway pavement condition rating criteria [11].

Condition	110 km/h Highways (mm/m)	Other Highways (mm/m)
Good	$IRI < 1.5$	$IRI < 1.5$
Fair	$1.5 < IRI < 1.9$	$1.5 < IRI < 2.1$
Poor	$IRI > 1.9$	$IRI > 2.1$

1.3. Introduction to IRI Measurement

1.3.1. State of the Practice

Considering the two substantial roles the IRI plays, it is not difficult to see the importance of IRI measurement. In Alberta, IRI measurement is run by specialized companies that use survey vehicles to measure IRI every 50-m along the pavement lanes. The survey vehicle, as shown in Figure 1.1, is mounted with PSP-7000 multi-laser inertial profiling system powered by 11 laser height sensors [12]. These lasers can measure the distances between the pavement surface and the front bumper of the vehicle. With these distances, the system can then plot the longitudinal profile of both the left and right wheel paths.



Figure 1.1. PSP-7000 survey vehicle [12]

After scanning the smoothed longitudinal profiles of the pavement (h_{ps}), the system then inputs the profiles into a computer algorithm established by Sayers [13] using a quarter-car model, as shown in Figure 1.2, to measure the IRI.

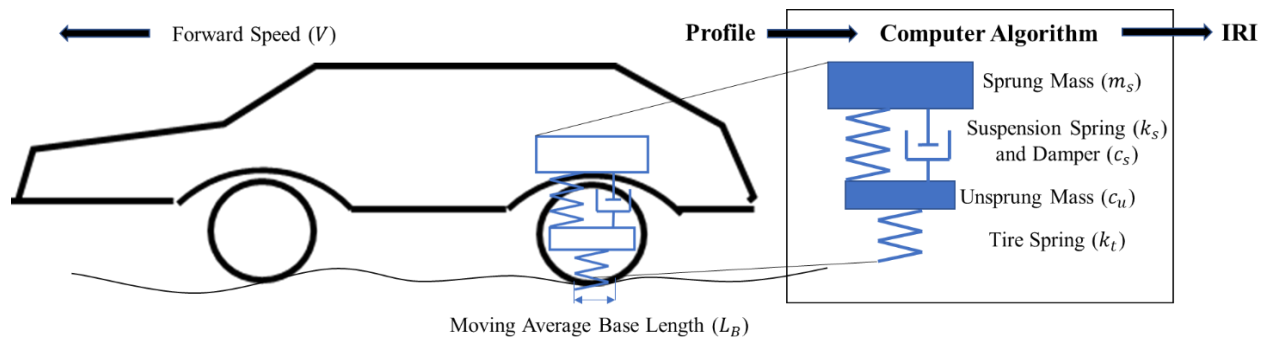


Figure 1.2. Quarter-car algorithm (adapted from [13])

National Cooperative Highway Research Program (NCHRP) [14] performed considerable research on ten alternative sets of parameters of this quarter-car simulation. The vehicle simulation with the best correlation is often called the Golden Car. Sayers adapted the Golden Car and normalized the parameters by the sprung mass, m_s . The normalized data set is shown below:

$$\begin{aligned}
c &= c_s/m_s = 6.0 \\
k_1 &= k_t/m_s = 653 \\
k_2 &= k_s/m_s = 63.3 \\
\mu &= m_u/m_s = 0.15 \\
V &= 80 \text{ km/h} \\
L_B &= 250 \text{ mm}
\end{aligned} \tag{1}$$

The quarter-car model then can be written in the following matrix form:

$$\begin{bmatrix} \dot{z}_s \\ \ddot{z}_s \\ \dot{z}_u \\ \ddot{z}_u \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -k_2 & -c & k_2 & c \\ 0 & 0 & 1 & 0 \\ 0 & \frac{c}{\mu} & -\frac{k_1 + k_2}{\mu} & -\frac{c}{\mu} \end{bmatrix} \begin{bmatrix} z_s \\ \dot{z}_s \\ z_u \\ \dot{z}_u \end{bmatrix} + h_{ps} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_1}{\mu} \end{bmatrix}, \tag{2}$$

where, h_{ps} is the smoothed profile elevation, z_s is the height of the sprung mass, z_u is the height of the unsprung mass, and the dot accents indicate time derivatives. The time (t) is defined by the simulated travelling speed (V) and the longitudinal distance (x):

$$t = x/V, \tag{3}$$

To solve the derivatives in Eq. (2), one needs to know the initial states at the starting time. Based on Sayer's research, he suggested using the first profile point as the initial z_s and z_u , and use the average change in profile per second over the first 11-m distance as the initial \dot{z}_s and \dot{z}_u . Based on his observation, only the first 20 m is influenced by the initialization, and then this influence diminishes quickly when the Golden Car travels longer distance.

The IRI is defined as an accumulation of the motion difference between the sprung and the unsprung mass, and then normalized by the length ($L = 50$ m in Alberta):

$$IRI = \frac{1}{L} \int_0^{L/V} |\dot{z}_s - \dot{z}_u| dt, \tag{4}$$

According to the above equation, units of IRI are m/km or mm/m. Instead of root mean square (RMS), the absolute difference is selected to ensure that the IRI scale is linearly proportional to the surface roughness (i.e., profile elevation) [7], and a larger IRI indicates a rougher surface.

As the key of IRI is the simulated vertical motion of the quarter-car model, it is reasonable to assume that factors such as aggregate size, cracks, potholes, and sealings can together determine the IRI because these factors affect the driving experience in our common sense. A study by Gong et al. [4] confirmed that factors like transverse cracks, fatigue cracks, rutting, block cracks, raveling, and potholes influence IRI in the manner that factors closer to the front position matter more. Or expressly, transverse cracks have the most impacts. These mentioned factors have a shared terminology named pavement distresses in the field of pavement management.

1.3.2. State of the Art

The cost of the current laser road surface tester is high in the aspects of operation and maintenance [15]. Therefore, many researchers turned to find low-cost alternative methods to identify IRI, and most of them chose Artificial Neural Network (ANN) models. The existing IRI estimation models using ANNs can be classified into two categories: 1) past-based and 2) present-based models.

Past-based ANN models are those networks whose inputs are historical data including, but not limited to, initial IRI, ages, distresses, climate, soil types, traffic, structure, etc. When the researchers built their models, most of them accessed the historical data from a program called the Long-Term Pavement Performance (LTPP) program [16] run by the FHWA. The LTPP program was initiated by the Strategic Highway Research Program (SHRP) in 1987 in order to extend pavement service life by a series of professional and systematic investigations. Ten of Canada's provincial highway agencies joined this program, including Alberta Transportation. LTPP collects performance data, including IRI, climates, traffic, deflection, distress, structure, maintenance, rehabilitation, etc. In 1992, Federal Highway Administration (FHWA) took over LTPP, and it remains this way until today. In LTPP's 2017 report [16], they proclaimed 2,509 pavement test sections with lengths of 500 ft (152.4 m) are monitored annually. This is a tiny portion compared to the United States' total highway mileage of 164,000 miles (264,000 km) [17], not to mention the total highway mileage around the world. More explicitly, if researchers want to predict the IRI of a specific road segment that is not included in the LTPP program by the most chance, they need to investigate the history of this segment by themselves to get all the necessary input data, which is obviously a cumbersome task. Due to this reason, past-based models are not yet widely used to predict the IRI.

Present-based models, on the contrary, take advantage of instant data, which can be all collected on-site at one time. The most common inputs are vibrations or accelerations. In spite of the lower accuracy in comparison to past-based models, present-based ones are more operable, and the schemes are closer to the nature of IRI definition (i.e., Sayer's quarter-car model) because the algorithm underlying the definition of IRI employs dynamic motion, which is a synonym of vibration and acceleration.

1.4. Problem Statement and Crowdsensing-Based Infrastructure Monitoring

As mentioned above, the most significant limitation of past-based models is that the public databases only cover an extremely small portion of the road sections. For most pavements, researchers have to collect the historical data by themselves during multiple investigations. Therefore, those models are usually not a practical measurement method.

In comparison, the present-based models are more practical in the sense that researchers can collect all the needed data on-site at once. However, all current present-based models use dynamic responses, including vibration and acceleration. Most of them need specially designed devices to collect the data. In addition, most of them require professional technicians to analyze and process those dynamic signals. Finally, it is noted that vibration data can be affected by the dynamic features of the testing vehicle [18].

In this context, imaging data can offer promising solutions. Images have the following advantages: 1) most of the modern cameras, especially sports cameras, have built-in stabilization systems; therefore, no specially designed devices are required to collect imaging data; 2) there exist many more DNN models initially designed to handle images than those to handle vibrations; 3) imaging data is more likely to exclude the effects of the dynamic features of the testing vehicle.

Additionally, this model could finally be implemented to the existing crowdsensing-based infrastructure-monitoring framework being developed by the researchers [19] [20] [21] [22].

This framework takes advantage of the rapid development of smartphones in terms of integrated sensors, data transmission, and computational power. Currently, the model can successfully employ smartphones mounted on vehicles as the vibration sensors, process signals and extract

features on board, pass the features to the cloud database. Finally, remote computers can systematically analyze the collected data to assess the bridges' health conditions.

To enrich the power of this framework, the IRI identification model in this thesis can ultimately be implemented in the same framework. First, the raw image data can be either collected by the cellphone cameras, the vehicle rearview cameras, or the other cameras connected to the phone by Bluetooth. Second, preprocessing of the image data, such as cropping and stitching images, can be accomplished onboard with the powerful processors of smartphones. Third, the data can be transmitted to and stored in the same cloud database. Fourth, remote computers can analyze the image data to estimate the IRI. It should be noted that all these steps can be carried out synchronously with the steps described in the previous paragraph sharing the same devices. It will thus make the current framework more integrated and efficient.

However, due to the limited scope, the implementation to embed the proposed model to the existing crowdsensing-based infrastructure monitoring framework is not accomplished in this thesis.

1.5. DNN and Transfer Learning

ANNs are inspired by biological neural networks, and they are the foundation of currently prevalent Artificial Intelligence (AI) technologies. ANNs are widely known because of their excellent performances in image recognition and classification [23]. ANNs treat input data as a bunch of numbers and tune the weights/biases parameters tactically to gain the optimum model. Generally, the deeper a network is, the more trainable parameters it has and thus the more information it can learn from the given datasets. So, ANNs are built deeper and deeper to solve more complicated problems, and these deeper networks are called Deep Neural Networks (DNNs).

On the other hand, Convolutional Neural Networks (CNNs) or Deep Convolutional Neural Networks (DCNNs) are the fundamental techniques that boost image recognition because of their excellent performances in extracting feature maps [24]. The core elements of CNNs are the convolution kernels, which are small matrices. These trainable kernels can not only slide along the width, height, and depth of an image to extract abstract features when propagating toward deeper layers but also reduce the number of computational parameters in ANNs [25]. In fact, to some extent, CNNs make it possible to solve complex image recognition tasks, which could never be

accomplished by classic ANNs [25]. Actually, there already exist many imaging-based models that can recognize pavement distresses using CNNs, and some of these models will be introduced in the literature review section.

DNNs have outstanding performances on classifying images by giving discrete properties, but IRI values are continuous decimal numbers that are not discrete. It raises the question of whether DNNs can predict continuous numbers as well. Fortunately, DNN can not only handle classification problems but can also work excellently in predicting continuous numbers [26]. The task to predict continuous numbers is called regression. There exist many applications using DNN to solve regression problems, such as using face images to predict human ages [27], using exterior and interior house images to predict house prices [28], and using DNA methylation level to predict human ages [29].

In spite of the incredible power of DCNNs, it is not always easy or practical to train a new model from scratch merely for a particular task. Transfer learning, on the other hand, is a practical approach to train neural networks in terms of saving time and dealing with limited data [30]. Transfer learning is inspired by the phenomenon that human beings can transfer knowledge across different tasks [30]. It is a helpful technique that allows developers to improve the performance or reduce the number of labelled data by transferring knowledge from a source domain to a target domain [30]. Therefore, transfer learning lends itself to developing new models for new circumstances, even though the effectiveness of such application may vary for different applications.

1.6. The Assumption in This Study

Despite the enormous successes DNNs have achieved in other regression problems, images of pavement surfaces have very different patterns from human face images or house images. The key features in a face image or a house image can be clearly distinguished from each other. For example, the essential elements in a face image might be the eyes, the nose, the mouth, and the number of wrinkles. In contrast, the critical components in a house image might be the furniture, the decoration, and the room size. These features are very different from each other. However, the differences in a pavement surface image are not always so obvious. For example, if the image is oriented or distorted, longitudinal cracks might be misrecognized as transverse cracks; sealed

cracks might be mixed with unsealed cracks. So, more evidence is necessary to support the assumption that DNNs can estimate IRI using pavement images.

The inspirations to assume that DNNs can identify IRI came from four articles. First, Ali et al. [24] showed that Inception-v3 could use imaging data as input and detect the cracks on structure surfaces. Second, Maeda et al. [31] declared that their team successfully used pavement images captured by smartphones and two different neural networks, SSD Inception V2 and SSD MobileNet, to classify pavement distresses such as longitudinal cracks, lateral cracks, alligator cracks, rutting, and potholes. Third, Mei et al. [32] successfully utilized the ConnCrack network and images captured by GoPro cameras to accomplish the task of crack semantic segmentation. Lastly, Karballaezadeh et al. [33] gathered the quantities of seven types of distresses as inputs and fed them into Multi-Layer Perceptron (MLP) Neural Network and Radial Basis Function (RBF) Neural Network to predict pavement condition index (PCI).

Every study contributes a critical link to the chain, which leads to the conclusion that IRI may be identified using pavement pictures. The first and second studies show that DNNs can not only detect but also distinguish different cracks. The third study shows that it is possible to measure the size of cracks and quantify cracks by counting the number of pixels occupied by predicted segmentation. With the efforts of the first three studies, it has been proved possible to obtain quantities of different distresses from the pavement images using DNNs. Combining this conclusion with the fourth team's analysis, it is reasonable to anticipate that DNNs can use pavement images as inputs and identify the IRI.

1.7. Objectives and Scope

Although there exist various DNN-based models to detect, classify, and quantify distresses using images and different NN models to identify the IRI using distresses, there is currently no model that directly uses pavement images as the input to estimate the IRI. Therefore, the objective of this thesis is to develop such a DNN-based model that could be the first attempt to use pavement images to estimate IRI directly. This model is expected to be able to predict IRI for a random 50-m lane segment. Rather than classifying pavement sections into the three rating conditions, this model considers IRI a regression problem because 1) the rating criteria are not identical between

provinces and countries; 2) agencies can determine the categorical conditions from the continuous IRI numbers.

To accomplish this objective, there is a necessary pre-processing step to find an automatic method to develop full-size undistorted pavement images. The images must be undistorted because sizes of distresses are significant when quantifying distresses, while it is difficult to measure their actual dimensions in a perspective image. So, the processes of preparing the 50-m long pavement images are also presented.

1.8. Organization of the Thesis

The organization of this thesis is as follows.

Chapter 2 presents the literature review of two types of ANN models to predict the IRI, past-based and present-based models. Although, to the best of the author's knowledge that there are no imaging-based DNN models to predict the IRI in the existing literature, many imaging-based DNN models have been developed to detect pavement distresses, which lays the foundation for the proposed research. Therefore, DNN models using imaging data to detect distress are also presented in this chapter.

Chapter 3 demonstrates the procedures of how data, including IRI ground truths and full-size pavement images, are collected and prepared.

Chapter 4 elaborates the architecture of a DNN called ResNet34 employed in the model. This chapter also shows how the transfer learning method is applied.

Chapter 5 records the training and testing results of the proposed model. As no empirical evidence is available, this chapter compares different training modes to suggest useful practices. Furthermore, this chapter also analyses how hyperparameters affect the performances and determines which values are the optimum ones. Based on the results, discussions about the proposed model and comparisons with other models are also addressed.

At last, Chapter 6 concludes this thesis and makes recommendations for future work.

2. CHAPTER 2: LITERATURE REVIEW

2.1. Past-Based ANNs to Predict IRI

Lin et al. [34] and his team may be the first team to attempt to use ANN methods to predict the IRI. In 2003, they built a tinny network with only 3 layers and 21 nodes. The network had 14 input variables, including road level and quantities of 13 types of distresses. A piece of record was a 1-km long lane segment where those 14 input variables were counted manually from the pavement images, and the corresponding IRI ground truths were measured by ARAN (Automatic Road Analyzer). They, in total, gathered 100 training records and 25 testing records but no validation records. After 15,000 training iterations, 9 IRI predictions in the testing dataset had errors smaller than 0.2 m/km, 17 of them had errors smaller than 0.5 m/km. To rephrase it, the accuracy was 9/25 (0.36) if taking 0.2 m/km as the error threshold, and 17/25 (0.68) if taking 0.5 m/km as the threshold. The accuracies may not be very high, but their efforts were still a vital milestone in using ANNs to predict the IRI.

As introduced in the introduction section, LTPP encourages researchers to use their data for all study purposes, so its database is publicly accessible around the world. Most of the oncoming models in the following paragraphs utilized the LTPP database for training their ANNs to predict the IRI.

After a decade from Lin's model, El-Hakim and El-Badawy [35] developed a slightly larger ANN with 4 layers and 36 nodes. Quantities of different distresses were still the input variables in their model, but the most significant change compared with Lin's input variables was that they included the initial IRI, namely the first IRI measurement after the pavement construction. They gathered a total of 184 samples from the LTPP program and randomly assigned 75% of them for training. Although they assigned the rest 25% to the testing dataset, they presented the overall predictive accuracy using all 184 samples but not the 25% testing dataset. Their model achieved an R squared value (i.e., coefficient of determination/goodness-of-fit) of 0.828 when comparing the predicted IRI with the measured IRI. The actual R squared of the testing dataset should be lower than 0.828 because ANNs are trained to fit against the training dataset as well as they can if overfitting is not considered.

Gong et al. [4] proposed a more refined modern ANN model using Random Forest Regression (RFR) technique. RFR algorithm, fundamentally, has many trees; each of them is an independent ANN model which can predict the IRI. The predictions of all the trees are then weighted averaged to get the final prediction. The team obtained a total of 11,715 data points from the LTPP program, and 25% of them were randomly allocated into the testing dataset. They reported very high accuracy with R square reaching 0.974 and mean square error (MSE) as low as 0.006 for the testing dataset. Besides the accuracy, they also analyzed the importance/sensitivity of every input variable. The initial IRI achieved the highest importance score of 100, and the other variables were compared with the initial IRI as the benchmark. The second and the third-essential variables were quantities of transverse cracks and fatigue cracks, which achieved 23.55 and 19.75 importance scores, respectively. In other words, the initial IRI was 4 to 5 times more important than the second and the third-ranking variables, and thus it dominated this RFR IRI prediction model.

After Gong et al. studied the sensitivities of the input variables, Zeiada et al. [36] and Fakhri et al. [37] separately reported similar results based on their observations. Zeiada et al.'s team investigated the warm regions in the LTPP program and developed a 30-neuron ANN whose R squared score of the testing dataset was 0.86. They also reported the initial IRI was more than twice important as the other variables. Instead of employing the LTPP database, Fakhri et al.'s team was based in Iran, and they collected their data using Laser Crack Measurement System (LCMS). They collected a total of 19,794 samples in the mode of 10-m road sections and 2,546 samples in the mode of 100-m section. Besides reporting the highest importance of the initial IRI, their team contributed new findings that the longer the pavement section was, the higher accuracies the identical ANN model could achieve. To numerically display a straight comparison for the two 15% testing datasets, the R squared were 0.647 and 0.834 for 10-m and 100-m segments, respectively.

Rather than the above models utilizing the initial IRI and distresses, Hossain et al.'s model [38] used only climate and traffic data to predict the IRI. They referred to the LTPP database and downloaded the required data for ten pavements located in four different climate regions across the U.S. Instead of having one model for all the four climate regions, they trained four independent ANNs. 15% of the dataset in each climate region was used for testing, and there were approximately ten testing points for each region. The results achieved very high R squared values,

0.98, 0.99, 0.98, and 0.99, respectively. Although the dataset in their study was relatively small, the authors admitted that some data was still missing from the LTPP database, and they had to assume or calculate them.

Georgiou et al. [39] and Marcelino et al. [40] had different schemes to predict the IRI than the previous models. They fed their models with the IRI for the past years and let the networks predict the IRI for the coming year. Despite the similar ideas of these two models, there were still some differences. Firstly, Georgiou et al.'s model relied merely on the IRI records for the past six years, whereas Marcelino et al.'s model consisted of traffic, pavement thickness, precipitation, temperature, structure and IRI for the past four years. Secondly, Georgiou et al.'s model measured the IRI with an interval of 10-m by their laser profile system, while Marcelino et al.'s model downloaded from both the LTPP and the Portuguese Road Administration database. The testing R squared score of Georgiou et al.'s models was 0.93, and the highest R squared score Marcelino et al.'s model achieved was 0.986.

To conclude, the major shortcoming of the past-based models is the difficulty of data collection. Firstly, many past-based models need the initial IRI, while the initial IRI can only be measured by other methods such as the laser profilometer plus Sayer's quarter-car model [12] and [13]. Secondly, although some models do not require the initial IRI, they need the quantities of distresses, which can only be counted manually¹. Thirdly, even with the existence of programs such as the LTPP, the overwhelming majority of the highways around the world are not covered. Thus, as mentioned previously, researchers must obtain data by themselves if they want to predict the IRI for a road segment that is not included in the existing programs.

2.2. Present-Based ANNs to Predict IRI

As early as 1991, Marcondes et al. [41] found that Power Spectral Density (PSD) characteristics of most pavement profiles are very similar. Their power density levels shift up and down with respect to a single parameter, the IRI. PSD curves are usually plotted on a logarithmically scaled Cartesian coordinate system, whose x-axis represents the spatial frequency (cycle/m), and the y-axis represents the power density (m^3/cycle). International Organization for Standardization

¹ Many models (e.g., imaging-based DNNs) have been developed to count distresses automatically, yet they are not approved by most of the transportation departments due to their accuracies and reliabilities.

(ISO) [42] further standardized PSD curves for different levels of IRI. Instead of spatial frequency, ISO normalizes the x-axis to be a frequency ratio by dividing it with a datum frequency while the y-axis remains the same. The normalized PSD curves are monotonically decreasing curves that move up and down with varying IRI levels. Therefore, one can determine the IRI classification of a pavement segment by comparing its PSD curve with the standard ISO PSD classifications.

Wong and Worden [43] proposed a 3-layer ANN by combining Nonlinear AutoRegressive with eXogeneous inputs (NARX) shunting neural network with a dynamic nonlinear regression model. They built their tribometer, which could record the displacement and friction of an actuator block with constant time intervals. The time series of the displacement was the input, while the corresponding friction forces were the output. They developed three sub-models for different excitation levels and reported the MSE to be 0.2282%, 0.3202%, and 0.1815% for the low, medium, and high excitation levels, respectively. Ngwangwa et al. [44] followed Wong and Worden's efforts and enlarged the NARX network to 22 layers. The updated network could use the vehicle's response in time domains as the input and predict the road profile as the output. After knowing the profile, the model transferred it into an existing function in MATLAB, which can generate the corresponding PSD curve. Comparing the predicted PSD curve with ISO PSD classifications, the IRI category could be determined for the target pavement segment. However, because the primary objective of Ngwangwa's team was to predict the profile and the PSD curve, the IRI prediction was a by-product obtained by manual visual comparison. Therefore, the accuracies of the IRI prediction were not reported in the article.

To achieve maximum utilization of the response signals and avoid unnecessary computation wasting, Qin et al. [45] proposed a Probabilistic Neural Network (PNN) classifier which employed both time and frequency domain features of the vehicle response and then reduced the number of features according to an algorithm called Maximal Relevance and Minimum Mutual Redundancy (mRMR). Only the filtered superior features were the final inputs of the PNN classifier. As its name implies, this PNN classifier model was designed to give discrete properties and distinguish the IRI types between six categories rather than continuous IRI values. The travelling speed of the vehicle was maintained to be 40 km/h (11.1 m/s), and the IRI classification interval was 1 sec (i.e., road section lengths were equal to 11.1-m). Data points equivalent to 600 sec of road sections were used for training, and 300 sec of road sections were collected for testing. In other words, 300 road

sections of 11.1-m were classified into six different road types for testing. This PNN classifier reached accuracies greater than 96% for the testing dataset.

However, Zhang et al. [18] revealed one shortcoming of those aforementioned models: the training models were designed for the specific dynamic configurations. It means that the models must be retrained if the probe vehicle is changed. To overcome this issue, Zhang et al. extracted common features by convolving vertical acceleration responses from two simulated vehicles having different dynamic characteristics. Only those extracted features were kept when training the networks, while the others were abandoned. Based on the author's opinion, the common features represented the shared information, the road roughness excitation, while the abandoned features described the unique vehicle mechanical properties. By that measure, the trained model can be applied to any acceleration data obtained from any regular travelling vehicles without considering its unique dynamic properties. Their proposed model contained two networks, one for the roughness classification task and the other for the IRI regression task. The ground truths of both IRI classification and regression are calculated by the PSD curves as mentioned above. For the classification job, Zhang et al.'s model consisted of 5 hidden layers containing 330 nodes and attained comparable testing accuracies compared to Qin et al.'s model. And for the regression job, which was untouched in Qin et al.'s model, the layer structure of the network was 13-10-1. The 10% testing dataset achieved an R squared of 0.9994, and 90% of the errors were within [-0.3, +0.3 m/km].

Given the growing popularity and functionality of smartphones, some researchers have been studying how to utilize them and exploring smaller, cheaper, and more convenient ways to evaluate road conditions. Souza et al. [46] collected accelerometer data using smartphones mounted on vehicles and further classified these features into two categories: temporal statistics and spectral statistics. They even ranked the importance of these features, with the irregularity of the data spectrum being the top one. They experimented with several networks and decided the Support Vector Machine (SVM) algorithm to be the best. However, considering their objective was for daily travelling purposes, they classified highway conditions into four labels rather than evaluating continuous IRI. Additionally, the way they defined a segment or an instance was by travelling time. For example, the travelling speed was 80 km/h (2.22 m/s), and the time interval was 10 sec for

highway conditions, so the interval length was 22.2 m. In total, they tested 4,688 instances, and the model achieved an average accuracy of 92.79%.

Compared with the difficulties of data collection in past-based models, the most significant advantage of present-based models is that all the required input data can be instantly obtained on-site at one time. However, the vibration/acceleration data are inconvenient to collect because they usually need carefully designed equipment, and these dynamic responses are prone to the vibration of the probe vehicles.

2.3. Imaging-Based DNNs in Distress Detection

To the best of the author's knowledge, although no imaging-based IRI identification DNNs have yet been discussed in the literature, imaging-based DNN methods have been widely used to detect, classify, and quantify distresses.

In Alberta, Surface Condition Rating (SCR) is required by the transportation department to provide important information about the pavement repair requirements and the estimated costs. It is clearly defined in the Contract Administration Manual for Highway and Bridge Maintenance [47] by Alberta Transportation that "Government field staff (or designate) manually rate pavement surfaces in terms of cracking, rutting, etc." However, "manually" means high labour and time investments. Thanks to the notable growing ability of DNNs on processing images, many imaging-based DNN models have been developed to accomplish these goals: detecting, classifying, and quantifying pavement distresses.

The first goal is to detect a specific category of distress. Ali et al. [24] modified the Inception-v3 network invented by Google Company and tuned it to detect cracks in tunnels. Due to the lack of labelled data, Gopalakrishnan et al. [48] chose transfer learning methods and developed multiple models to detect pavement cracks by exploiting ImageNet pre-trained VGG-16 DCNN with various final classifier layers, including Single Neural Network (NN) layer classifier, Random Forest (RF) classifier, Extremely Random Trees (ERT) classifier, Support Vector Machines (SVM) classifier, and Logistic Regression (LR) classifier. Rather than detecting cracks, Kumar et al. [49] tuned the Inception-v2 network, and Milhomem et al. [50] tuned the Xception network to detect potholes.

Succeeding the detection of a specific type of distress, the next goal of pavement distress recognition is to classify multiple categories of distresses. Peraka et al. [51] obtained pavement surface data in videos and images from various public agencies and then converted them into gray-scale images to reduce computational costs. They drew rectangle bounding boxes to manually label the images with 18 distress classes and trained them by a YOLOv4 (You Only Look Once version 4) network. In the validation experiment, the model achieved 87.44% mean average precision. Ibragimov et al. [52] preferred to use merely top view images, so they collected images by two cameras mounted at the rear of a vehicle. They labelled the distresses with bounding boxes in three colours representing three distress types: linear cracking, alligator cracking, and partial patching. For the network structure, they used Faster R-CNN. In the performance evaluation, they reported average precisions of 38.1%, 77.1%, and 84.6% for these three distress categories, respectively.

After detecting and classifying distresses, the final goal is to quantify the distresses. The most used technology is semantic segmentation, which can measure the distresses at pixel level by plotting the actual shapes of the distresses. Yet again, semantic segmentation can also be instantiated by DNNs. Bang et al. [53] utilized the CrackForest dataset and trained 71 images with the Deep Residual Network by transfer learning. In the testing results on the remaining 47 images, the model achieved a precision of 93.57%. Mei and Gul [32] are credited with developing a novel DNN called ConnCrack for crack semantic segmentation. They compared ConnCrack with the other methods such as CrackTree, ResNet152-FCN, CrackNet-V, etc., and found that ConnCrack consistently achieved the highest scores on both external datasets (CFD) and self-developed datasets (EdmCrack600).

3. CHAPTER 3: DATA COLLECTION

To train an imaging-based DNN for IRI identification, not only the image data for all pavement segments are needed but also the IRI ground truths of them. Therefore, the first step is to get the IRI ground truths. The IRI measured posted by the Government of Alberta are taken as the ground truth for this application. The second step is to collect the imaging data of pavement segments. During this step, a series of image operations are required to obtain a full-size nondistorted top view image of each pavement segment.

3.1. IRI Ground Truths

Alberta Transportation annually posts the IRI data in Excel sheets [54]. Information including road numbers, lane codes, control section numbers, GPS coordinates, and IRI are easily accessible to the public. For each 50-m segment, the latitudinal and longitudinal coordinates of the starting and ending points are recorded as well as its IRI. The sheets record the inside (left-wheel path) and outside (right-wheel path) IRI for each segment. Table 3.1 is a sample of how the information is organized in the government documents.

Table 3.1. Sample of International Roughness Index Data [54].

Road No	Lane Code	From Latitude	From Longitude	To Latitude	To Longitude	IRI Inside	IRI Outside	Date Collected
831	R1	54.1706	-112.8007	54.1710	-112.8007	2.08	1.20	8/25/2020
831	R1	54.1710	-112.8007	54.1715	-112.8007	1.26	0.98	8/25/2020

Each row in the table represents a 50-m long segment. With the precise GPS coordinates, the starting and ending points are pronounced for every segment and can be used to link each image to its corresponding pavement segment.

Before using the IRI datasheets, those road sections where repair and/or rehabilitation were conducted after IRI measuring dates were filtered out by comparing the measuring dates with the 2020/2021 Provincial Construction Program [55] released by the Government of Alberta. The remaining road sections had not undergone any significant changes after the last time the IRI was measured until the image data collection. The road lanes were restricted to those running from south to north so that the potential influences of the vehicle's shadow were eliminated because the

GoPro camera was mounted at the rear of the car. Thus, during summer, the vehicle's shadow was in front of the vehicle in the Northern Hemisphere. Afterwards, five road sections were picked near the City of Edmonton from the filtered road sections, and they are indicated in Figure 3.1 with thick blue lines. Their corresponding starting and ending GPS coordinates are illustrated in Table 3.2. These sections are not dispersed but relatively concentrated in northeast of the city due to budget and time limitations.

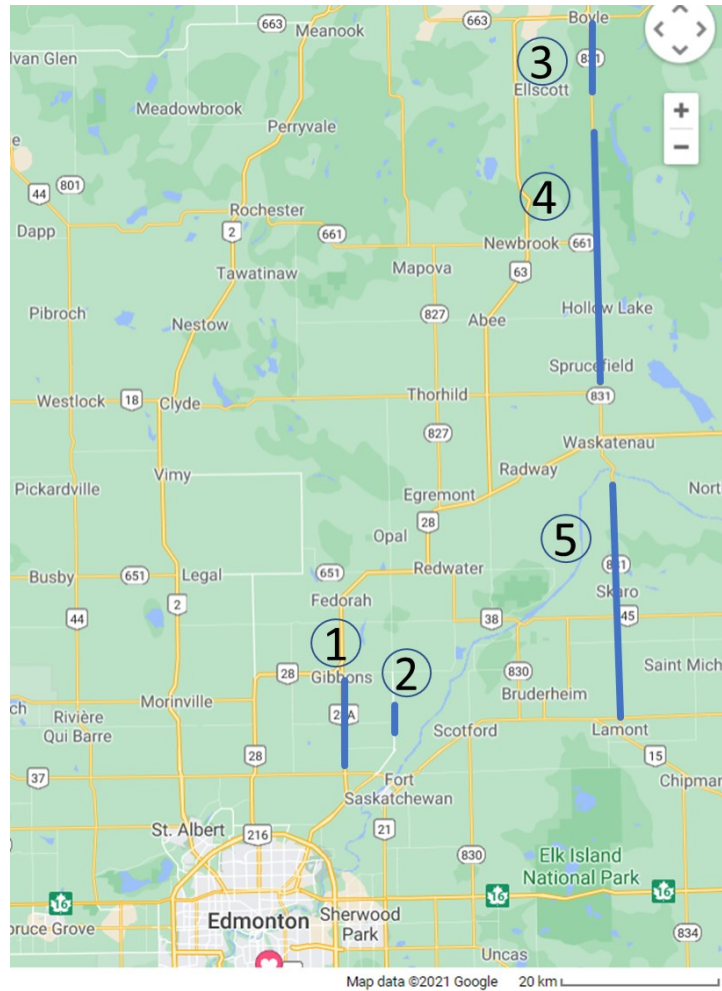


Figure 3.1. Maps of the investigated road sections (modified from [56]).

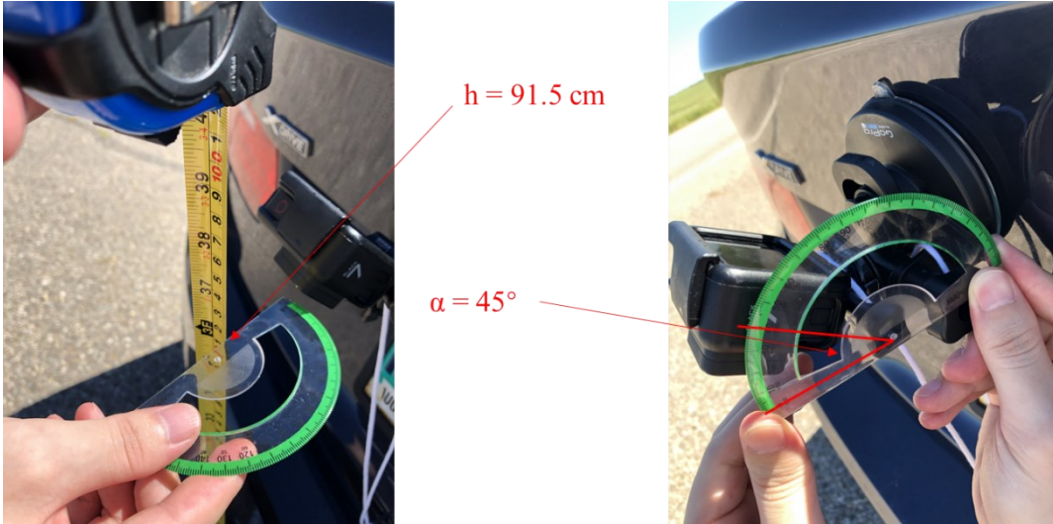
Table 3.2. Coordinates of the investigated road sections.

Road Section Number	Starting Coordinates	Ending Coordinates
1	53.721275, -113.322186	53.831332, -113.322202
2	53.767252, -113.223227	53.798220, -113.223478
3	54.505855, -112.803361	54.584044, -112.803360
4	54.170603, -112.800752	54.461410, -112.803575
5	53.777423, -112.777537	54.045223, -112.777529

3.2. Collection of Image Data

3.2.1. Camera Setup

To collect imaging data more efficiently, videos were first filmed in linear mode (i.e., fish-eye effect eliminated) with a GoPro Hero 7 Black camera and then converted to images. The camera was mounted at the rear of a BMW 328i xDrive, as shown in Figure 3.2. This GoPro Hero 7 Black camera can capture high-resolution videos of 1920 x 1080 pixels at a frame rate of 120 fps. Owing to the built-in stabilization systems, this camera can capture high-quality images even if the cruise speed is 100 km/h. The mounting height is denoted as $h = 91.5$ cm, and the camera angle is $\alpha = 45^\circ$ as indicated.



a. Height
 b. Shooting angle
 Figure 3.2. GoPro Hero 7 Black mounted at the rear of BMW 328i xDrive.

The cruise speed was 100 km/h. Still, deviations happened when passing through villages or in a situation where the speed of the front vehicle was slower because the testing vehicle was not allowed to change lane or overtake but must stay within the investigated lane to avoid missing any data. But these deviations had no impacts when stitching the full-size images because the stitching processes relied only on the GPS coordinates rather than the speeds.

3.2.2. Overview of Image Preparation Procedures

Figure 3.3 illustrates the overview of the steps to collect imaging data. Videos are taken at the beginning and then converted into sorted (i.e., in order of time) image frames in step 1. For each 50-m segment, there should be one relevant image carrying all the distress information. Any extra portions outside of this segment must be excluded from this image. In addition, this image must be top view because sizes and quantities of distresses are critical in IRI identification. As an example, if the image is a perspective view, two cracks on the pavement surface of equal length will occupy different pixels in the image due to perspective distortion, with the one farther from the lens taking fewer pixels. This effect could greatly influence IRI identification. Therefore, the calculated perspective transformations are performed on those images taken with a rear-mounted camera to get their top view versions in step 2. Nonetheless, it is difficult to have a single image covering the entire 50-m long segment, so several sorted images are stitched together in the designed sequences to compose a completed image in step 3. Since the IRI is divided into the left

wheel path and right wheel path IRI in the ground truths, the stitched images are further cropped into the left wheel path images and the right wheel path images in step 4. Finally, the left and right wheel paths images are ready to be fed into the training networks. The detailed explanations for image preparation are illustrated in the following sections.

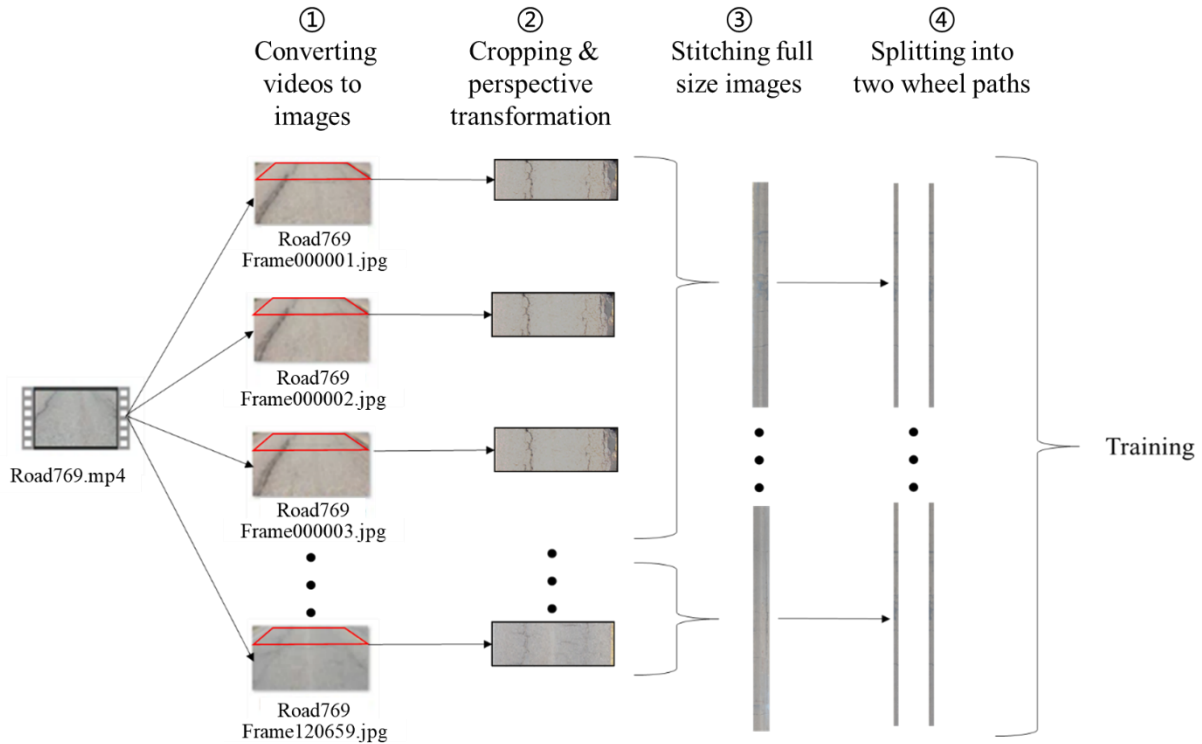


Figure 3.3. Schematic procedures of the imaging data preparation.

3.2.3. Perspective Transformation

The side view of the experimental setup is illustrated in Figure 3.4. The vertical field of view (FOV) of this camera is 55.2° [57]. A, B, C and D are the points on the pavement surface. A', B' and C' are the corresponding projecting points on the filmed images. Theoretically, A', B' and C' can translate freely along the camera's looking direction depending on the size of the image, but these points are forced to align on a line crossing point C in order to simplify the calculations.

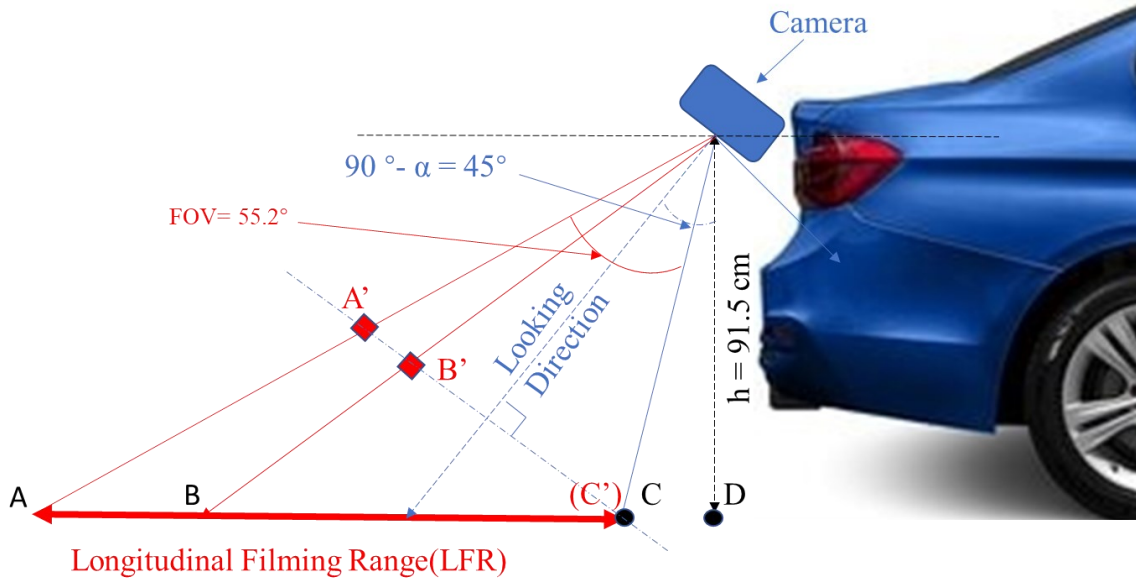


Figure 3.4. Side view of the camera setup (modified from [58]).

By doing some simple geometrical calculations, we can get the LFR indicated in the above figure:

$$\begin{aligned}
 AD &= h \cdot \tan(90^\circ - \alpha + FOV/2) = 291.977cm \\
 CD &= h \cdot \tan(90^\circ - \alpha - FOV/2) = 28.674cm, \\
 LFR &= AC = AD - CD = 263.303cm
 \end{aligned}
 \tag{5}$$

Figure 3.5 shows a sample image of the pavement surface. The portion surrounded by trapezoid Q'R'S'T' is of interest because only this portion contains the pavement information across the entire lane width R'S'. In other words, the part closer to the camera has some missing pavement information, as indicated in the dashed blue triangles.

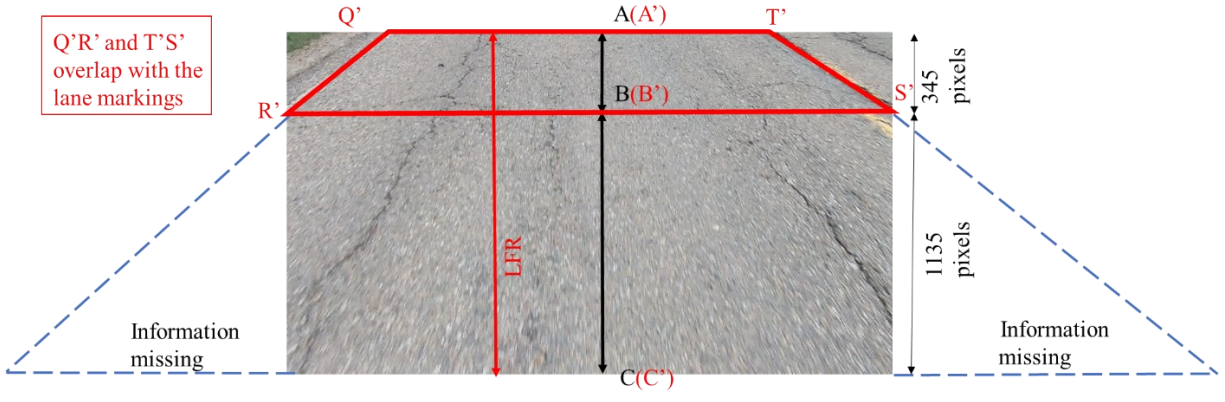


Figure 3.5. Sample image.

If we want to perform a perspective transformation on trapezoid $Q'R'S'T'$ to get its real shape on the actual pavement surface, as shown in Figure 3.6, the actual ratio between QR and RS is needed. In addition, it is known that the lane width (length of RS) of highways in Alberta is 3.5 m (350 cm). Thus, the problem turns to calculate the length of QR or AB on the actual pavement surface.

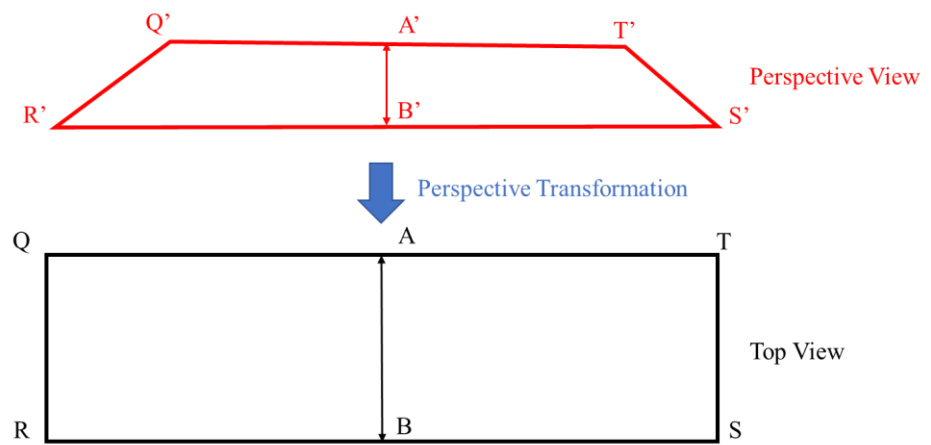


Figure 3.6. Perspective transformation.

To calculate AB , Figure 3.4 is modified as shown in Figure 3.7. We take D as the origin $(0, 0)$ and establish a Cartesian coordinate system. All the known coordinates are labelled in cm. $L1$ connects points A' , B' and C' . $L2$ is the line connecting A and its projecting point A' in the image. Similarly, $L3$ connects B and its projecting point B' .

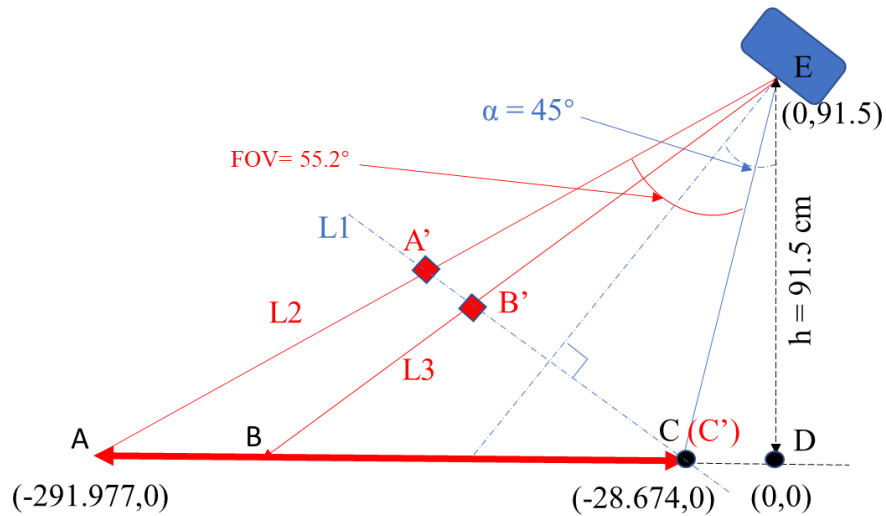


Figure 3.7. Side view in a Cartesian coordinate system.

The steps to calculate AB are shown below:

As L1 inclines 135° and L1 crosses C $(-28.674, 0)$, so L1: $y = -1(x + 28.674)$

L2 connects A $(-291.977, 0)$ and E $(0, 91.5)$, so L2: $y = 0.313(x + 291.977)$

L1 intersects L2 at A', so A' $(-91.442, 62.768)$

$\frac{A'B'}{B'C'} = \frac{345 \text{ pixels}}{1135 \text{ pixels}}$ (Figure 3.5), so B' $(-76.810, 48.136)$

L3 crosses B' $(-76.810, 48.136)$ and E $(0, 91.5)$, so L3: $y = 0.5646x + 91.5$

L3 intersects the x-axis at B, so B $(-162.062, 0)$

A $(-291.977, 0)$ and B $(-162.062, 0)$, so AB = 129.9 cm

Thus, the aspect ratio of the transformed rectangle QRST is 129.9/350. A proportional resolution of 325 x 875 pixels is appropriate for the rectangle QRST. Figure 3.8 shows a sample that demonstrates the performance of the proposed perspective transformation.

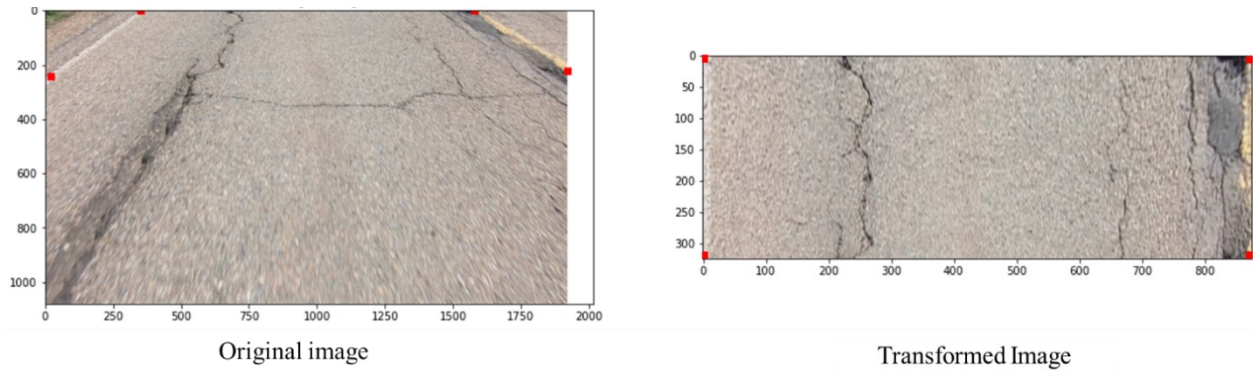


Figure 3.8. Sample of calculated perspective transformation.

3.2.4. Verification of the Proposed Perspective Transformation

To verify the theoretical calculations in last section and confirm the proposed perspective transformation is acceptable, it is necessary to measure the length AB in situ. But considering safety risks on an actual highway pavement, an indoor experiment was conducted to simulate a highway lane, and the configuration is shown in Figure 3.9. The camera is mounted on the wall with the same height and angle in the actual case, and it is located in the middle of the simulated 3.5-m wide highway lane. The rectangle Q'R'S'T' is the portion of interest.

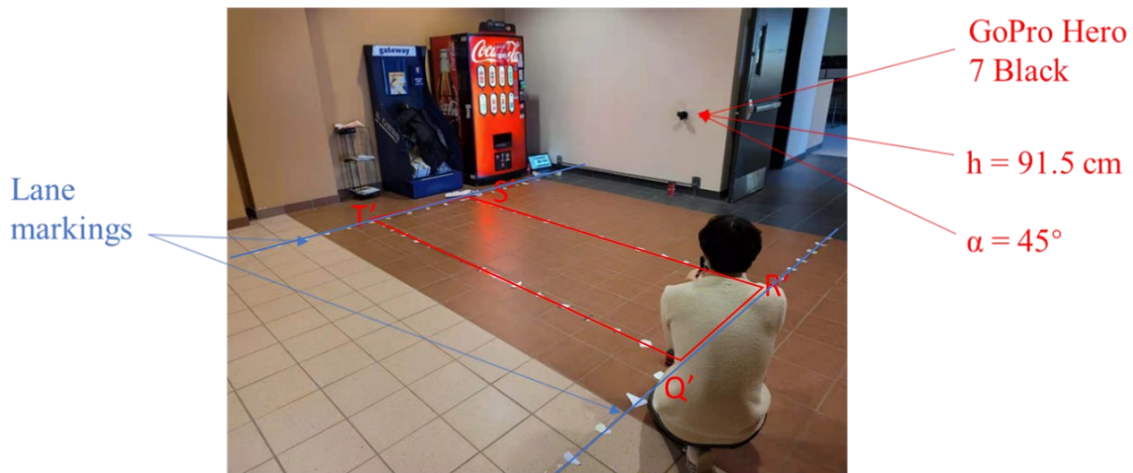


Figure 3.9. Scene simulation of a highway lane.

The dimensions of the rectangle and its distance from the wall (i.e., simulated car tail) were measured onsite. The calculated and the measured results are compared in Figure 3.10. Compared with the measured 120 cm, the calculated 129 cm is very close and acceptable.

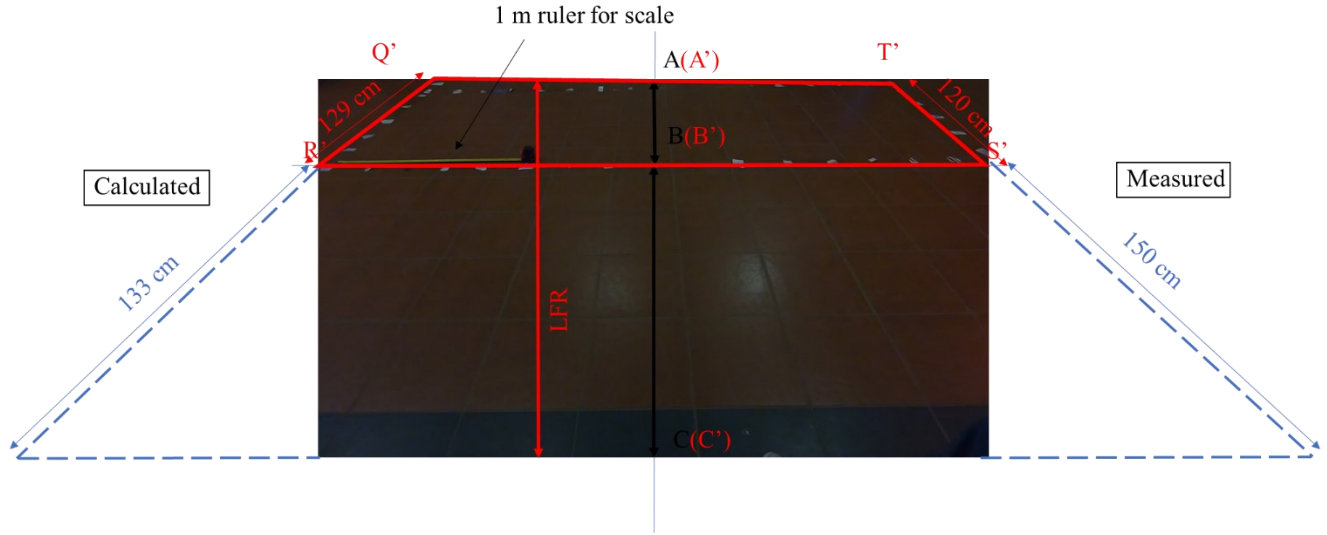


Figure 3.10. Comparison between theoretical and measured AB length.

3.2.5. Image Stitching

GoPro Hero 7 Black camera can not only film high-resolution images at a high travelling speed but also record the geographical coordinates once several frames, as shown in Table 3.3.

Table 3.3. Sample of geographical data recorded by GoPro Hero 7 Black.

Date	GPS(Lat.) [deg]	GPS(Long.) [deg]	Relative time [sec]	Frame number
2021-07-10T18:44:03.114Z	53.7061583	-113.3200184	0	0
2021-07-10T18:44:03.480Z	53.7062536	-113.3200185	0.366	43.92
2021-07-10T18:44:03.847Z	53.7064032	-113.3200174	0.733	87.96
2021-07-10T18:44:04.214Z	53.7064307	-113.3200178	1.1	132
2021-07-10T18:44:04.819Z	53.7064714	-113.3200172	1.705	204.6

The first column is the date and the time that the video was recorded, where “T” means nothing but a separator between date and time and “Z” stands for Zulu time (i.e., Greenwich Mean Time (GMT)). In the fourth column, relative time counts the passing time from the first frame by calculating the time difference between the current row and the first row. As the FPS of the camera is set to 120, relative time is multiplied by 120 fps to get the frame number in the last column.

Recalling the IRI ground truth sheets (Table 3.1), each row represents a 50-m long segment with its starting and ending coordinates recorded. By comparing its starting GPS coordinates with the GoPro ones, the corresponding starting frame number can be attained. If there are no identical coordinates in the GoPro GPS sheet, linear interpolation will be applied to locate the frame number. The frame number is finally rounded to the closest integer if it is decimal. To describe the interpolation method, an example is given in Figure 3.11. It is given a starting point of a segment, and its GPS coordinates are (53.7063143,-113.3200180). The interpolations are performed only referring to the latitudinal coordinates in the sense that the vehicle travelled only in the south-north direction. The trivial deviations in the longitudinal coordinates are due to the vehicle's sway and thus can be ignored.


	Date	GPS(Lat.) [deg]	GPS(Long.) [deg]	Relative time [sec]	Frame number
①	2021-07-10T18:44:03.114Z	53.7061583	-113.3200184	0	0
②	2021-07-10T18:44:03.480Z	53.7062536	-113.3200185	0.366	43.92
	Starting point:	53.7063143	-113.3200180		62
③	2021-07-10T18:44:03.847Z	53.7064032	-113.3200174	0.733	87.96

Figure 3.11. Sample calculation of using linear interpolation method get the frame number.

To find the corresponding starting frame number, firstly, we need to find its location in the GoPro GPS sheet, which is between row ② and row ③. Then we calculate the difference between the starting point and row ②, also the difference between row ③ and row ②, and the ratio of the two differences:

$$\frac{53.7063143 - 53.7062536}{53.7064032 - 53.7062536} = 0.405749, \quad (6)$$

Afterwards, use this ratio to find the corresponding starting frame number:

$$43.92 + 0.405749 \cdot (87.96 - 43.92) = 61.79, \quad (7)$$

Finally, the decimal number 61.79 is rounded to the closest integer number 62 as the frame numbers must be integers in real life.

Or equivalently, the above interpolation procedures can be described in the following equation form if the latitude of the given point is Lat_i , and the two latitudes bounding Lat_i are Lat_{low} and Lat_{high} in the IRI ground truth sheet:

$$Frame_i = int \left[Frame_{high} + \frac{Lat_i - Lat_{high}}{Lat_{low} - Lat_{high}} \cdot (Frame_{low} - Frame_{high}) \right], \quad (8)$$

where $Frame_i$ is the corresponding frame number of the given point, $Frame_{low}$ and $Frame_{high}$ are the related frame numbers of Lat_{low} and Lat_{high} .

Using the same method, one can also get the ending frame numbers. After knowing the starting and ending frame number of each 50-m (5,000-cm) long segment, it is ready to stitch the transformed frames together to get a completed image. If the starting frame number is N and the ending frame number is M , and each frame will contribute the same small segment length of l_{cm} if the speed is assumed to be constant within the interesting segment. As indicated in Figure 3.12, l_{cm} can then be calculated as:

$$l_{cm} = \frac{5,000cm}{(M - N + 1)}, \quad (9)$$

The ordinary length of l_{cm} is 25 cm and it takes around 200 frames to compose a full-size lane image.

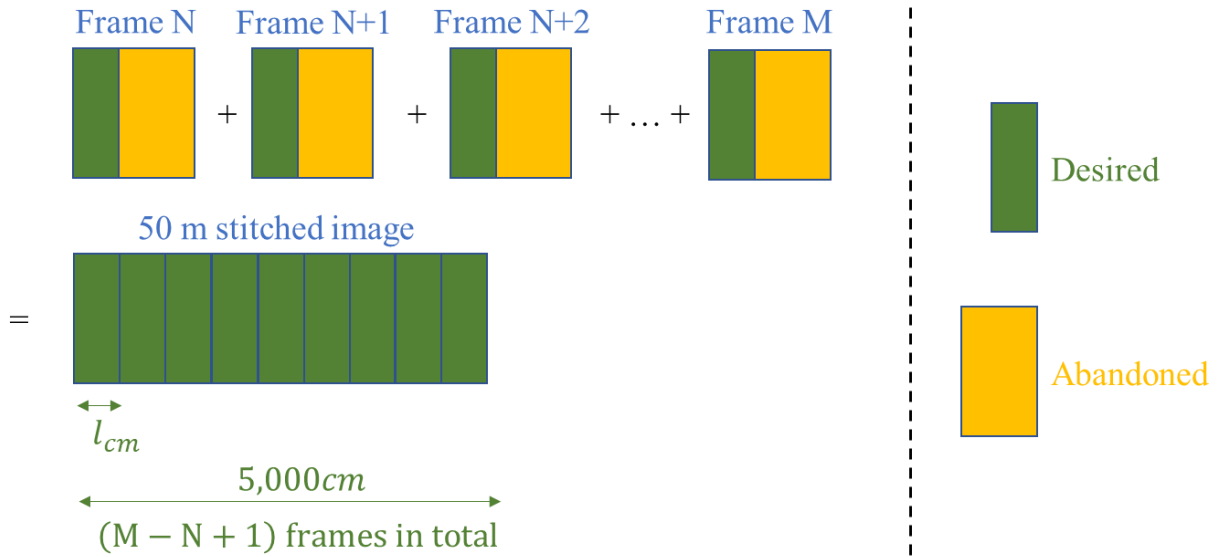


Figure 3.12. Stitching transformed frames to get a completed image for one segment.

Using the proportionality between the actual rectangle size (129 x 350 cm) and the image pixels (325 x 875 pixels) as indicated in the transformed image in Figure 3.13, the desired length in the unit of pixels, l_{pixel} , can be written as:

$$l_{pixel} = \frac{l_{cm}}{129cm} \cdot 325pixels = \frac{5,000cm}{129cm(M - N + 1)} \cdot 325pixels, \quad (10)$$

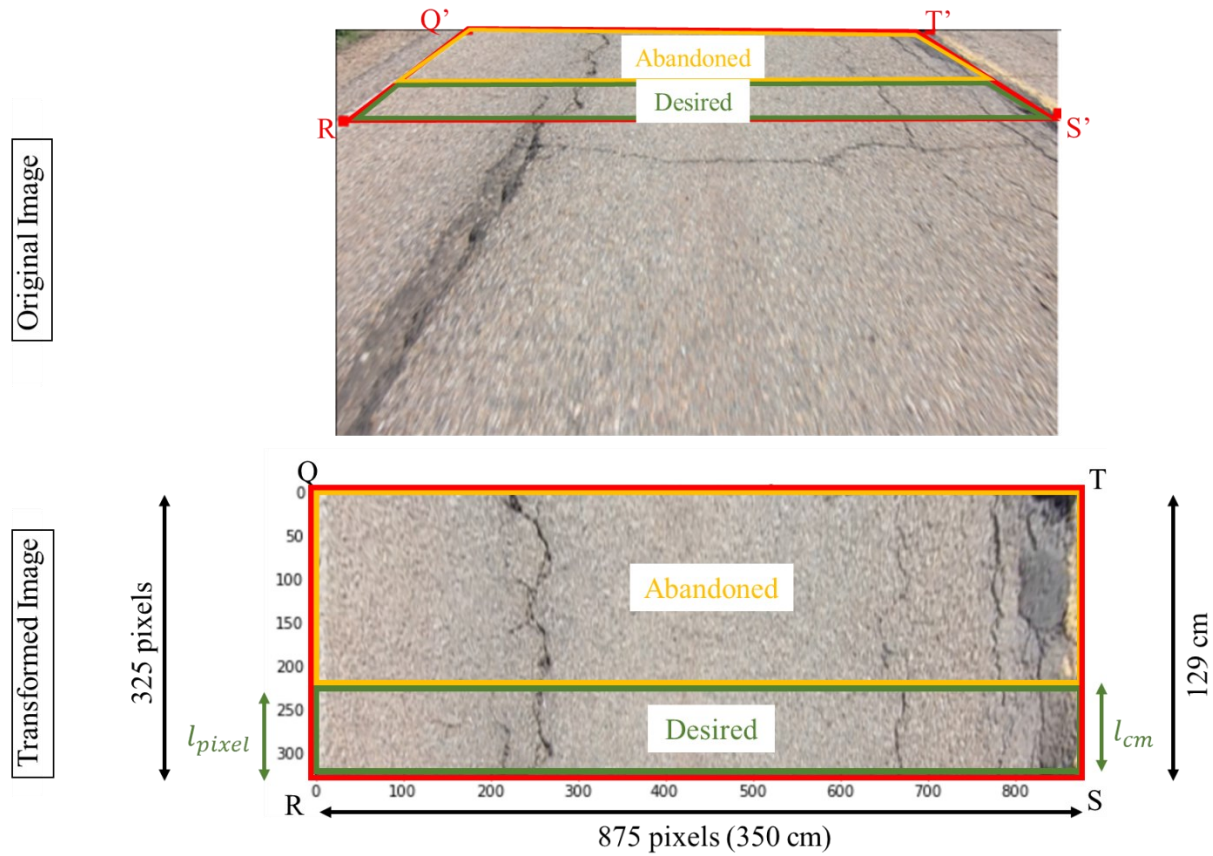


Figure 3.13. Sample of how a transformed frame is cropped.

As indicated in the original image in Figure 3.13, within the trapezoid $Q'R'S'T$, which covers the full lane width, the bottom portion surrounded in green box is preferred because it is closer to the lens and thus has undergone less perspective transformation. Stitching all the transformed frames together, the resolution of the stitched full-size image is $12,500 \times 875$ pixels, which is proportional to 50×3.5 m ($5,000 \times 350$ cm) on a real 50-m long lane segment. However, due to insufficient computer memory, the original resolution of $12,500 \times 875$ pixels is too large to experiment with. Therefore, the images are furtherly resized to $6,250 \times 438$ pixels, and the height/width ratio is unchanged.

To demonstrate the performance of the proposed stitching method, Figure 3.14 shows one of the stitched full-size lane segment images and the zooming in details.

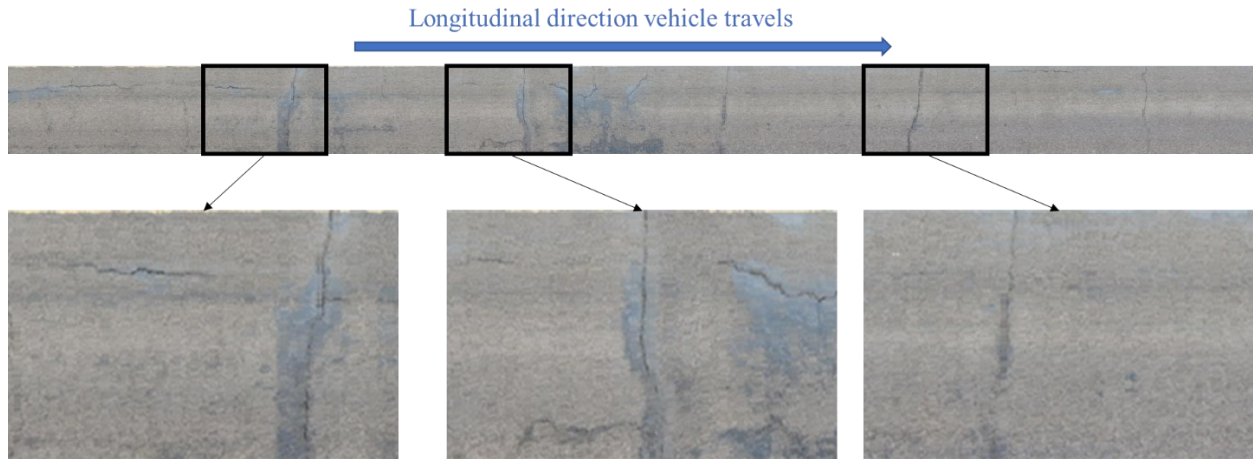


Figure 3.14. A sample of the stitched full-size image composed of 200 frames (rotated 90°).

However, to be consistent with a pair of IRI for left and right wheel paths, it is necessary to further crop the stitched images into the left and right wheel path images. However, it is almost impossible to control the vehicle in the perfect middle of the lane at all times. Thus, the cropped images must cover wider than the wheel width to take into account the sway of the vehicle. Therefore, two image strips of 6,250 x 145 pixels were cropped to represent two 1.16-m wide wheel paths. 1.16-m is much broader than the real wheel path width because it is necessary to ensure all useful information is included. In total, 2,128 images were collected, half of them representing left wheel path segments and the other half representing right wheel path segments.

3.2.6. Verification of GoPro GPS Accuracy

Because the stitching method relied solely on the GPS coordinates to locate the image frames, it is necessary to ensure that the accuracy of the GoPro GPS receiver is acceptable. So, an experiment that utilized Google Maps [56] was designed to check the GoPro GPS accuracy. The experiment was carried out in the Varsity Field on campus at the University of Alberta. Firstly, an obvious mark that is visible on the satellite layer in Google Maps was selected, and its coordinates were recorded (Point 1). Secondly, an image was taken at this point with the GoPro camera and the coordinates were recorded too (Point 2). Finally, the distance between these two points was calculated using a website application called Movable Type Scripts [59], which can not only calculate the distance between two GPS coordinates by also show them in Google Maps, as

indicated in Figure 3.15. The described steps were repeated three more times, and the results are listed in Table 3.4.



Figure 3.15. Screen shot of the Movable Type Scripts (modified from [59]) [56]).

Table 3.4. Errors of GoPro GPS coordinates.

Point 1 [deg]	Point 2 [deg]	Distance [m]
53.5245479, -113.5291026	53.524546, -113.5291179	1.033
53.5247350, -113.529678	53.5247303, -113.5296764	0.5332
53.5243730, -113.5301385	53.5243736, -113.5301417	0.2218
53.5247405, -113.5292207	53.5247357, -113.5292384	1.286
Average Error:		0.7685

As indicated in the above table, the average error of the GPS receiver in the GoPro camera was 0.7685 m. In Figure 3.16, red circles represent the possible erroneous camera locations.

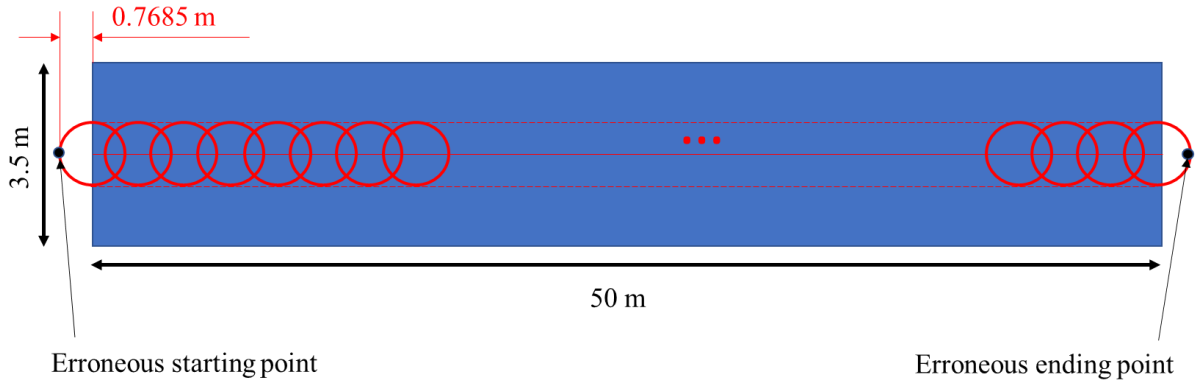


Figure 3.16. Configuration of the camera path in a highway lane (not to scale).

Their centres compose the actual path of the camera, which is denoted by the solid red line in the figure. On the other hand, the area composed by all the red circles is the possible erroneous path of the camera. In the figure, it is the area enclosed by the dashed line. As indicated that the possible camera path is entirely within the lane width in the transverse direction, so it is not expected to cause any problems when locating the image frames transversely. In the longitudinal direction, the erroneous areas are the two semicircles at the two ends. In the extreme condition, the stitching method could wrongly locate the starting point at the leftmost of the erroneous camera path and locate the ending point at the rightmost as indicated in the figure by the two black dots, so the possible longitudinal error in percentage is $\frac{2 \cdot (0.7685m)}{50m} \times 100\% = 3.07\%$, which is small. Therefore, the overall accuracy of the GoPro camera GPS receiver is acceptable.

4. CHAPTER 4: DNN MODEL AND TRANSFER LEARNING

One of the most well-known effective DNNs is ResNet, invented by He et al. [60] in 2015. The novel residual learning blocks improved the performance of the ResNet in image recognition. They made the family of ResNet quickly become one of the most popular DNNs after its publication. The specific network used in this model is called ResNet34, where 34 represents the number of convolutional layers. Its initial task was to classify images between up to 1,000 categories. To accomplish the regression problem of IRI identification, it was necessary to modify several portions of ResNet34. Additionally, we also updated it with multiple latest techniques, which were not applied in He et al.'s work in order to make the model more efficient and accurate. In this chapter, the first version of ResNet34 in He et al.'s paper is demonstrated in section 4.1. Then, the modifications and the corresponding reasons are addressed in section 4.2. At last, transfer learning of a pre-trained model is presented in section 4.3.

4.1. The Invention of ResNet34

4.1.1. Overview Architecture of ResNet34

As indicated in Figure 4.1, the depth of the feature map changes when passing through convolutional layers. After convolutions, the FC layers consist of an average pooling layer, and the softmax layers are appended to calculate the property distribution over output classes.

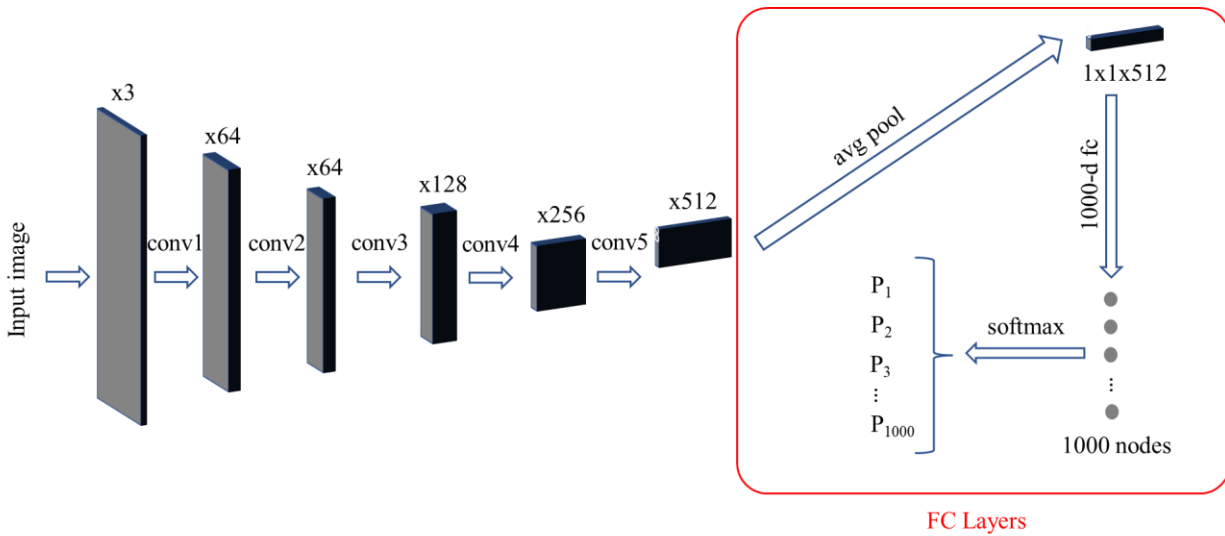


Figure 4.1. Architecture of ResNet34 with the depth of each layer annotated.

4.1.2. Convolutional Layers

As introduced in section 2.3 in the literature review section, convolutional layers are the key elements in CNNs in order to extract features of images. However, in fact, these convolutional layers (i.e., conv1, conv2, etc.) in Figure 4.1 are composed of one or more convolutional kernels and/or pooling layers. The detailed arrangements of each layer are shown in Figure 4.2.

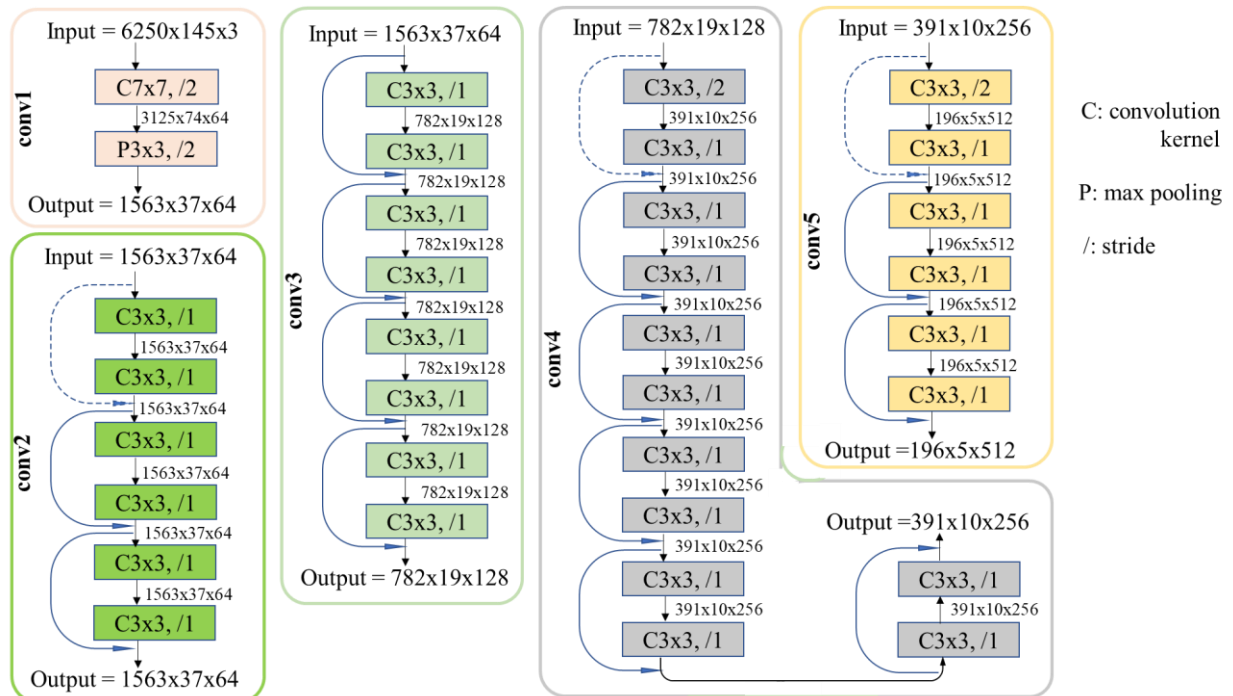


Figure 4.2. Compositions of convolutional layers in ResNet34.

To better explain the benefits of the residual network, it is necessary to introduce its counterpart version, the plain network, which has the exact same overall architecture in Figure 4.1 with only slight changes in the convolutional layers as indicated in Figure 4.3.

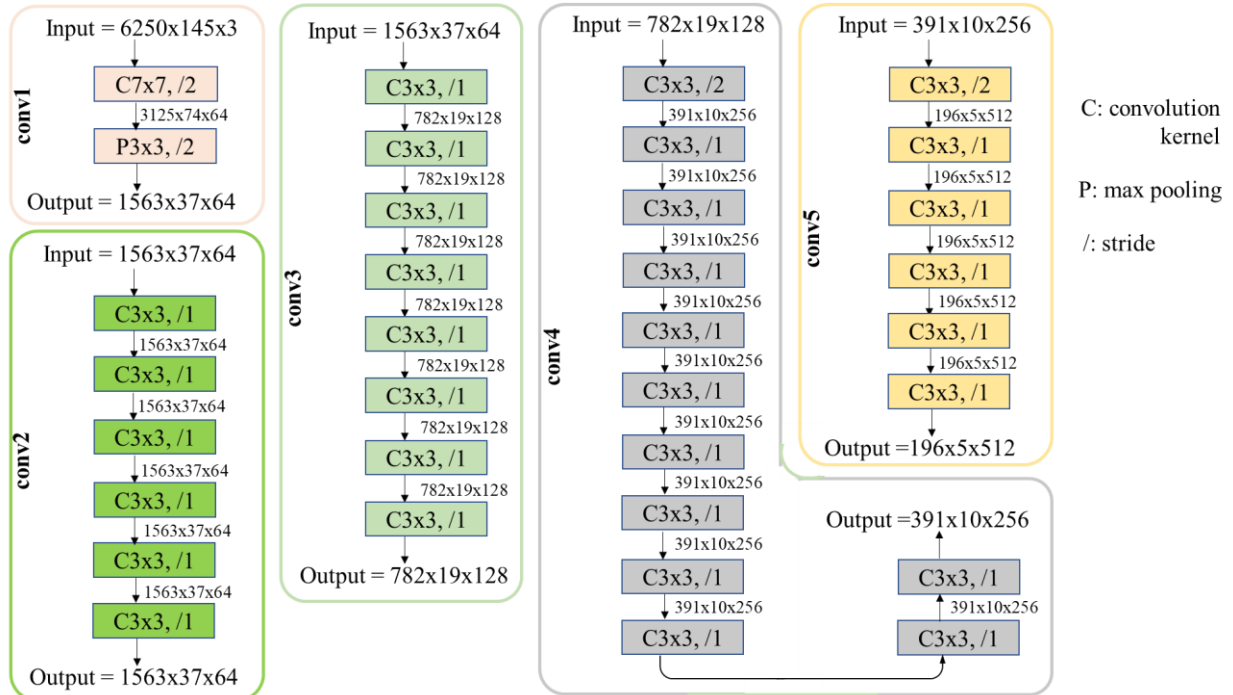


Figure 4.3 Composition of the plain network.

Taking a close observation of these two figures, it is not hard to see that the nuances are those curved arrows in Figure 4.2. As a matter of fact, those curved arrows in blue (dashed line arrows representing dimension/channel changes) are the cores that distinguish a residual network from a plain one. These curved arrows are shortcut connections, and this algorithm is called residual learning. Namely, the network learns the residual mapping instead of an unreferenced mapping.

4.1.3. Residual Learning

To distinctly show the advantages of residual learning, one typical building block from ResNet34 and the other one from the plain network are taken out, respectively, and they are put under amplifier as demonstrated in Figure 4.4.

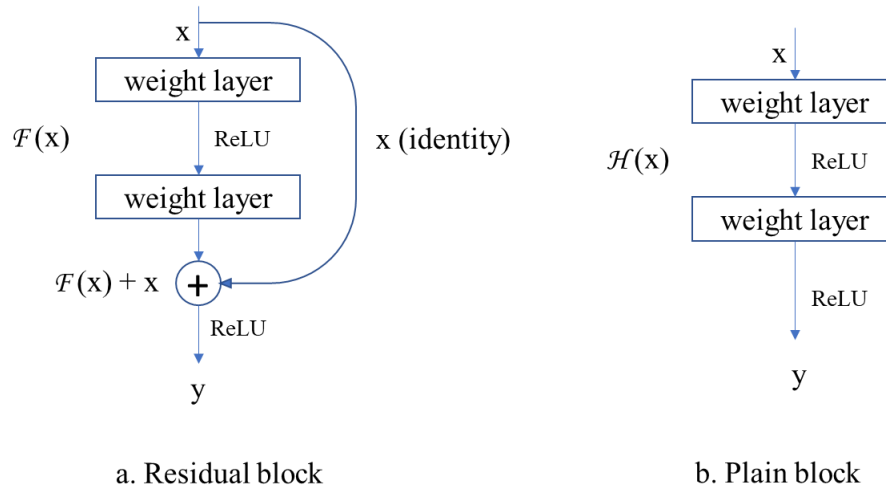


Figure 4.4 Comparison of residual block and plain block.

Say the output ground truth of these two blocks is \tilde{y} , the task of the blocks is to use gradient descent methods to optimize the deviation between y and \tilde{y} . However, this task can be done in two ways, one is by optimizing the original, unreferenced mapping $\mathcal{H}(x)$, and the other is by optimizing the residual mapping $\mathcal{F}(x) = \mathcal{H}(x) - x$. The transformation from a plain block to a residual block is realized by the shortcut connection, which is merely an identity mapping without adding any extra trainable parameters or computational complexity. Based on He et al.'s hypothesis, the residual mapping would be easier to optimize. One extreme example is that if the optimal function of $\mathcal{H}(x)$ were an identity mapping, it would be easier for the optimizer to drive the residual $\mathcal{F}(x)$ to zero than to find the identity mapping by a series of nonlinear functions.

However, the above transformation between a plain block and a residual block has a prerequisite that \mathcal{F} and x must have the exact dimensions. To solve the situations where their dimensions do not match (those dashed shortcut connections), x is multiplied by a linear projection \mathcal{W}_s . Although \mathcal{W}_s contains trainable parameters, the total number of these parameters is negligible for a 34-layer network, which costs computational power of 3.6 billion FLOPs (floating point operations per second).

He et al. performed comparison experiments with the residual and plain network on multiple datasets, including ImageNet, CIFAR-10 and COCO. The results showed that the residual network

always performs better than the plain network, and the gap expands when the depth of the network increases. That is to say, the residual network outperforms more if both networks have more layers.

4.1.4. FC Layers

As indicated in Figure 4.1, the feature map is a 1-dimensional vector after passing through the average pooling layer. And as introduced before, ResNet34 was initially designed to recognize images between 1,000 output classes; thus, 1,000 corresponding properties are wanted. Two steps are required to accomplish it, namely linear projection (fc-1000) and softmax algorithm.

To expand the 1x1x512 vector to 1,000 categories, a fully connected linear projection as shown in Figure 4.5 is performed. However, these 1,000 numbers are yet not properties between 0 and 1 that have an accumulated summation of one.

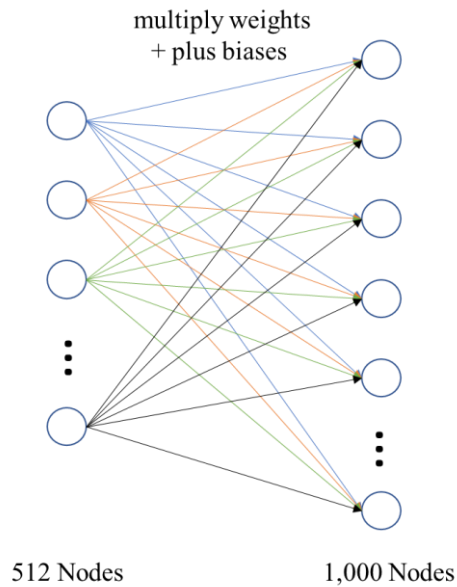


Figure 4.5. Linear projection of fc-1000.

To obtain the property distribution of these 1,000 output classes, the softmax function shown in Eq. (11) is employed:

$$P_i = \frac{e^{y_i}}{\sum_{j=1}^n e^{y_j}}, \quad (11)$$

where P_i is the property of the image belonging to category i , y_i is the corresponding input number and e is the Euler's number.

The main reason for choosing softmax is that it can not only convert these 1,000 random numbers into the corresponding properties between 0 and 1 but also ensure they add to one.

4.2. Modifications of ResNet34

4.2.1. Overview Architecture of the DNN after Modification

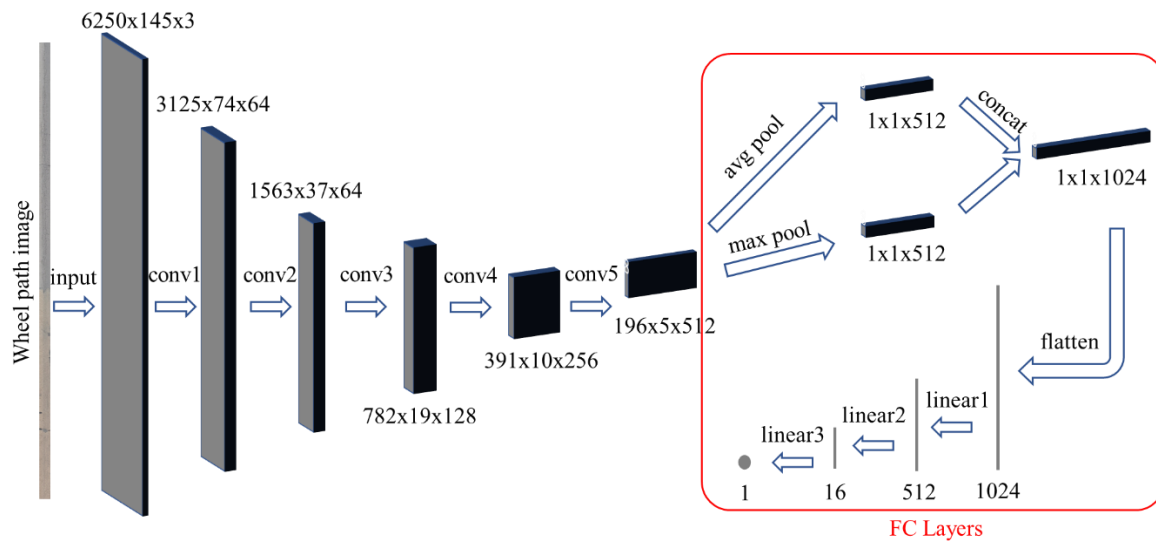


Figure 4.6. Architecture of the modified ResNet34 [60].

Taking one of the wheel path images as the sample input, the dimension of the feature map changes when passing through convolutional layers, as indicated in the above Figure 4.6. To accomplish the regression problem of IRI prediction, the head part or fully connected (FC) layers bounded by the red box in Figure 4.1 are removed and replaced with new FC layers, as shown in Figure 4.6. The following sections explain how the FC layers are modified.

4.2.2. Mixed Pooling Layer

Max pooling and average pooling layers have their own advantages when solving different problems. However, determining which will dominate in a new circumstance is utterly empirical, and there is yet no theoretical conclusion on it. To address this situation, Yu et al. [61] proposed a

mixed pooling method by adding trainable weights to both max and average pooling operators and then concatenating them to get a new mixed pooling layer by the following Eq. (12).

$$y = \lambda \cdot y_{max} + (1 - \lambda) \cdot y_{avg}, \quad (12)$$

where λ is a random weight between 0 to 1, y_{max} is the result of max pooling, and y_{avg} is the result of average pooling.

In this model, λ was fixed to be 0.5 to keep both features from max and average pooling layers but meanwhile reduce the trainable weight parameters to save computing resources.

4.2.3. Linear Transformation

To reduce the 1,024-dimensional feature vector to a 1-dimensional vector, three trainable linear functions as shown in Eq. (13) are appended after flattening the 1x1x1,024 tensor.

$$\mathbf{y} = \mathbf{w}\mathbf{x} + \mathbf{b}, \quad (13)$$

where \mathbf{y} is the output vector, \mathbf{x} is the input vector, \mathbf{w} is the weight and \mathbf{b} is the bias.

For example, to reduce a 4D vector \mathbf{x} to a 2D vector \mathbf{y} :

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \quad (14)$$

By doing so, $2 \cdot 4 + 2 = 10$ parameters were added to this layer. Similarly, by reducing the dimension within three steps shown in Figure 4.6, 533,025 trainable parameters were added rather than only 1,025 if reduced within one step.

The inputs are normalized, and 50% dropouts are applied before every linear transformation to accelerate the training and avoid quick overfitting. Additionally, ReLU (rectified linear unit) activation function was employed after every linear layer. The detailed information of the tuned FC layers is listed in Table 4.1.

Table 4.1. Summary of FC layers.

Layer Type	Output Shape	Parameter Number	Trainable
AvgPool2d	1 x 1 x 512	0	False
MaxPool2d	1 x 1 x 512	0	False
Flatten	1024	0	False
BatchNorm1d	1024	2,048	True
Dropout	1024	0	False
Linear	512	524,800	True
ReLU	512	0	False
BatchNorm1d	512	1,024	True
Dropout	512	0	False
Linear	16	8,208	True
ReLU	16	0	False
Linear	1	17	True

4.2.4. Loss Function

The other changed thing is the loss function, which is not indicated in the architecture overview part. It is known that the critical task of a DNN is to find the parameters that minimize the loss specified by the experimenter. In He et al.’s ResNet34, cross-entropy in Eq. (15) was employed because it was a classification problem, and the property distribution was discrete.

$$C = - \sum_{i=1}^n y_i \log(P_i), \quad (15)$$

where C is the cross entropy, n is the number of categories to classify, y_i and P_i are the ground truth and prediction property of category i , respectively.

However, to meet the needs of this study, a new loss function is required for the regression problem. There are two main loss functions to address regression problems, namely L1 and L2 losses. L1 loss (i.e., absolute error loss) accumulates the absolute differences between ground truth and prediction pairs. L2 loss (i.e., squared error loss), on the contrary, calculates the squared differences between each pair and then sums them up. Both of them were experimented with, but it was discovered that neither of them produced satisfactory outcomes. Therefore, another loss

function, the smooth L1 loss function, is introduced. It is the sum of Huber's losses as shown in the following Eq. (16):

$$Huber's\ loss = \begin{cases} \frac{1}{2}(IRI_{ground} - IRI_{pred})^2 & \text{for } |IRI_{ground} - IRI_{pred}| \leq \delta, \\ \delta \left(|IRI_{ground} - IRI_{pred}| - \frac{1}{2}\delta \right), & \text{otherwise.} \end{cases}, \quad (16)$$

where, IRI_{ground} is the ground truth and IRI_{pred} is the prediction. δ is equal to 1 in default.

As shown in Figure 4.7, the Huber's loss mitigates the sensitivity to the outliers compared with L2 loss but meanwhile keeps the advantage of L2 loss that the gradient gets smaller when the network is about to reach the minimum loss point.

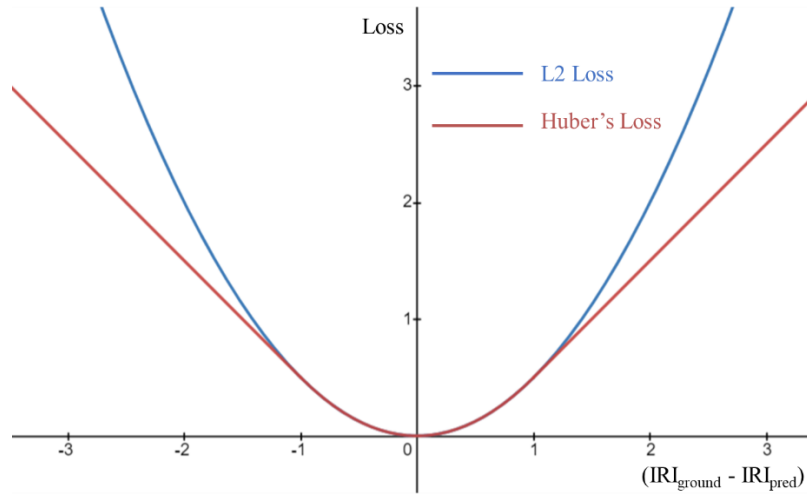


Figure 4.7. Comparison of L2 loss with Huber's loss in terms of $(IRI_{ground} - IRI_{pred})$.

4.2.5. Adam Optimization Algorithm

After defining the loss function, an efficient and feasible optimizer is needed to minimize the loss quickly and save computation power. He et al.'s ResNet34 employed Stochastic Gradient Descent (SGD) algorithm (Eq. (17)) [60].

$$w = w - \eta \nabla Q(w) \quad (17)$$

where w is the trainable parameters, η is the learning rate, and $\nabla Q(w)$ is the gradient of the loss function.

The authors used some tricks during iterations to save the computation power, such as updating the gradient according to a mini-batch size of 256 and dividing the learning rate by 10 when the error plateaus. However, because of the fast development of machine learning, another optimization algorithm called Adaptive Moment Estimation (Adam), proposed by Kingma and Ba [62] in 2015, has been proven more effective. Ruder [63] in 2016 compared the most common optimization algorithms in his paper “An overview of gradient descent optimization algorithms”. Based on his suggestion, “Adam might be the best overall choice”.

Adam is an extension of SGD that combines the advantages of two other algorithms: Adaptive Gradient Algorithm (AdaGrad) [64] that can deal with sparse gradients, and Root Mean Square Propagation (RMSProp) [65] that can deal with non-stationary objectives. At every timestep (t), Adam firstly estimates the first moment (m_t) and the second moment (v_t) of the gradients (g_t) as Eq. (18):

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \quad (18)$$

$g_t = \nabla_w Q_t(w_{t-1})$ are the gradients with respect to w at timestep t , where w is trainable parameters. $\beta_1, \beta_2 \in [0,1)$ are the exponential decay rates for the first and second moment estimates, respectively. The suggested default settings are $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial values of moment vectors are zero (i.e., $m_t = v_t = 0$ when $t = 0$), but the authors of Adam noticed that these two moments are biased towards zero at the first few iterations. To address this problem, they introduced bias-corrected moment estimates (i.e., \widehat{m}_t and \widehat{v}_t) as shown in Eq. (19):

$$\begin{aligned} \widehat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \widehat{v}_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned} \quad (19)$$

And then, the corrected moment estimates are employed to update the weights w :

$$w_t = w_{t-1} - \eta \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \varepsilon}}, \quad (20)$$

where, η is the learning rate and its default value is 0.001, ε is the parameter for numerical stability to avoid zero in the denominator, and its default value is 10^{-8} .

In this study, the hyperparameters changed a bit that $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\varepsilon = 10^{-5}$.

4.2.6. Cyclical Learning Rate

It is known that the learning rate might be the most crucial hyperparameter when training a DNN, which is usually determined by empirical knowledge or by experiments on the entire network with varying learning rates. However, the learning rate used in this DNN is intentionally left blank in the previous sections because there are no empirical suggestions for imaging-based IRI identification DNNs. It seems the only option is exhaustive searching which may cost a lot of time and computation power. However, fortunately, Smith [66] developed a new strategy that trains the network over a few iterations and cyclically varies learning rates between reasonable boundary values at each mini batch. The recorder will plot the losses against learning rates, and typically the point with the deepest decreasing slope represents the optimum learning rate. Figure 4.8 shows the cyclical learning results based on the dataset and model in this thesis. The red point denotes the deepest slope where the fastest convergence is happening. It is between 10^{-3} to 10^{-2} , and we picked the lower boundary to avoid overshooting.

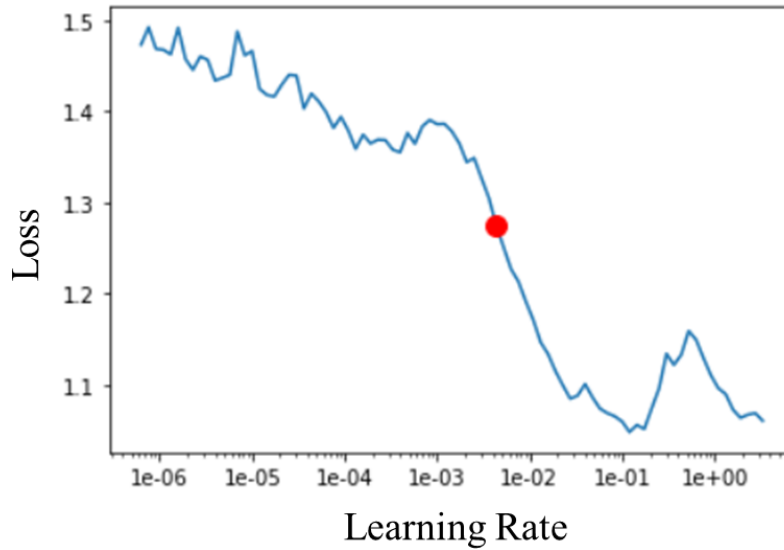


Figure 4.8. Loss vs learning rate curve.

4.3. Transfer Learning

As discussed in the introduction section, transfer learning is a practical option to develop new models for new circumstances, even though the effectiveness of such application may vary for different applications. Also, considering the limited size of the labelled dataset, the proposed DNN model took advantage of a pre-trained ResNet34 model from the PyTorch library [67], which had been pre-trained on ImageNet [68] database to classify 1,000 categories. Again, only part of the model was employed due to the differences between a classification and a regression problem. Expressly, only the initial parameters in the convolutional layers (i.e., body part) were adopted from this pre-trained ResNet34 model.

For the FC layers (i.e., head part), there was no source domain that the model could transfer parameters from because these layers were explicitly designed for the IRI regression problem. Therefore, the network had to assign random initial values to those parameters in the FC layers. However, it is not economical to train the whole network knowing that the FC layers are in the state of absolute randomness and disorder. So, before starting training the entire network, the parameters in the convolutional layers were frozen, and only those in FC layers were trained for one epoch. Afterwards, the convolutional layers were unfrozen, and the entire network was trained.

5. CHAPTER 5: RESULTS AND DISCUSSIONS

5.1. Training and Testing Results

Since the dataset size with the IRI larger than 4 was too small to be trained and tested on, those were removed for the simplicity of the study. In other words, this study only focused on the data that have an IRI smaller than 4. After filtering, 1,013 left wheel path images and 1,022 right wheel path images remained. The histograms in Figure 5.1 show the distributions of these two datasets. These distributions were consistent with the overall highway conditions across Alberta that 58% were in good conditions, 26.4% in fair conditions, and 15.6% in poor conditions [69].

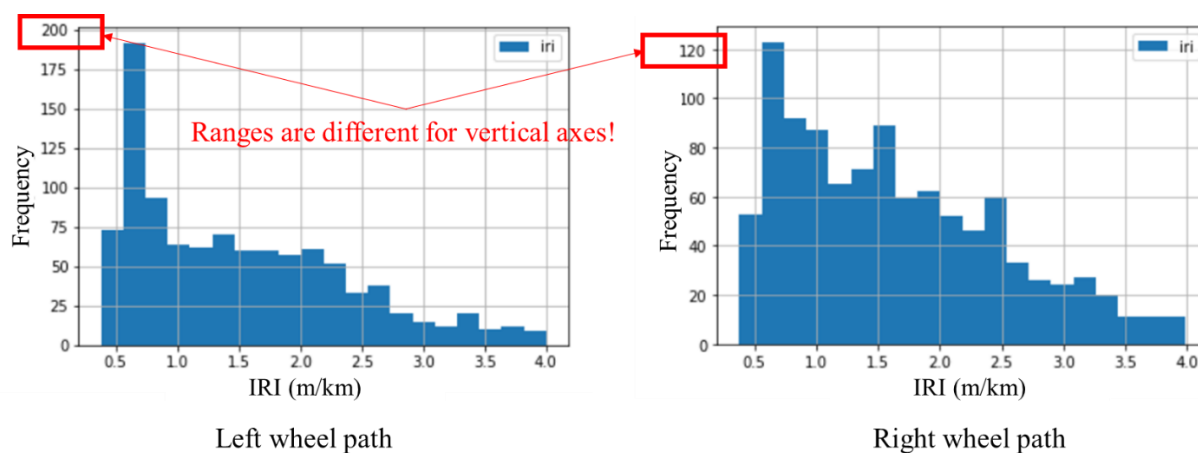


Figure 5.1. IRI distributions.

The complete data was divided into three datasets: the left wheel path dataset, the right wheel path dataset, and a dataset combining both the left and right wheel paths. Three same independent models were trained and tested on the three datasets, respectively. In each dataset, 68% was randomly assigned for training, 17% was assigned for validation, and the rest 15% was used for testing. It means that there was no overlapping between training, validation, and testing subsets within every group of experiments. Data in the training subsets were used to train the model, while data in the validation subsets were used to monitor the instant performance after every training iteration. After every model was appropriately trained, the testing subsets were employed to test the final performances of the predictions. The results of training, validation, and testing are listed and discussed in the following sections. All the experiments were run on Google Colab Pro, where a Tesla V100-SXM2 GPU with 16 GB memory was available.

5.1.1. Separated Left and Right Wheel Path

The number of training epochs was set to 100 for each group of experiments, and the batch size was set to eight. Figure 5.2 shows the learning curves for the two groups. The vertical axis is the Huber's loss, while the horizontal axis is the number of iterations but not epochs. The difference between epoch and iteration is that: an iteration means a batch of data (i.e., eight images) passing through the model, whereas an epoch means that the entire training subset is visited once by the model. For example, to complete one epoch of the left wheel path training subset having $1,013 \times 68\% \cong 689$ images, it took $689/8 \cong 87$ iterations. As the total number of training epochs was 100, the total number of iterations was $87 \times 100 = 8700$.

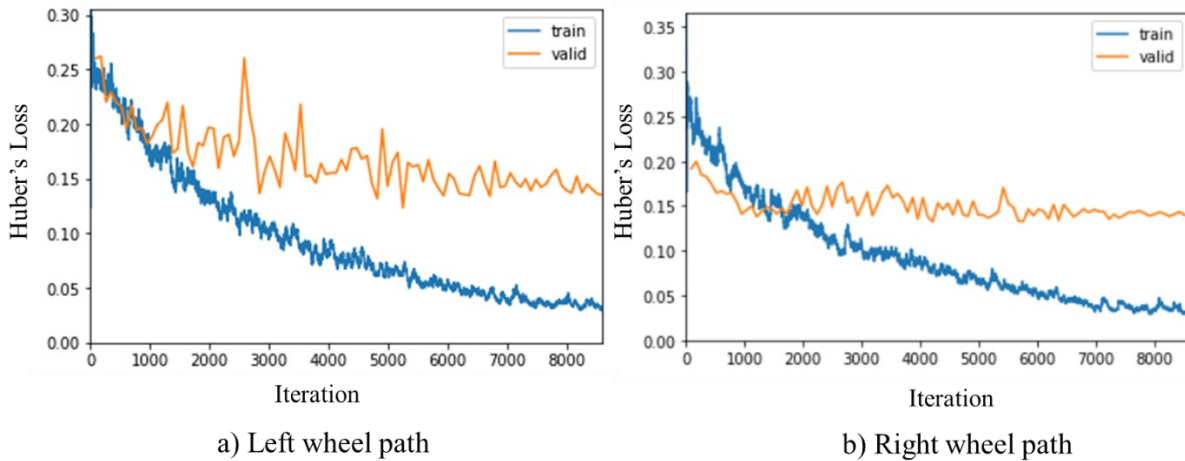


Figure 5.2. Training and validation loss of left wheel path and right wheel path separately.

As shown in the above learning curve diagrams, the initial training and validation losses of the left wheel path dataset started approximately from the same value of 0.25, while the validation loss of the right wheel path dataset was 0.05 lower than the training loss. To reveal the reasons, the experiments were repeated multiple times only to find that these two patterns occurred stochastically for both datasets. It might be because the training and validation datasets were randomly split, and thus the initial fitting conditions against these two subsets varied with different splitting subsets. This problem could be fixed if the sizes of datasets were large enough so that one random subset could capture the same statistical characteristics of the other random subsets. However, due to the limited resources and efforts, it was not practical to collect more data in this study. This was also the main reason that the transfer learning method was used.

Another characteristic of Figure 5.2 is that the validation losses of both groups started to level out after approximately 1,500 iterations (~ 20 epochs), while the training losses kept decreasing. This was because the networks were trying to fit against the training subsets as well as they could but not the validation subsets. This phenomenon of the divergences between training and validation loss is called overfitting. Although it seemed that overfitting helped lower the training loss, it was not a good sign because the model could not accurately predict an unseen image.

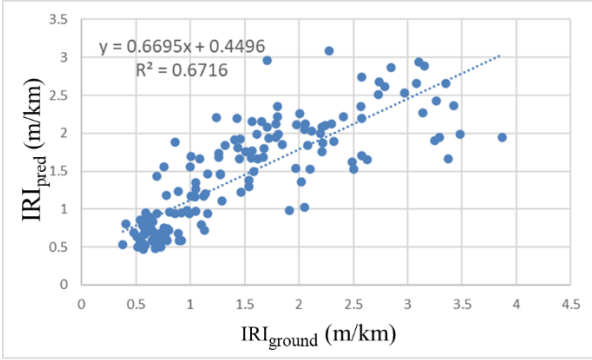
Therefore, to avoid overfitting problems, only the best model with the smallest loss on the validation dataset was grasped and saved during the training processes, but not the one with the lowest training loss or the final one after the 100th epoch. Theoretically, the DNN could still stick to Huber's loss as the criterion to determine the best model, but root mean squared error (RMSE) was intentionally chosen because it is the most employed measurement of accuracy to demonstrate performances in the literature. Therefore, it is more convenient to compare the performances of the proposed model with others by using RMSE. The equation of RMSE is shown in Eq. (21).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (IRI_{ground,i} - IRI_{pred,i})^2}{n}}, \quad (21)$$

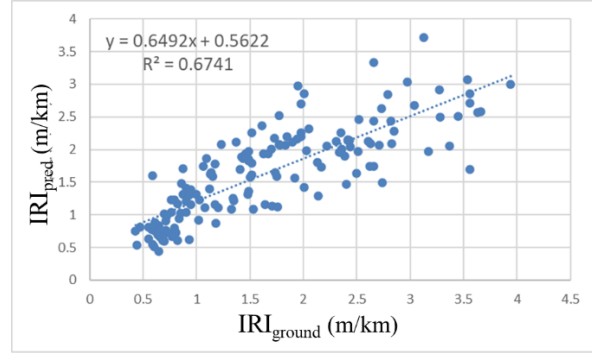
where RMSE is the root mean squared error of a subset, n is the size of the dataset, $IRI_{ground,i}$ is the IRI ground truth of segment i , and $IRI_{pred,i}$ is the IRI prediction of segment i .

The observed lowest RMSE of the left and right wheel path validation dataset were 0.492900 and 0.499238, respectively.

To verify the performances of the proposed models, the trained networks were applied on the testing subsets, and their results were plotted in the form of scatter diagrams, as shown in Figure 5.3. The horizontal axis is the IRI ground truth, and the vertical axis is the predicted IRI. The dashed line and its related equation represent the smallest sum of squared residuals, and the R^2 is the percentage of predicted IRI variation that the linear equation explains. R squared is always between 0 and 1, with a higher R squared means a better fitting between ground truths and predictions. Ideally, if the predictions were 100% accurate, all the points should align perfectly on the line of $y = 1.0 x + 0$, and the corresponding R squared is 1.0.



a) Left wheel path



b) Right wheel path

Figure 5.3. Relationships between IRI ground truths and predictions of testing datasets.

From the above graphs, the R squared of the left and right wheel path testing subsets were 0.6716 and 0.6741, respectively. It indicated that very close performances were achieved with both groups. Besides R squared, the RMSEs of the testing subsets were also computed. It was 0.488861 for the left wheel path and 0.507214 for the right wheel path. The very close RMSEs again confirmed similar performances on the two groups.

5.1.2. Combining Left and Right Wheel Path Datasets

To figure out whether left wheel path and right wheel path images could be mixed and trained together, the datasets of left and right wheel paths were combined, and the same training/testing approaches in section 5.1.1 were reapplied, including the splitting percentages of the dataset, the batch size, and the number of training epochs. The learning curve is shown in Figure 5.4. The corresponding lowest RMSE of validation was 0.465691.

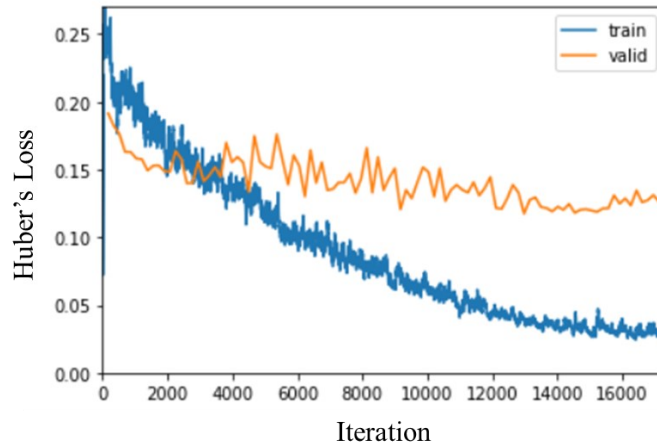


Figure 5.4. Training and validation loss of combined dataset.

The RMSE achieved with the testing subset was 0.536741. The scatter diagram with R squared equal to 0.6111 is shown in Figure 5.5.

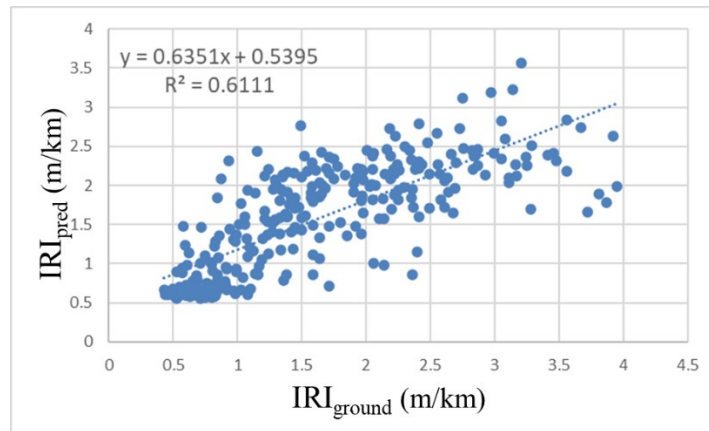


Figure 5.5. Relationships between ground truths and predictions of the combined testing subset.

5.1.3. Intuitive Demonstration of Prediction Results

To demonstrate the testing predictions, three left wheel path images were randomly picked from the testing dataset and then predicted by the trained models. The results are shown in Figure 5.6.

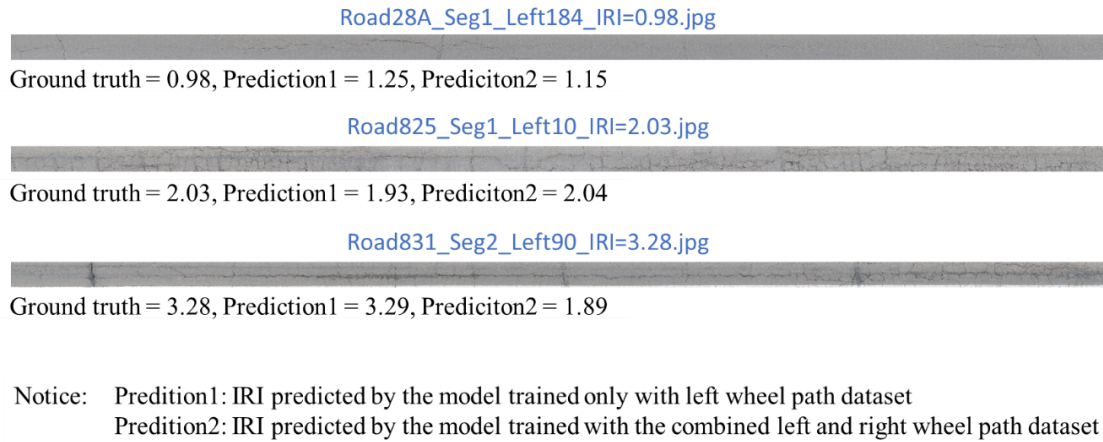


Figure 5.6. Samples of prediction results.

As indicated in the above figure, the predictions of the first two images were relatively good, while the prediction2 of the last image was much lower than the ground truth. The reason could be that the training data of the larger IRI were relatively insufficient compared with the smaller IRI. The other reason might be that prediction2 was given by the model trained on the combined dataset, which was not exclusively designed for the left wheel path.

5.2. Discussions

5.2.1. Comparison between Results of Different Dataset Combinations

In Table 5.1, the testing results of the left and right wheel path datasets (i.e., results from section 5.1.1) are listed in the first rows. In the third row, the averaged values of the above two rows are calculated. Finally, the last row records the testing results of the combined dataset (i.e., section 5.1.2).

Table 5.1. Testing results.

Dataset	RMSE	R squared
Left wheel path dataset	0.488861	0.6716
Right wheel path dataset	0.507214	0.6741
Averaged	0.498037	0.6728
Combined dataset	0.536741	0.6111

Comparing the last two rows shows that the accuracies of the combined dataset were lower than the averaged accuracies in terms of both RMSE and R squared. The RMSE increased by approximately 6.0%, while the R squared dropped nearly 9.0%. So, the assumption that the left and right wheel path datasets can be mixed and trained together was denied. The underlying reason could be that the left and right wheel paths have different characteristics due to their mirror-symmetrical differences, and thus they should not be mixed up and trained together.

5.2.2. Effects of δ in the Loss Function

As stated in section 4.2.3, the loss function employed was the Huber’s loss, and the default δ was 1. This δ is a hyperparameter in the model that determines between which error range (i.e., $|IRI_{ground} - IRI_{pred}| \leq \delta$) should the model be sensitive with. On the other hand, δ defines which points are the outliers that the model should have more tolerance to. Therefore, it is worthy of learning how δ affects the network and what is the optimum value of δ .

To answer this question, controlled experiments were performed on the left wheel path dataset, and the only independent variable was δ . Everything else, including learning rate, number of epochs, training-validation-test splitting ratio, and settings of the optimizer, remained identical. The results are listed in Table 5.2 and plotted in Figure 5.7.

Table 5.2. Results of controlled experiments on δ .

δ	Train RMSE	Test RMSE	Test R ²
0.25	0.4376	0.5129	0.6969
0.5	0.5179	0.6028	0.5569
0.75	0.4904	0.3954	0.7468
1.0	0.4929	0.4888	0.6716
1.25	0.5348	0.5319	0.6372
1.5	0.5273	0.4778	0.6860
2.0	0.5376	0.5272	0.5680
3.0	0.4758	0.6266	0.4753
4.0	0.4740	0.5460	0.7048
8.0	0.5267	0.5949	0.5072

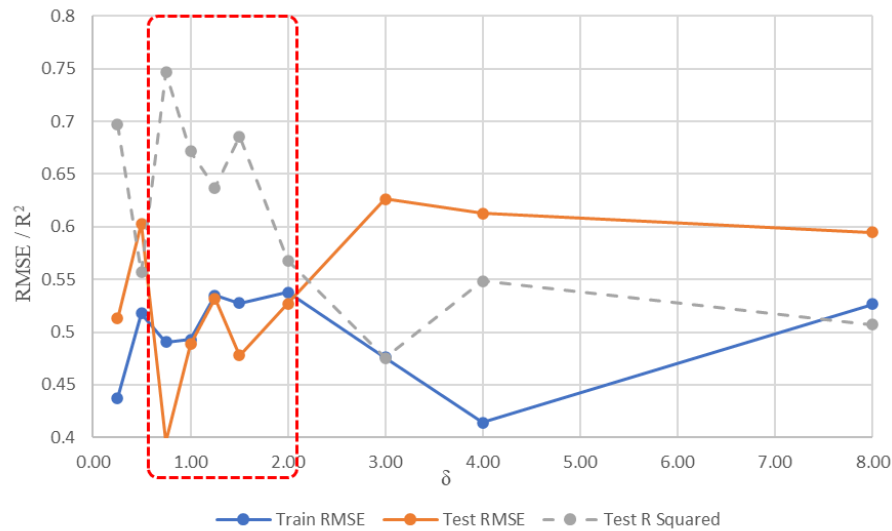


Figure 5.7. Results of controlled experiments on δ .

Comparing training and testing RMSE, it indicates that overfitting has happened if δ is out of the range $[0.75, 2.0]$ (i.e., portions out of the red dashed line box) because testing errors are significantly greater than training errors, though only the best model that has the smallest RMSE on validation dataset was captured. But this phenomenon changes in the range $[0.75, 2.0]$, the testing errors are smaller or very close to training errors. Besides that, the R squared coefficients are greater in this region than the other regions at the two ends, which implies better fittings. Among this region, the lowest error and the best fitting result occur when $\delta = 0.75$. This suggests 0.75 might be the optimum value of δ . To confirm this, the same experiment was performed on

the right wheel path dataset, but the R squared dropped to 0.5631 this time. Therefore, the previous assumption was denied, and $\delta = 0.75$ was not the optimum value. It seemed one was a more stable value for δ . The following discussion will be stuck to the results obtained with $\delta = 1$.

5.2.3. Effects of Batch Sizes

As explained in section 5.1.1, batch size determines the number of samples passing through the network in one iteration. Usually, a larger batch size requires more memory, so there exists a limit on the batch size depending on the GPU memory. In this study, the maximum batch size was 22, and any batch sizes exceeding this limit will crash the training processes. Therefore, controlled experiments were performed with the batch size varying from 2 to 22 with a step of two. All the other hyperparameters remained unchanged, including the number of epochs, learning rate, data splitting ratio, settings of the optimizer, and $\delta = 1$. The below Table 5.3 and Figure 5.8 record the results.

Table 5.3. Results of controlled experiments on batch size.

Batch Size	Train RMSE	Test RMSE	Test R ²
2	0.5365	0.7048	0.3686
4	0.4254	0.6433	0.5093
6	0.5853	0.5846	0.6222
8	0.4929	0.4888	0.6716
10	0.5712	0.4483	0.6696
12	0.4778	0.5605	0.6146
14	0.4753	0.4586	0.7185
16	0.4969	0.4995	0.6546
18	0.4485	0.5395	0.6336
20	0.4297	0.4774	0.6155
22	0.4873	0.4897	0.6532

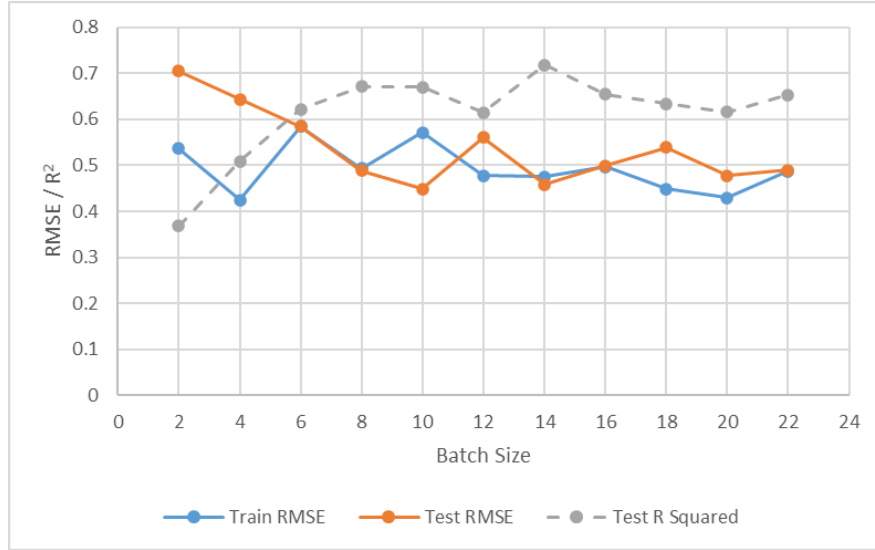


Figure 5.8. Results of controlled experiments on batch size.

The testing errors are significantly higher than training errors when batch sizes are smaller than eight and the R squared values are lower in this region. On the contrary, for the part where batch sizes are greater than eight, the testing and training errors are close and significant better fittings (R squared) are observed. So, it is concluded that the optimum batch size for this application should be equal to or greater than eight.

5.2.4. Comparison With the Other Present-Based Models

To verify the proposed model's performance, comparisons were made with two other latest present-based models published in 2020. However, to the best of the author's knowledge, imaging-based IRI identification models have not been discussed in the literature yet. Thus, the following two models are both based on dynamic responses.

Jeong et al. [70] utilized the dynamic responses from smartphones mounted on anonymous (i.e., calibration-free) vehicles to train their DNN model. Their model identified the IRI for 8.5-m segment intervals. As the testing results, their model achieved extremely high accuracies with the highest R squared of 0.95 and the lowest RMSE of 0.549. As shown, although RMSE (0.498) in this study is smaller, the R squared (0.673) still has room for improvement.

The other model proposed by Mirtabar et al. [71] collected the dynamic responses with self-assembled accelerometer systems. Even though their methodology swerved from DNN to

crowdsourcing technology, comparisons were still made because of its freshness. Rather than 8.5-m intervals above, their model identifies the IRI for varying segment lengths: 10-m, 20-m, 50-m, and 100-m. They found that the R squared increased when segment length extended, which indicated that a better goodness-of-fit could be obtained if the segment length increases. The R squared of the same segment length (50-m) in this thesis was 0.778, which was approximately 15% higher than the reported value in this thesis (0.673). For the RMSE, they only reported for 100-m segment since this longer segment length gave better accuracies. They repeated the testing experiments by three trials, and the average RMSE was 0.41, which was 18% lower than the one reported in this thesis (0.498).

Although the current accuracy of this model is acceptable at the highway network level in the majority of countries, it is apparent that it is far from perfect. However, it still gives a novel direction that imaging-based DNN models can be a possible method to accurately identify the IRI if appropriate improvements can be achieved in the future. The potential approaches include larger datasets, more suitable DNNs, and more precise image stitching methods. In addition, more experiments should be designed and conducted on both vibration and imaging-based models in the future to statistically learn the trade-off relationships between efficiency and accuracy.

6. CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS

6.1. Summary and Conclusions

In this thesis, a novel imaging-based DNN model was proposed and developed to identify the pavement IRI. Despite the construction of the network, the procedures of collecting imaging data and stitching a full-size image were also presented. Although there is no existing literature about imaging-based DNN methods to identify the IRI, comparisons were still made between the proposed model and several other latest models using different methods.

In the data preparation part, the model took advantage of the government-released IRI documents containing the GPS coordinates for each 50-m long segment. On the other hand, the GoPro camera took pavement images in series and meanwhile recorded their GPS coordinates. By matching the GoPro GPS coordinates with the starting/ending points of each 50-m segment, the related images were picked out and then stitched together to get the corresponding full-size image. Additionally, to get a nondistorted and nonoverlapping full-size image, a carefully designed perspective/cropping transformation was performed on each original image. To verify the accuracy of this transformation while avoiding the risks on real highways, an indoor experiment was conducted to simulate a highway lane, and the results implied that the proposed transformation was very accurate. Afterwards, another simple experiment was also conducted, and it successfully verified that the accuracy of the GoPro camera GPS receiver was acceptable. Finally, a sample full-size image was demonstrated to show the good performance of the proposed stitching method.

In the DNN model section, the ResNet34 designed for classification tasks was first demonstrated. After that, the modifications to tune it to accomplish the IRI regression problem along with their reasons were described. Besides the necessary modifications, several latest techniques, which could improve the performances or save labour/computing efforts were introduced. Specifically, those techniques include mixed pooling layers, the Huber's loss functions, the Adam optimization algorithm, the cyclical learning rate strategy, and the transfer learning method.

In the results section, experiments were repeated on different datasets, namely the left wheel path dataset, the right wheel path dataset, and the combined left/right wheel path dataset. In each group of experiments, the same training hyperparameters were employed, including the dataset splitting percentages, the batch size, and the number of training epochs. The accuracy results were reported

in terms of RMSE and R squared values. Comparing the results from separated wheel path datasets and the combined dataset, it was found that it is better to not mix the left and right wheel path datasets together because the separated ones achieved better accuracies. In addition to the numerical results, three full-size left wheel path sample images were intuitively presented along with their IRI predictions by the trained models. In the discussion section, controlled experiments were conducted by keeping all the other hyperparameters the same but changing the δ in Huber's loss functions. The results of varying δ values suggested that the range of [0.75, 2.0] is the optimum region. Besides δ , controlled experiments were also performed to study the other hyperparameter, the batch size. Furthermore, comparisons were made between the proposed model and several other state-of-the-art models that estimate the IRI. The comparisons implied that the proposed model still has room for improvements though it is acceptable at the highway network level.

Although the current model has lower accuracies compared with the other present-based models that use other type of equipment, data, and information, the proposed model has these advantages:

- 1). It is not necessary to design and build special devices to collect imaging data.
- 2). Imaging data is less likely to be affected by the vehicle vibration.
- 3). There are effective tools (e.g., DNNs) to process images.

6.2. Recommendations for Future Work

Although the proposed model is already acceptable at the highway network level to identify the pavement IRI, improvements are necessary if one wants to apply it at segment level one day and obtain similar performance as the current industrial IRI measurement methods [12].

The most important measure is to enlarge the size of the dataset. Constrained by the small data size, the model could not comprehensively learn the characteristics of different levels of IRI between 0 and 4, not to mention those having IRI greater than 4.

Secondly, only ResNet34 was experimented with in this study. However, there are actually many other DNNs that have been proven powerful in different areas. Because the particular objects in this study are long and narrow wheel paths, the height/width ratio of the images is very special

compared with other normal images. So, ResNet34 might not be the perfect DNN to extract feature maps of those long and narrow images. There may be other DNN structures that perform better when dealing with this particular kind of images. Therefore, it is necessary to try more other DNNs to find the potential optimum one.

Thirdly, the GPS coordinates recorded by the GoPro camera are not highly accurate, and it will thus potentially affect the quality of the stitched images. According to the author's knowledge, there exist high precision GPS receivers that can reach ± 1 mm precision requirement and have been used to monitor dam deformations [72]. In future experiments, one can bind this kind of accurate GPS receivers with the GoPro camera to correct the GPS coordinates and improve the quality of the stitched images.

Finally, at this stage, there is no sound evidence that statistically shows the efficiency of using imaging data compared with vibration data. Therefore, in the future, more experiments could be conducted to reveal the trade-off between the efficiency and accuracy of these two different approaches.

REFERENCES

- [1] G. o. Canada, "The Canadian Transportation System," 08 May 2018. [Online]. Available: <https://www144.statcan.gc.ca/tdih-cdit/cts-rtc-eng.htm>. [Accessed 26 September 2021].
- [2] Transport Canada, "Transportation in Canada Overview Report," 2018.
- [3] B. R. Christopher, C. Schwartz and R. Boudreau, *Geotechnical Aspects of Pavements Reference Manual*, Federal Highway Administration, 2006.
- [4] H. Gong, Y. Sun, X. Shu and B. Huang, "Use of random forests regression for predicting IRI of asphalt pavements," *Construction and Building Materials*, vol. 189, pp. 890-897, 2018.
- [5] R. B. Kulkarni and R. W. Miller, "Pavement Management Systems: Past, Present, and Future," *Transportation Research Record*, vol. 1853, no. 1, pp. 65-71, 2003.
- [6] M. W. Sayer, T. D. Gillespie and W. D. O. Paterson, *Guidelines for the Conduct and Calibration of Road Roughness Measurements*, Washington, D.C.: The World Bank, 1986, p. iii.
- [7] M. W. Sayers and S. M. Karamihas, *A little book of Profiling*, University of Michigan, 1998, p. 55.
- [8] M. W. Sayers, T. D. Gillespie and C. A. V. Queiroz, *The International Road Roughness Experiment: Establishing Correlation and a Calibration Standard for Measurements*, Washington: The World Bank, 1986.
- [9] A. Moh and J. Roy, "Uses & Comparison of IRI with other Jurisdictions (Asphaltic Concrete Pavements)," Edmonton, Alberta Infrastructure, 2000, p. 3.
- [10] Infrastructure and Transportation, "Guidelines for assessing pavement preservation treatments and strategies: edition 2," Edmonton, Infrastructure and Transportation, 2006, p. 4.

- [11] R. Jurgens and J. Chan, "Highway Performance Measures for Business Plans in Alberta," in *Session of the 2005 Annual Conference of the Transportation Association of Canada*, Calgary, 2005.
- [12] Tetra Tech, "Pavement Asset Management Program," 2015.
- [13] M. W. Sayers, "On the Calculation of International Roughness Index from Longitudinal Road Profile," *Transportation Research Record*, no. 1501, 1995.
- [14] TRB, National Research Council, "NCHRP Report 228: Calibration of Response-Type Road Roughness Measuring Systems," Washington, D.C., 1980.
- [15] Y. Du, C. Liu, D. Wu and S. Jiang, "Measurement of International Roughness Index by Using Z-Axis Accelerometers and GPS," *Mathematical Problems in Engineering*, vol. 2014, 2014.
- [16] Federal Highway Administration, *The Long-Term Pavement Performance Program*, McLean: FHWA, 2017.
- [17] FHWA, "Our Nation's Highways: 2011," 7th November 2014. [Online]. Available: <https://www.fhwa.dot.gov/policyinformation/pubs/hf/pl11028/chapter1.cfm>. [Accessed 17 October 2021].
- [18] Z. Zhang, Z. Zhang, R. Bridgelall and M. Sun, "Application of a Machine Learning Method to Evaluate Road Roughness from Connected Vehicles," *Journal of Transportation Engineering*, vol. 144, no. 4, 2018.
- [19] Q. Mei and M. Gül, "A crowdsourcing-based methodology using smartphones for bridge health monitoring," *Structural Health Monitoring*, vol. 18, no. 5-6, pp. 1602-1619, 2019.
- [20] Q. Mei, M. Gül and N. Shirzad-Ghaleoudkhani, "Towards smart cities: crowdsensing-based monitoring of transportation infrastructure using in-traffic vehicles," *Journal of Civil Structural Health Monitoring*, vol. 10, p. 653–665, 2020.

- [21] Q. Mei, M. Gül and N. Shirzad-Ghaleroudkhani, "Densely connected deep neural network considering connectivity of pixels for automatic crack detection," *Automation in Construction*, vol. 110, p. 103018, 2020.
- [22] Q. Mei and M. Gül, "A cost effective solution for pavement crack inspection using cameras and deep neural networks," *Construction and Building Materials*, vol. 256, p. 119397, 2020.
- [23] S.-C. Wang, "Artificial Neural Network," in *Interdisciplinary Computing in Java Programming*, vol. 743, Boston, MA: The Springer International Series in Engineering and Computer Science, 2003, p. 83.
- [24] R. Ali, J. Zeng and Y.-J. Cha, "Deep learning-based crack detection in a concrete tunnel structure using multispectral dynamic imaging," in *SPIE Smart Structures + Nondestructive Evaluation*, 2020.
- [25] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a Convolutional Neural Network," in *International Conference on Engineering and Technology (ICET)*, Antalya, 2017.
- [26] J. Du and Y. Xu, "Hierarchical deep neural network for multivariate regression," *Pattern Recognition*, vol. 63, pp. 149-157, 2017.
- [27] P. C. S. Reddy, K. Sarma, A. Sharma, P. V. Rao, S. Rao, G.R.Sakthidharan and K.Kavitha, "Enhanced age prediction and gender classification (EAP-GC) framework using regression and SVM techniques," *Materials Today: Proceedings*, 2020.
- [28] O. Poursaeed, T. Matera and S. Belongie, "Vision-based real estate price estimation," *Machine Vision and Applications*, vol. 29, no. 4, pp. 667-676, 2018.
- [29] P. Karir, N. Goel and V. K. Garg, "Human age prediction using DNA methylation and regression methods," *International Journal of Information Technology*, vol. 12, no. 2, p. 373–381, 2020.

- [30] Y. Wei, Y. Zhang, J. Huang and Q. Yang, "Transfer Learning via Learning to Transfer," in *35th International Conference on Machine*, Stockholm, 2018.
- [31] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama and H. Omata, "Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone," *Computer Aided Civil and Infrastructure Engineering*, vol. 33, no. 13, pp. 1127-1141, 2018.
- [32] Q. Mei and M. Gul, "A cost effective solution for pavement crack inspection using cameras and deep neural networks," *Construction and Building Materials*, vol. 256, 2020.
- [33] N. Karballaezadeh, F. Zaremotekhasas, S. Shamshirband, A. Mosavi, N. Nabipour, P. Csiba and A. R. Várkonyi-Kóczy, "Intelligent Road Inspection with Advanced Machine Learning; Hybrid Prediction Models for Smart Mobility and Transportation Maintenance Systems," *Energies*, vol. 13, no. 7, 2020.
- [34] J.-D. Lin, J.-T. Yau and L.-H. Hsiao, "Correlation Analysis Between International Roughness Index (IRI) and Pavement Distress by Neural Network," in *82nd Annual Meeting of the Transportation Research Board*, 2003.
- [35] R. A. El-Hakim and S. El-Badawy, "International Roughness Index Prediction for Rigid Pavements: An Artificial Neural Network Application," *Advanced Materials Research*, vol. 723, pp. 854-860, 2013.
- [36] W. Zeiada, S. A. Dabous, K. Hamad, R. Al-Ruzouq and M. A. Khalil, "Machine Learning for Pavement Performance Modelling in Warm Climate Regions," *Arabian Journal for Science and Engineering*, vol. 45, pp. 4091-4109, 2020.
- [37] M. Fakhri, S. M. Karimi and J. Barzegaran, "Predicting International Roughness Index Based on Surface Distresses in Various Climate and Traffic Conditions Using Laser Crack Measurement System," *Transportation Research Record*, 2021.

- [38] M. Hossain, L. S. P. Gopiseti and M. S. Miah, "Artificial neural network modelling to predict international roughness index of rigid pavements," *International Journal of Pavement Research and Technology*, vol. 13, pp. 229-239, 2020.
- [39] P. Georgiou, C. Plati and A. Loizos, "Soft Computing Models to Predict Pavement Roughness: A Comparative Study," *Hindawi Advances in Civil Engineering*, vol. 2018, 2018.
- [40] P. Marcelino, M. d. L. Antunes, E. Fortunato and M. C. Gomes, "Transfer learning for pavement performance prediction," *International Journal of Pavement Research and Technology*, vol. 13, pp. 154-167, 2020.
- [41] J. Marcondes, G. J. Burgess, R. Harichandran and M. B. Snyder, "Spectral analysis of highway pavement roughness," *Journal of Transportation Engineering*, vol. 117, no. 5, pp. 540-549, 1991.
- [42] International Organization for Standardization, "ISO 8608:1995 Mechanical vibration — Road surface profiles — Reporting of measured data," 1995.
- [43] C. Wong and K. Worden, "Generalised NARX shunting neural network modelling of friction," *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 553-572, 2005.
- [44] H. Ngwangwa, P. Heyns, F. Labuschagne and G. Kululanga, "Reconstruction of road defects and road roughness classification using vehicle responses with artificial neural networks simulation," *Journal of Terramechanics*, vol. 47, no. 2, pp. 97-111, 2010.
- [45] Y. Qin, C. Xiang, Z. Wang and M. Dong, "Road excitation classification for semi-active suspension system based on system response," *Journal of Vibration and Control*, vol. 24, no. 13, p. 2732–2748, 2018.
- [46] V. M.A.Souza, RafaelGiusti and A. J.L.Batista, "Asfalt: A low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning," *Pervasive and Mobile Computing*, vol. 51, pp. 121-137, 2018.

- [47] Alberta Transportation, Contract Administration Manual for Highway and Bridge Maintenance, Government of Alberta, 2010.
- [48] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials*, vol. 157, pp. 322-330, 2017.
- [49] A. Kumar, Chakrapani, D. J. Kalita and V. P. Singh, "A Modern Pothole Detection technique using Deep Learning," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, Bhopal, 2020.
- [50] S. Milhomem, T. d. S. Almeida, W. G. d. Silva, E. M. d. Silva and R. L. d. Carvalho, "Weightless Neural Network with Transfer Learning to Detect Distress in Asphalt," *International Journal of Advanced Engineering Research and Science*, vol. 5, no. 12, pp. 294-299, 2018.
- [51] N. S. P. Peraka, K. P. Biligiri and S. N. Kalidindi, "Development of a Multi-Distress Detection System for Asphalt Pavements: Transfer Learning-Based Approach," *Transportation Research Record*, pp. 1-16, 2021.
- [52] E. Ibragimov, H.-J. Lee, J.-J. Lee and N. Kim, "Automated pavement distress detection using region based convolutional neural networks," *International Journal of Pavement Engineering*, 2020.
- [53] S. Bang, S. Park, H. Kim, Y.-s. Yoon and H. Kim, "A deep residual network with transfer learning for pixel-level road crack detection," in *35th International Symposium on Automation and Robotics in Construction*, 2018.
- [54] Government of Alberta, "International Roughness Index and Rut Data," Edmonton, 2020.
- [55] Government of Alberta, "Provincial Construction Program," Edmonton.
- [56] "Google Maps," Google, [Online]. Available: <https://www.google.com/maps>. [Accessed 26 August 2021].

- [57] "HERO7 Field of View (FOV) Information," GoPro, [Online]. Available: https://gopro.com/help/articles/question_answer/hero7-field-of-view-fov-information?sf96748270=1. [Accessed 26 August 2021].
- [58] J.D. Power, 2021. [Online]. Available: <https://www.nadaguides.com/cars/2016/bmw/3-series/Sedan-4D-328i-I4-Turbo/Pictures/Print>. [Accessed 15 July 2021].
- [59] Movable Type Scripts, "Calculate distance, bearing and more between Latitude/Longitude points," [Online]. Available: <https://www.movable-type.co.uk/scripts/latlong.html>. [Accessed 10 11 2021].
- [60] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778, 2016.
- [61] D. Yu, H. Wang, P. Chen and Z. Wei, "Mixed Pooling for Convolutional Neural Networks," in *International Conference on Rough Sets and Knowledge Technology*, 2014.
- [62] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *The 3rd International Conference for Learning Representations*, San Diego, 2014.
- [63] S. Ruder, "An overview of gradient descent optimization algorithms," 2016.
- [64] J. Duchi, E. Hazan and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 2121-2159, 2011.
- [65] G. Hinton, N. Srivastava and K. Swersky, *rmsprop: Divide the gradient by a running average of its recent magnitude*.
- [66] L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," 2015.
- [67] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga and A. Lerer, "Automatic differentiation in PyTorch," in *31st Conference on Neural Information Processing Systems*, Long Beach, 2017.

- [68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009.
- [69] Alberta Government, "Physical Condition of Provincial Highway Surfaces, Alberta," 26 June 2018. [Online]. Available: <https://open.alberta.ca/opendata/physical-condition-of-provincial-highway-surfaces-alberta>. [Accessed 07 11 2021].
- [70] J.-H. Jeong, H. Jo and G. Ditzler, "Convolutional neural networks for pavement roughness assessment using calibration-free vehicle dynamics," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 11, p. 1209–1229, 2020.
- [71] Z. Mirtabar, A. Golroo, A. Mahmoudzadeh and F. Barazandeh, "Development of a crowdsourcing-based system for computing the international roughness index," *International Journal of Pavement Engineering*, 2020.
- [72] W. Li and C. Wang, "GPS in the tailings dam deformation monitoring," *Procedia Engineering*, vol. 26, pp. 1648-1657, 2011.