



# A computer vision-based method to identify the international roughness index of highway pavements

Jiangyu Zeng, Mustafa Gül, Qipei Mei \*

Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Alberta, T6G 2W2, Canada



## ARTICLE INFO

### Keywords:

International roughness index  
Deep neural network  
Computer vision  
Pavement condition assessment

## ABSTRACT

The International Roughness Index (IRI) is one of the most critical parameters in the field of pavement performance management. Traditional methods for the measurement of IRI rely on expensive instrumented vehicles and well-trained professionals. The equipment and labor costs of traditional measurement methods limit the timely updates of IRI on the pavements. In this article, a novel imaging-based Deep Neural Network (DNN) model, which can use pavement photos to directly identify the IRI values, is proposed. This model proved that it is possible to use 2-dimensional (2D) images to identify the IRI other than the typically used vertical accelerations or 3-dimensional (3D) images. Due to the fast growth 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. These significant improvements make it not only more convenient to collect high-quality 2D images, but also easier to process them than vibrations or accelerations. In the proposed method, 15% of the imaging data were randomly selected for testing and had never been touched during the training steps. The testing results showed an averaged coefficient of determination (R square) of 0.6728 and an averaged root mean square error (RMSE) of 0.50.

## 1. Introduction

Pavement surface roughness is a critical factor that substantially dominates the highway system's driving experience and transportation safety. Many researchers and engineers dedicated enormous time and efforts to standardizing the indicator of pavement roughness throughout the world. Various indices, including Ride Index (AAA & GTH, 1981), Present Serviceability Index (Highway Research Board, 1962), and Pavement Condition Index (PCI) (Shahin et al., 1977), have been proposed in the last century. However, such indicators are not replicable across agencies or even within the same agency but at a different time period (Sayer et al., 1986). In 1986, the World Bank established the IRI (Sayers et al., 1986), and it soon became the most commonly used pavement roughness indicator worldwide due to its high reproducibility.

The IRI can be measured in either direct or indirect ways (Huang, 2004). Direct methods include traditional longitudinal surveys by rod and level or advanced laser-type profilometers. Indirect methods measure longitudinal profile by response-type road roughness meters. In Alberta, Canada, the European Banking Authority (EBA) Company runs the measurement of IRI for the provincial government by a direct method. EBA Company scans longitudinal profile for every 50-m interval

using their special vehicle mounted with 11 laser sensors (Tehrani, 2014). Then, the system inputs the profile into a quarter-car model algorithm established by Sayers and Karamihas (Sayers and Karamihas, 1998) to get its IRI value. This mathematical model calculates the simulated suspension motion on the scanned profile, accumulates it, and finally divides it by the travelling distance to get the average suspension motion per unit length. Therefore, the units of IRI are m/km or mm/m. IRI scale is linearly proportional to the surface roughness (Sayers and Karamihas, 1998), and a larger IRI indicates a rougher road surface.

As the key to 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. (2018) confirmed that factors like transverse cracks, fatigue cracks, rutting, block cracks, raveling, and potholes influence IRI, in the sequences from the highest importance score to the lowest. In other words, transverse cracks have the most impact. These mentioned factors have a shared terminology named pavement distresses in the field of pavement management.

Alberta Infrastructure has adapted the IRI as its pavement performance measurement since 1998 (Moh and Roy, 2000) and applied it in

\* Corresponding author.

E-mail address: [qipei.mei@ualberta.ca](mailto:qipei.mei@ualberta.ca) (Q. Mei).

two main aspects: 1) the trigger value for rehabilitation; 2) the pavement rating standards. To improve the pavement management efficiency and serve the public with better transportation infrastructure, the transportation department, as a government body, must make more methodical and effective decisions on whether the roads require repair, rehabilitation, or routine maintenance. Thus, the IRI is employed as a trigger threshold between preventative maintenance and preservative repair. That is to say, preventative maintenance is required if the IRI value 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 in Alberta are listed in Table 1 concerning varying traffic levels.

To make the highway conditions transparent and clear to the public, Alberta Infrastructure (Moh and Roy, 2000) employs the IRI again as a standard to rate highway conditions following the criteria proposed by Jurgens and Chan (Jurgens and Chan, 2005) in 2005. The detailed criteria are shown in Table 2.

Considering these two critical functions that the IRI serves, the importance of IRI measurement is obvious. However, the cost of the current laser road surface tester is high in the aspect of operation and maintenance (Du et al., 2014). Therefore, researchers turned to finding low-cost alternative methods to identify the IRI, and most of them have chosen DNN-based models (El-Hakim and El-Badawy, 2013; Zeiada et al., 2020; Hossain et al., 2020; Ngwangwa et al., 2010; Qin et al., 2018).

Artificial Neural Networks (ANNs) are inspired by biological neural networks and are one of the most popular Artificial Intelligence (AI) methods. ANNs are widely known because of their excellent performance in image recognition and classification (Wang, 2003). ANNs treat input data as a bunch of numbers and tune the weights/biases tactically to achieve the optimum model. Generally, the deeper the network is, the more trainable parameters it has and thus the more information it can learn from the given data. Therefore, 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 (Ali et al., 2020). 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 (Albawi et al., 2017). In fact, to some extent, CNNs make it possible to solve some complex image recognition tasks that could never be accomplished by classic ANNs (Albawi et al., 2017). CNNs have been extensively used in studies regarding pavement management (Ali et al., 2020; Mei and Gul, 2020; Karballaezadeh et al., 2020; Maeda et al., 2018).

Despite the incredible power of DCNNs, it is not always easy or practical to train a brand new one 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 (Wei et al., 2018). Transfer learning is inspired by the phenomenon that human beings can transfer knowledge across different tasks (Wei et al., 2018). It is a helpful technique that allows developers to improve the performance and reduce the number of labelled data by transferring knowledge from a source domain to a target domain (Wei et al., 2018).

The existing predictive models using ANNs can be classified into two categories: past-based and 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 (Federal Highway Administration, 2017) run by the Federal Highway Administration (FHWA). The LTPP program was initiated by the Strategic Highway Research Program (SHRP) in 1987 to extend pavement service life through a series of professional and systematic investigations. Ten of Canada's provincial highway agencies joined this program, including Alberta Transportation. The LTPP collects performance data, including IRI values, climates, traffic, deflection, distress, structure, maintenance, rehabilitation, etc. In LTPP's 2017 report (Federal Highway Administration, 2017), they proclaimed that 2509 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) (FHWA, 2011), not to mention the total highway mileage around the world. For 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 collected on-site at once. The most common inputs are vertical vibrations or accelerations of the driving probe vehicles, measured by either specially designed sensors or some commercial sensors including smartphones (Ngwangwa et al., 2010; Wong and Worden, 2005; Zhang et al., 2018; Souza et al., 2018). In spite of the lower accuracy in comparison to past-based models, present-based models 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.

However, compared with the vibration data, imaging data are more convenient to collect and store and also be less likely to be affected by the vibration of the vehicle. In addition, the proposed method can collect data at the free-flow traffic which means that no traffic control is required. In comparison, using acceleration data, one has to not only maintain a constant speeding during every measurement, but also carefully calibrate the results when the speed varies (Zhang et al., 2018; Mirtabar et al., 2022; Zhao et al., 2017; Nagayama et al., 2013). Although various DNN-based models to detect, classify, and quantify distresses using 2D images (Ali et al., 2020; Gopalakrishnan et al., 2017; Kumar et al., 2020; Milhomem et al., 2018), there is currently no model that directly uses 2D pavement images as the input to estimate the IRI. Additionally, DNN models using 3D images to evaluate the pavement roughness (Abohamer et al., 2021; Tong et al., 2018), but the cost of the required equipment is high (Ragnoli et al., 2018). Therefore, the model presented in this paper aims to be the first attempt to use 2D images and DNNs to directly predict IRI. To accomplish this objective, there is a necessary pre-processing step to find an automatic method to develop full-size orthorectified pavement images. The images must be orthorectified because the sizes of distresses are significant when quantifying the distresses, whereas it is difficult to measure their actual dimensions in images taken from oblique angles. So, the processes of preparing the 50-m long pavement images are also presented.

The paper is organized as follows: Section 2 describes the procedures and methods of ground truth collection and imaging data preparation. Section 3 demonstrates the structure and training processes of the DNN models. In Section 4 and Section 5, analyses and discussions of the testing

**Table 1**

IRI trigger thresholds categorized by Annual Average Daily Traffic (AADT) (Edmonton, 2006).

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

**Table 2**

Highway pavement condition rating criteria (Jurgens and Chan, 2005).

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

results are presented. Finally, Section 6 concludes the article and lists the potential improvements in the future.

## 2. Data collection and preparation

To train an imaging-based DNN for IRI identification, the images for pavement segments and their corresponding IRI ground truths are required. Therefore, the first step is to get the IRI ground truths. Herein, the IRI values measured by the EBA company for Alberta, Canada are taken as the ground truths. The second step is to collect the images of the corresponding pavement segments. During this step, a series of image operations are required to obtain a full-size orthorectified satellite view image of each 50-m pavement segment. 50-m intervals are used to merely match the current government-released IRI data.

### 2.1. IRI ground truths

Alberta Transportation annually posts the IRI data in Excel spreadsheets (Government of Alberta, 2020). Information including road numbers, lane codes, control section numbers, Global Positioning System (GPS) coordinates, and IRI values are accessible to the public. For each 50-m segment, the latitudinal and longitudinal coordinates of the starting and ending points are recorded along with its IRI values. The spreadsheets record the inside (left-wheel path) and outside (right-wheel path) IRI values for each segment. is a sample of how the information is organized in the spreadsheets.

Table 3 is a sample of how the information is organized in the spreadsheets.

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

Before using the IRI datasheets, those road sections where repair and/or rehabilitation were conducted after the IRI measuring dates, were filtered out by comparing them with the 2020/2021 Provincial Construction Program (Government of Alberta) released by the Government of Alberta. The remaining road sections had not undergone any changes after the last time that EBA Company measured the IRI values. The road

**Table 3**  
Sample of international roughness index and rut data (Government of Alberta, 2020).

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

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 their distributions are listed in Appendix A.

### 2.2. Pavement surface images

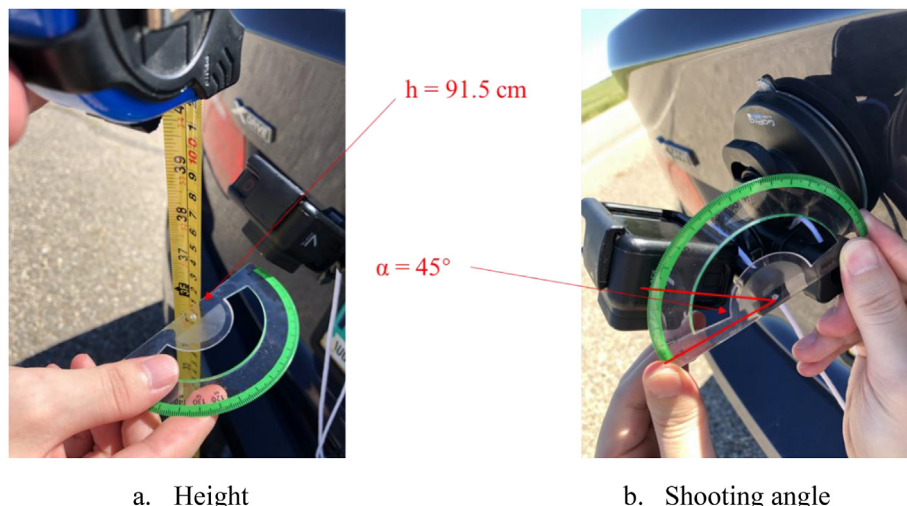
#### 2.2.1. Camera setup

To collect imaging data more efficiently, videos were first filmed in linear mode (i.e., fish-eye effect disabled) 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 Fig. 1. This GoPro Hero 7 Black camera can capture high-resolution videos of  $1920 \times 1080$  pixels at a frame rate of 120 fps. The mounting height is denoted as  $h = 91.5$  cm, and the camera angle is  $\alpha = 45^\circ$ , as indicated.

The cruise speed was 100 km/h, but 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 lanes or overtake, but must stay within the investigated lane to avoid missing any data. But these deviations had no impact when stitching the full-size images because the stitching processes relied only on the GPS coordinates rather than the speeds (see the following sections for more details).

#### 2.2.2. Overview of image preparation procedures

Fig. 2 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 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 the sizes and quantities of the 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 numbers of pixels in the image due to perspective distortion, with the one farther from the lens taking fewer pixels. This



**Fig. 1.** GoPro Hero 7 Black mounted at the rear of BMW 328i xDrive.

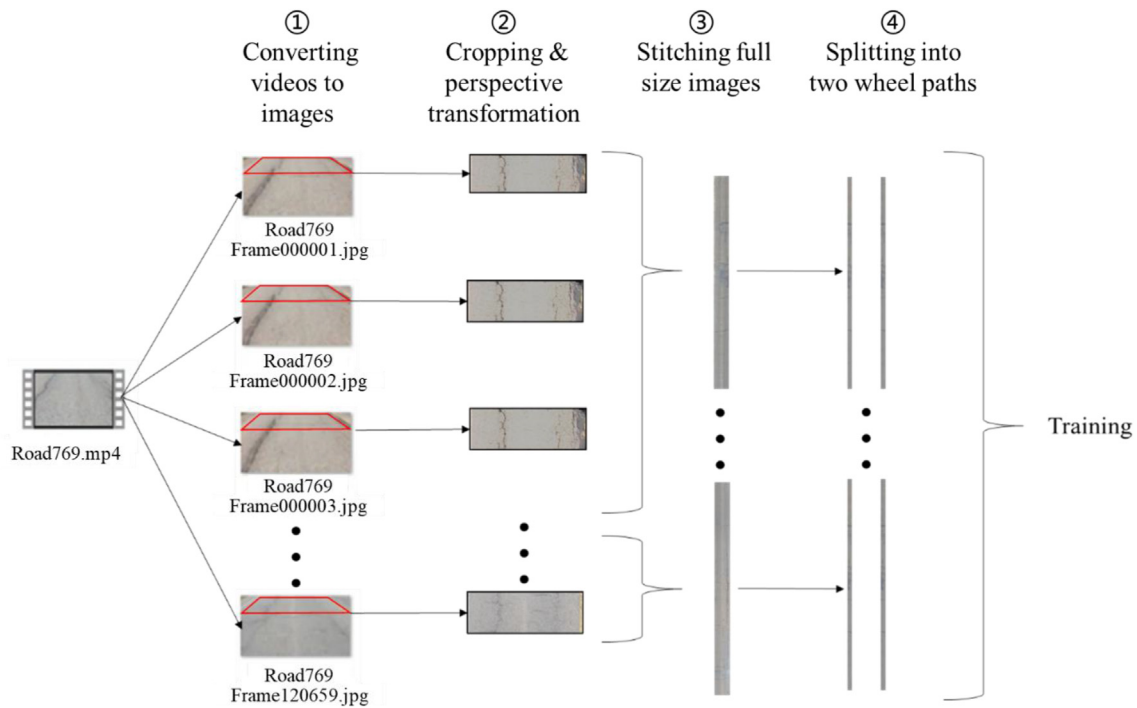


Fig. 2. Schematic procedures of the imaging data preparation.

effect could greatly influence IRI identification. Therefore, the calculated perspective transformations are performed on those images taken with the 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 because the resolution at the far end is too low to process, so several sorted images are stitched together in the designed sequences to compose a completed full-size image in step 3. Since the IRI values are divided into the left wheel path and right wheel path IRIs 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 images of the left and the right wheel paths are ready to get fed into the neural network. The detailed explanations for the image preparation are illustrated in the following sections.

2.2.3. Perspective transformation

The side view of the experimental setup is illustrated in Fig. 3. The vertical field of view (FOV) of this camera is 55.2° (Anonymous). 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 in a line crossing point C to simplify the calculations.

By performing some simple geometrical calculations, we can obtain the Longitudinal Filming Range (LFR) indicated in the above figure:

$$\begin{aligned}
 AD &= h \cdot \tan(90^\circ - \alpha + FOV/2) = 291.991cm \\
 CD &= h \cdot \tan(90^\circ - \alpha - FOV/2) = 28.674cm \\
 LFR &= AC = AD - CD = 263.303cm
 \end{aligned}
 \tag{1}$$

Fig. 4 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 by the dashed blue triangles.

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 Fig. 5, the actual ratio between QR and RS is needed. In addition, it is

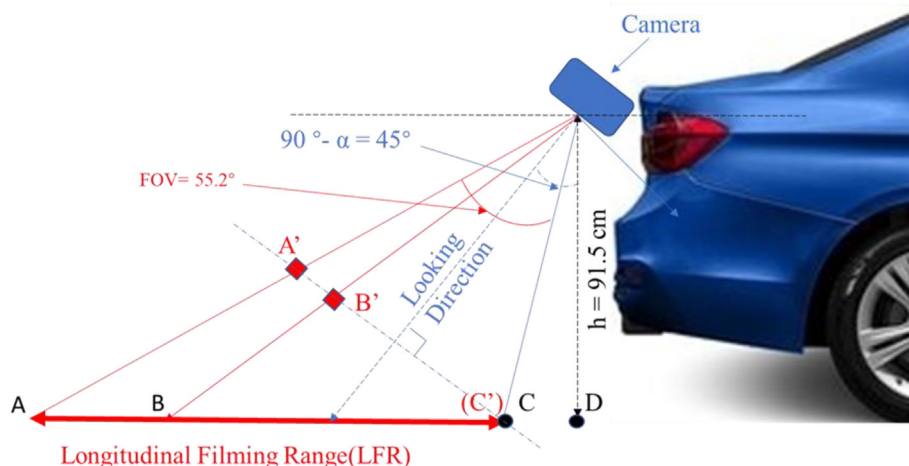


Fig. 3. Side view of the camera setup (modified from (Power, 2021)).

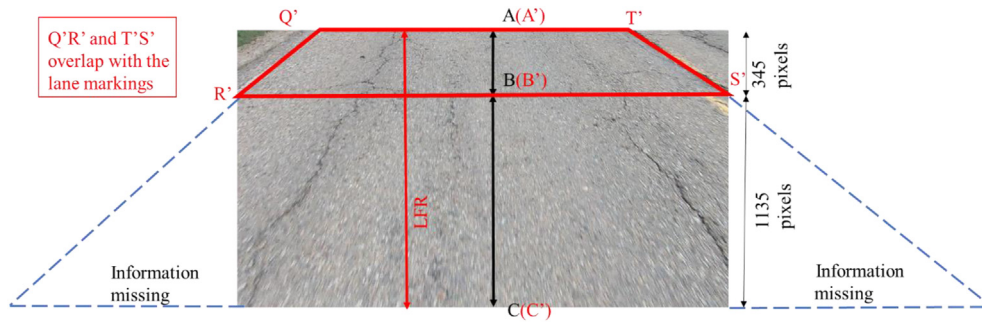


Fig. 4. Sample image.

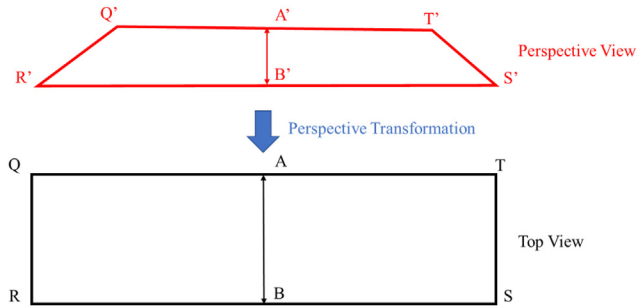


Fig. 5. Perspective transformation.

known that the lane width (length of RS) of highways in Alberta is 3.5 m (350 cm). Thus, the problem turns to calculating the length of QR or AB on the actual pavement surface.

By doing some geometrical calculations we can compute the length of AB is equal to 129.9 cm. The procedures are presented in Appendix B.

Thus, the aspect ratio of the transformed rectangle QRST is 129.9/350. A proportional resolution of 325 × 875 pixels is appropriate for the rectangle QRST in every image. Fig. 6 shows a sample that demonstrates the performance of the proposed perspective transformation.

2.2.4. Verification of perspective transformation

To verify the theoretical computations in the last section and confirm that the proposed perspective transformation is acceptable, it is necessary to measure the length of AB in situ. However, considering safety risks on actual highway pavement, an indoor experiment was conducted to simulate a highway lane, and the configuration is shown in Fig. 7. The camera is mounted on the wall at 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.

The dimensions of the rectangle and its distance from the wall (i.e., simulated car tail) were measured onsite. The calculated and measured results are compared in Fig. 8. Compared with the measured 120 cm, the

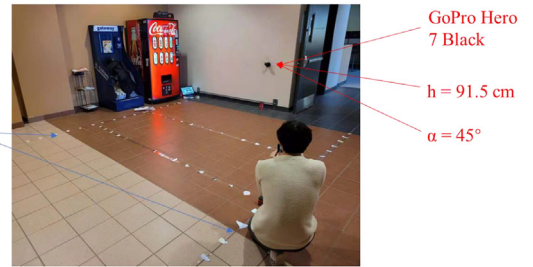


Fig. 7. Scene simulation of a highway lane.

calculated 129 cm is very close and acceptable. One of the possible sources of error could be human errors such as that the camera angle was not exactly set up to 45°.

2.2.5. Image stitching

The 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 4.

The first column is the date and the time that the video was recorded, where “T” is 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 was set to 120, relative time was multiplied by 120 fps to get the frame number in the last column.

Recalling the IRI ground truth sheets (Table 3), 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 could 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. The interpolation procedures can be described in the following equation 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 sheets:

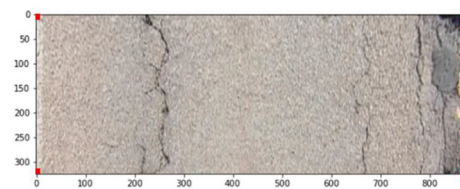
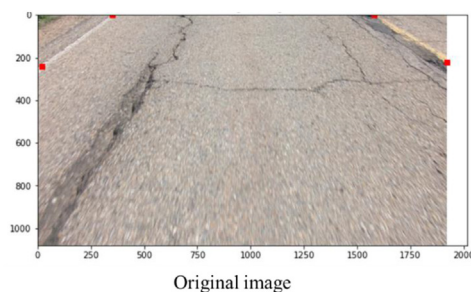


Fig. 6. Sample of calculated perspective transformation.

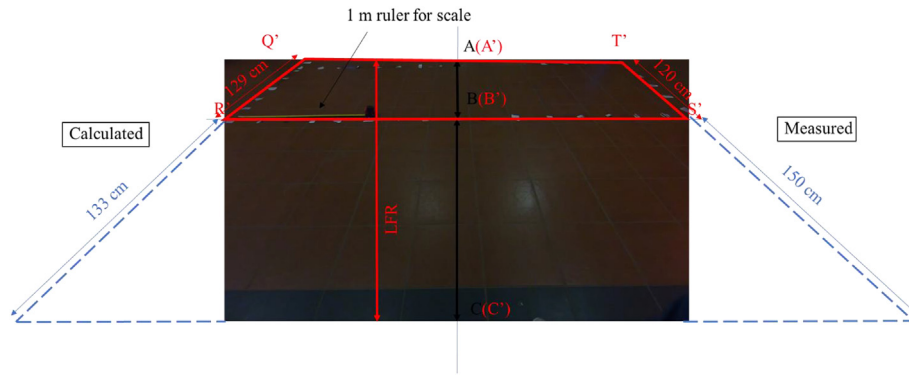


Fig. 8. Comparison between theoretical and measured AB length.

**Table 4**  
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

$$Frame_i = \text{int} \left[ Frame_{high} + \frac{Lat_i - Lat_{high}}{Lat_{low} - Lat_{high}} \cdot (Frame_{low} - Frame_{high}) \right] \quad (2)$$

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  $Lat_{high}$ .

An interpolation example can be found in Appendix C.

Using the same method, the ending frame numbers can be obtained as well. After knowing the starting and ending frame number of each 50-m (5000-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  $M$ , each frame will contribute a small segment length of  $l_{cm}$  if assuming the speed is constant within the interesting segment. As indicated in Fig. 9,  $l_{cm}$  it can then be calculated as:

$$l_{cm} = \frac{5,000cm}{(M - N + 1)} \quad (3)$$

The typical length of  $l_{cm}$  is 25 cm, and it takes approximately 200 frames to compose a full-size lane image.

Using the proportionality between the actual rectangle size (129 × 350 cm) and the image pixels (325 × 875 pixels) as indicated in the

transformed image in Fig. 10, the desired length in the unit of pixels,  $l_{pixel}$ , can be written as:

$$l_{pixel} = \frac{l_{cm}}{129cm} \cdot 325pixels = \frac{5,000cm}{129cm(M - N + 1)} \cdot 325pixels \quad (4)$$

As indicated in the original image in Fig. 10, within the trapezoid Q'R'S'T which covers the full lane width, the bottom portion surrounded in the green box is preferred because it is closer to the lens and thus has undergone a minor perspective transformation. Stitching all the transformed frames together, the resolution of the stitched full-size image is 12,500 × 875 pixels which are proportional to 50 × 3.5 m (5000 × 350 cm) on an actual 50-m long lane segment. But due to insufficient computer memory, the original resolution of 12,500 × 875 pixels is too large to experiment with. Therefore, the images are further resized to 6250 × 438 pixels, but the height/width ratio is unchanged.

To demonstrate the performance of the proposed stitching method, Fig. 11 shows one of the stitched full-size lane segment images and the zooming-in details.

However, to be consistent with a pair of IRI values for the left and right wheel paths, it is necessary to further crop the stitched images into

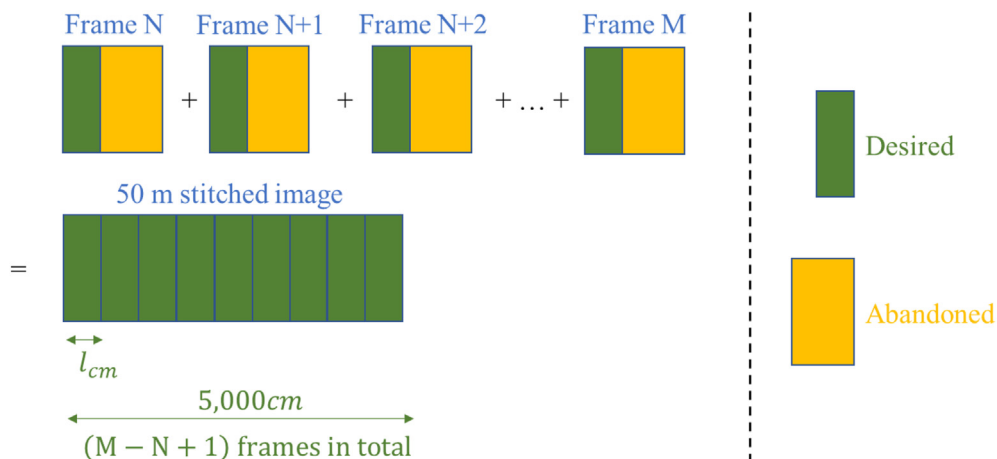


Fig. 9. Stitching transformed frames to get a completed image for one segment.

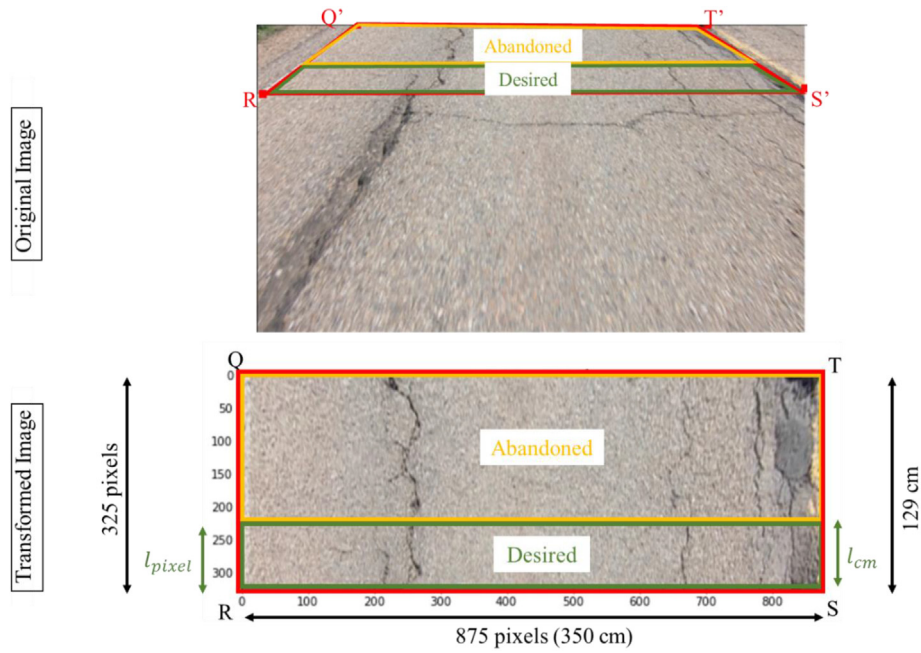


Fig. 10. Sample of how a transformed frame is cropped.

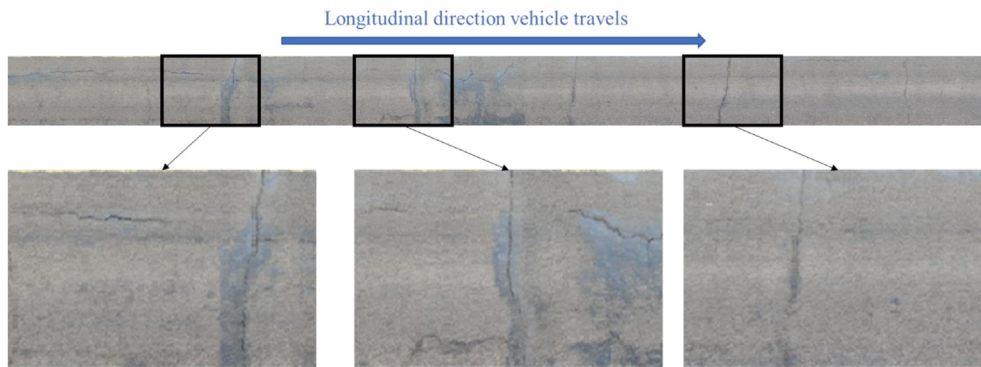


Fig. 11. A sample of the stitched full-size image composed of 200 frames (rotated 90°).

the left and right wheel path images. However, it is almost impossible to control the vehicle at the perfect middle of the lane at all times. Thus, the cropped images must cover widths wider than the wheel width to consider the vehicle's sway. Therefore, two image strips of  $6250 \times 145$  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 that all the useful information is included. In total, 2128 images were collected, half of them representing the left wheel path segments and the other half representing the right wheel path segments.

### 2.2.6. 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 utilizing Google Maps was designed to check the GoPro GPS accuracy. Before explaining the verification experiments, the authors want to specify the precondition revealed by Al-Gaadi (2005) that the error of GPS measurement is not proportional to the vehicle's speed. Therefore, we can perform the verification statically rather than dynamically in order to avoid safety issues. The experiment was carried out in the Varsity Field on campus at the University of Alberta. Firstly, a noticeable mark that is visible on the satellite layer in Google Maps was selected, and its coordinates (Point 1) were recorded. Secondly, an image was taken at this point with the GoPro

camera and the coordinates (Point 2) were recorded too. Finally, the distances between these two points were computed using a website application called Movable Type Scripts (Movable Type Scripts), which can not only calculate the distance between two sets of GPS coordinates but also show them on Google Maps, as indicated in Fig. 12. The described steps were repeated three more times, and the results are listed in Table 5.

As indicated in the above table, the average error of the GPS receiver in the GoPro camera was 0.7685 m. Fig. 13 draws red circles to represent the possible erroneous camera locations.

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 of all the red circles is the possible erroneous path of the camera. In the figure, it is the area filled with solid grey color. As indicated, 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.

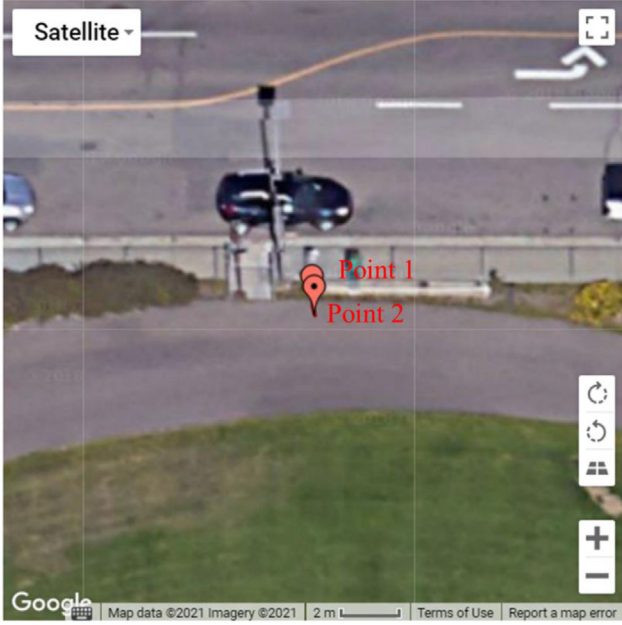


Fig. 12. Screenshot of the movable type scripts (modified from (Movable Type Scripts)).

Table 5  
Errors of 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

### 3. DNN model and transfer learning

An effective DNN called ResNet was invented by He et al. (2016) in 2015. Because of its great performance in image recognition, ResNet's popularity expanded steadily after its publication. The specific network used in this model is called ResNet34, where 34 represents the number of convolutional layers.

#### 3.1. Overview architecture of ResNet34

If we take 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 Fig. 14. The portion bounded in the red box is called fully connected (FC) layers, and the details will be explained in the following sections. The remaining portions on the left side compose the convolutional layers. Readers can refer to He et al.'s paper for more

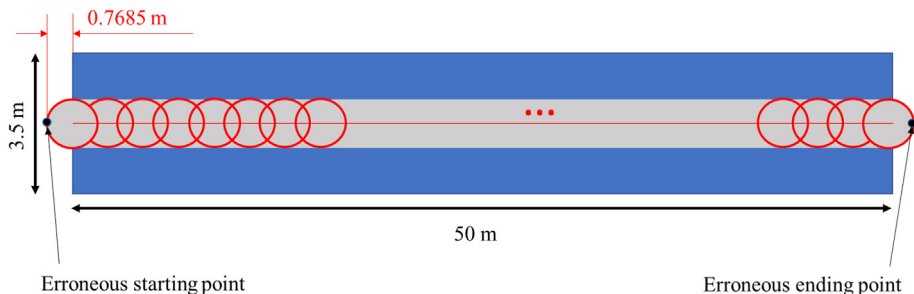


Fig. 13. Configuration of the camera path in a highway lane (not to scale).

information about the residual learning algorithm employed in the convolutional layers. The key feature of residual learning is that it optimizes the residual mapping instead of optimizing the original, unreferenced mapping. Based on the authors' hypothesis, the residual mapping would be easier to optimize. One extreme example is that, if the optimal original function were an identity mapping, it would be easier for the optimizer to drive the residual function to zero than to find the identity mapping by a series of nonlinear functions.

#### 3.2. Mixed pooling layer

Max pooling and average pooling layers have their 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. (2014) 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. (5).

$$y = \lambda \cdot y_{max} + (1 - \lambda) \cdot y_{avg} \quad (5)$$

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

In the model used in this paper,  $\lambda$  was fixed to be 0.5 to keep both features from max and average pooling layers and reduce the trainable weight parameters to save computing resources.

#### 3.3. FC layers

To reduce the 1024-dimensional feature vector to a 1-dimensional vector, three trainable linear functions, as shown in Eq. (6) are appended after flattening the  $1 \times 1 \times 1,024$  tensor.

$$y = wx + b \quad (6)$$

where  $y$  is the output vector,  $x$  is the input vector,  $w$  is the weight and  $b$  is the bias.

For example, to reduce a 4D vector  $x$  to a 2D vector  $y$ :

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (7)$$

By doing so,  $2 \cdot 4 + 2 = 10$  parameters were added to this layer. Similarly, by reducing the dimension within three steps shown in Fig. 14, 533,025 trainable parameters were added rather than only 1025 if reduced within one step. The inputs are normalized, and 50% dropouts are applied before every linear transformation to accelerate the training while avoiding quick overfitting. Additionally, rectified linear unit (ReLU) activation function was employed after every linear layer. Detailed information on the FC layers is attached in Appendix D.

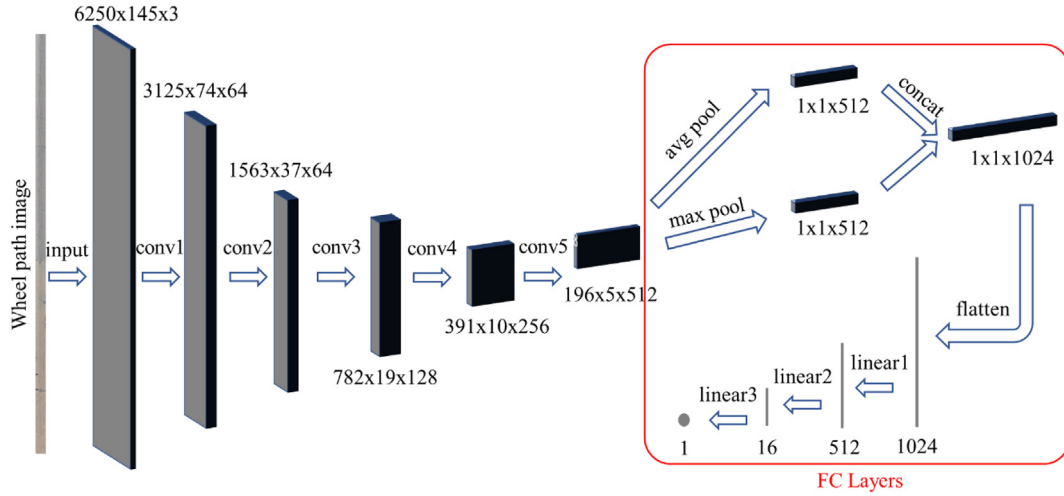


Fig. 14. Architecture of the modified ResNet34 (He et al., 2016).

### 3.4. Loss function

It is known that the critical task of a DNN is to find the parameters that minimize the loss specified by the experimenter. Cross-entropy is the most employed loss function in classification problems. To meet the needs of IRI identification problems, a new loss function suitable for regression problems is required. There are two main loss functions to address regression problems, L1 and L2 losses. L1 loss (i.e., absolute error loss) accumulates the absolute differences between ground truths and predictions. L2 loss (i.e., squared error loss), on the contrary, calculates the squared differences between them and then sums them up. Both of them were experimented with, but it was found that neither of them produced satisfactory outcomes. The possible reasons could be that the L1 loss can not converge smoothly near the minimum loss point while L2 loss is too sensitive to the outliers due to the effect of exponential growth. Therefore, another loss function, the smooth L1 loss function, is introduced. It is the sum of Huber's losses. And Huber's loss is shown in the following Eq. (8):

$$\text{Huber's loss} = \begin{cases} \frac{1}{2}(IRI_{ground} - IRI_{pred})^2 & \text{for } |IRI_{ground} - IRI_{pred}| \leq \delta, \\ \delta \left( |IRI_{ground} - IRI_{pred}| - \frac{1}{2}\delta \right), & \text{otherwise.} \end{cases} \quad (8)$$

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

As shown in Fig. 15, Huber's loss mitigates the sensitivity to the outliers compared with L2 loss. Meanwhile, it keeps the advantage of L2 loss

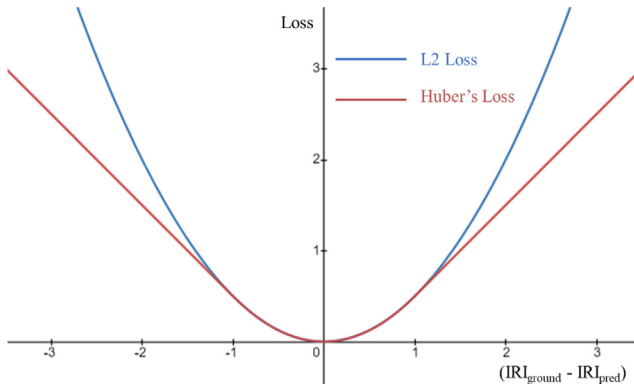


Fig. 15. Comparison of L2 loss with Huber's loss as a function of  $(IRI_{ground} - IRI_{pred})$ .

in the manner that the gradient gets smaller when the network is about to reach the minimum loss point.

### 3.5. 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 (2016) ResNet34 employed Stochastic Gradient Descent (SGD) algorithm. The authors used some tricks during iterations to save the computational 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 (Kingma and Ba, 2014) in 2015, has been proven more effective.

Adam is an extension of SGD that combines the advantages of two other algorithms: the Adaptive Gradient Algorithm (AdaGrad) (Duchi et al., 2011) which can deal with sparse gradients, and Root Mean Square Propagation (RMSProp) (Hinton et al.) which can deal with non-stationary objectives. At every timestep ( $t$ ), Adam first estimates the first moment ( $m_t$ ) and the second moment ( $v_t$ ) of the gradients ( $g_t$ ), as shown Eq. (9):

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned} \quad (9)$$

$g_t = \nabla_w Q_t(w_{t-1})$  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.,  $\hat{m}_t$  and  $\hat{v}_t$ ), as shown in Eq. (10):

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \end{aligned} \quad (10)$$

Then, the corrected moment estimates are employed to update the weights  $w$ :

$$w_t = w_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (11)$$

where,  $\eta$  is the learning rate and its default value is 0.001,  $\epsilon$  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 slightly that  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ ,  $\epsilon = 10^{-5}$ .

### 3.6. Cyclical learning rate

It is known that the learning rate is one of the most crucial hyperparameters while training a DNN. This paper follows a strategy developed by Smith (2015) to determine the learning rate. We train the network over a few iterations and cyclically vary learning rates between reasonable boundary values at each mini batch. The loss recorder will plot the losses against learning rates, and typically the point with the deepest decreasing slope represents the optimum learning rate. Fig. 16 shows the cyclical learning results based on our datasets and model. 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.

### 3.7. Transfer learning

Transfer learning is a practical option to develop new models for new circumstances. Considering the limited size of the labelled datasets, the proposed DNN model took advantage of a pre-trained ResNet34 model from the PyTorch library (Paszke et al., 2017), which had been pre-trained on ImageNet (Deng et al., 2009) database to classify 1000 image categories. Only part of the pre-trained 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 from which 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 the FC layers were trained for the first several epochs. However, our experiments showed that it was not necessary to train the FC layers more than once because the epochs after the first one barely reduce the training losses. Therefore, the convolutional layers were only frozen for the first epoch then unfrozen, and the entire network was finally trained.

## 4. Training and testing results

Since the dataset size with IRI values larger than 4 was too small to be trained and tested on, those were removed for the simplicity of this study. In other words, this paper only focuses on the data that have IRI values smaller than 4. After filtering, 1013 left wheel path images and 1022 right wheel path images remained. The histograms in Fig. 17 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 condition (Alberta Government, 26 June 2018). Small variations occurred for the bar between 0.55 and 0.75. This could be a stochastic phenomenon because of the relatively small sample sizes.

The complete data were 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. The left wheel path dataset did not overlap with the right wheel path dataset, and these two datasets composed the combined dataset. Three independent models were trained and tested on the three datasets, respectively. In each dataset, 68% were randomly assigned for training, 17% were assigned for validation, and the rest 15% were 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 trained properly, 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 sub-sections. All the experiments

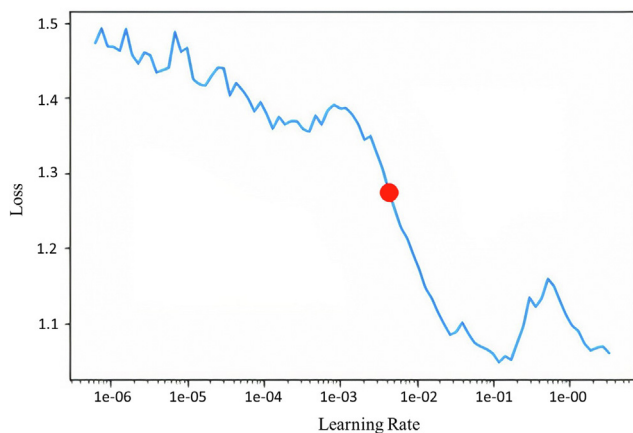


Fig. 16. Loss vs learning rate curve.

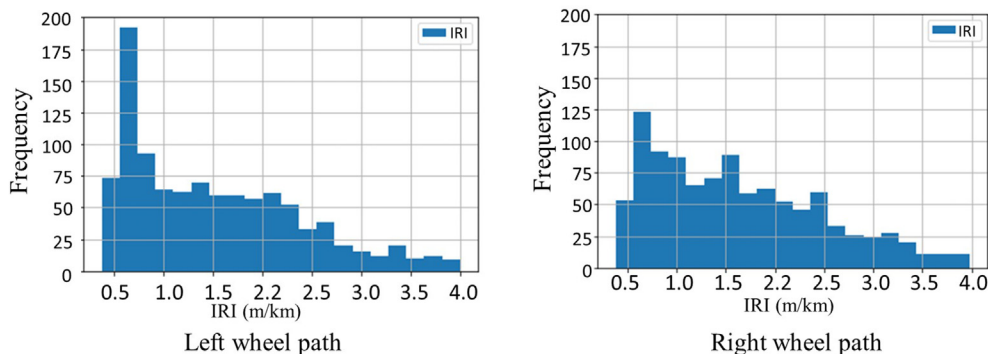


Fig. 17. IRI distributions.

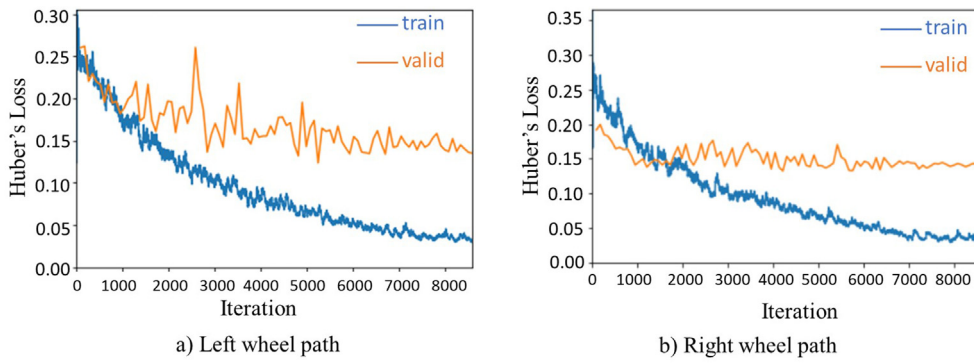


Fig. 18. Training and validation loss of left wheel path and right wheel path separately.

were run on Google Colab Pro, where a Tesla V100-SXM2 GPU with 16 GB memory was available.

#### 4.1. Separated left and right wheel path datasets

The number of training epochs was set to 100 for each group of experiments, and the batch size was set to eight. Fig. 18 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 implies 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  $1013 \times 68\% \cong 689$  images, it took  $689/8 \cong 87$  iterations. As the total number of training epochs was 100, the total number of iterations was  $87 \times 100 = 8700$ .

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 as 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 Fig. 18 is that the validation losses of both groups started to level out after approximately 1500 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. These phenomena of the divergences between training and validation loss are 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 if the network was only trained to fit the training data.

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 regression performance 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. (12).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (IRI_{ground,i} - IRI_{pred,i})^2}{n}} \quad (12)$$

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 the segment  $i$ , and  $IRI_{pred,i}$  is the IRI prediction of the segment  $i$ .

The lowest observed RMSEs of the left and the right wheel path validation dataset were 0.49 and 0.50, respectively. The lowest RMSEs of the left and the right wheel path validation datasets were 0.49 and 0.50, respectively. To verify the performances of the proposed models, the trained networks were applied to the testing subsets, and their results were plotted in the form of scatter diagrams, as shown in Fig. 19. 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 the ground truths and the predictions. Ideally, if the predictions were 100% accurate, all of the points should align perfectly on the line of  $y = 1.0x + 0$ , and the corresponding R squared is 1.0.

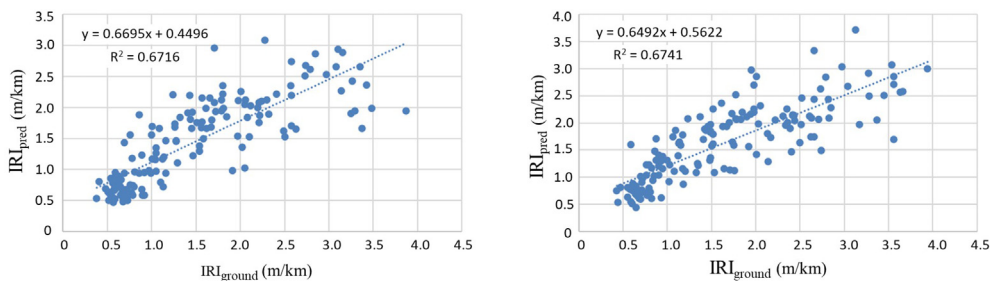


Fig. 19. Relationships between IRI ground truths and predictions of testing datasets.

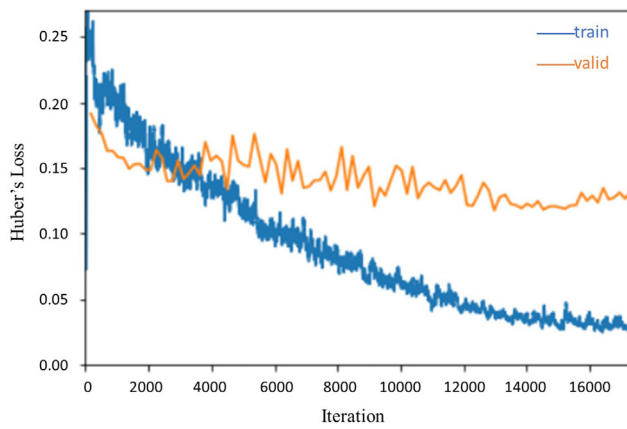


Fig. 20. Training and validation loss of combined dataset.

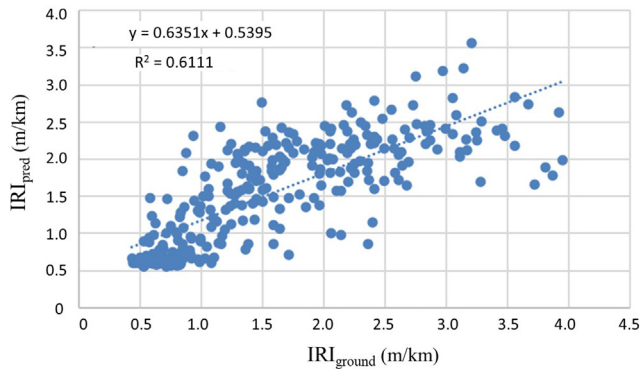


Fig. 21. Relationships between ground truths and predictions of the combined testing subset.

From the above graphs, the R squared values of the left and the right wheel path testing subsets were 0.6716 and 0.6741, respectively. It indicates that very close performance was achieved with both groups. In addition to R squared, the RMSEs of the testing subsets were also computed. It was 0.49 for the left wheel path and 0.51 for the right wheel path. The very close RMSEs again confirmed similar performances in the two groups.

4.2. Combining left and right wheel path datasets

To determine whether the left wheel path and right wheel path images could be mixed and trained together, the datasets of the left and

right wheel paths were combined, and the same training/testing approaches in section 4.1 were reapplied, including the splitting percentages of the dataset, the batch size, and the number of training epochs. The learning curves are shown in Fig. 20. The corresponding lowest RMSE of validation was 0.46.

The RMSE achieved with the combined testing dataset is 0.54. The fitted regression line of R squared equal to 0.6111 is shown in Fig. 21.

Comparisons between the separated datasets and the combined dataset are presented in section 5.1.

4.3. Intuitive demonstrations of the testing results

To demonstrate the testing predictions, three left wheel path images were randomly picked from the testing dataset and predicted by the trained models. The results are shown in Fig. 22.

As indicated in the above figure, the predictions of the first two images were relatively good, while the prediction 2 of the last image was much lower than the ground truth. The reason could be that the training data of larger IRI were relatively insufficient compared with smaller IRI. The other reason might be that prediction 2 was given by the model trained on the combined dataset, which was not exclusively designed for the left wheel path. In other words, if the objective of the network is to predict the IRI values of the left wheel paths, providing more right wheel path images may not help because the focus of the network is spread and its objective is generalized to both wheel paths.

5. Discussion

5.1. Comparison between the results of various dataset combinations

Table 6 lists the testing results of the left and right wheel path datasets (i.e., the results from section 4.1) in the first two 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 4.2).

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 by 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. Technically, if a model is designed to predict IRI values of the left wheel path, providing more right wheel path images may not be a good practice because it will spread the focus of the network and generalize its objective.

[Road28A\\_Seg1\\_Left184\\_IRI=0.98.jpg](#)

---

Ground truth = 0.98, Prediction1 = 1.25, Prediction2 = 1.15

[Road825\\_Seg1\\_Left10\\_IRI=2.03.jpg](#)

---

Ground truth = 2.03, Prediction1 = 1.93, Prediction2 = 2.04

[Road831\\_Seg2\\_Left90\\_IRI=3.28.jpg](#)

---

Ground truth = 3.28, Prediction1 = 3.29, Prediction2 = 1.89

Notice: Prediction1: IRI predicted by the model trained only with left wheel path dataset  
 Prediction2: IRI predicted by the model trained with the combined left and right wheel path dataset

Fig. 22. Samples of prediction results.

**Table 6**  
Testing results.

Dataset	RMSE	R squared
Left wheel path dataset	0.49	0.6716
Right wheel path dataset	0.51	0.6741
Averaged	0.50	0.6728
Combined dataset	0.54	0.6111

### 5.2. Comparison with the other present-based models

To verify the proposed model's performance, comparisons were made with three other latest present-based models published within two years. However, to the best of the author's knowledge, 2D image-based IRI identification models have not yet been discussed in the literature. Thus, the following three models are either based on vibrations or 3D images.

Jeong et al. (2020) utilized the dynamic responses from smartphones mounted on testing vehicles to train their DNN model. Their model identified the IRI values for 8.5-m segment intervals. But it should be noted that their model has only been numerically verified on four types of passenger vehicles. 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. By comparison, although our RMSE (0.505) is smaller, our R squared (0.673) still has room for improvement.

The other model proposed by Mirtabar et al. (2020) 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 the constant 8.5-m intervals used in the previous example, their model identifies the IRI values for varying segment lengths: 10-m, 20-m, 50-m, and 100-m. They found that the R squared increased when segment length was extended, which indicated that a better goodness-of-fit could be obtained if the segment length increases. Their R squared of the same segment length (50-m) in this paper was 0.778, which was approximately 15% higher than the reported value in this article (0.673). Their RMSE was only reported for the 100-m segments since longer segment length gave better accuracy. They repeated the testing experiments by three times, and the average RMSE was 0.41, which was 18% lower than the RMSE reported in this paper (0.50).

Abohmer et al. (2021) developed a 3D image-based CNN model to predict IRI values. In their model, they extracted 3D images of 34 road segments with a length of 160-m (0.10-mi) from the government-released datasets in Louisiana, the US, along with their ground truth IRI values. Every segment was composed of 25 equally divided small 3D images that share the same average IRI ground truth. The developers then randomly selected 20 images from each segment for training purposes while the remaining five were used for testing. Technically, the authors of this article do not support the way they split the training/testing datasets because of the close proximity of pavement characteristics within one segment. Normally people use a segment as a whole and employ many of them for training, and then use the other unseen segments for testing. Despite this, their testing results are still presented here due to the lack of existing 3D image-based DNN models to predict pavement roughness indices. Their model reached a high R squared of 0.985 and a relative RMSE of 5.9%.

Although the current accuracy of this model is acceptable at the highway network level in the majority of countries, it is clearly far from perfect. However, it still gives a novel direction that imaging-based DNN models can be a possible method to accurately identify IRI if appropriate improvements can be achieved in the future. The potential approaches include larger datasets, more suitable DNNs, more precise image stitching methods, and narrowing the width of the wheel path images to discard as many unnecessary pavement surfaces as possible. 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. Conclusions

This article proposed a novel imaging-based DNN model that was verified in real highway scenarios. This approach could be an alternative to currently popular vibration-based models, to identify the IRI values of pavements. The proposed model proved that 2D images may not contain all the longitudinal profile information of the pavements compared with vibrations, but they can still be employed to identify pavement IRI values. The possible reason could be that, the distress information can be implicitly indicated in 2D images in the form of contrast and brightness change even though the absolute depth differences are not explicitly known. In addition to the architecture of the network, the procedures of collecting imaging data and stitching full-size images were also presented. Considering an R squared of 0.6728 and an RMSE of 0.50, the authors understand that the current model has relatively lower accuracies compared with some of the vibration-based models, but the proposed model has these advantages:

- 1) Existing commercial cameras on average can accomplish imaging data collection even if no specially designed equipment is involved.
- 2) Imaging data are independent of the speed of the testing vehicle and thus no calibration is required.
- 3) 2D images encapsulate the longitudinal profile information to some extent if not all.

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 the segment level and obtain similar performance as the current industrial IRI measurement methods. The authors will continue with the 2D image-based methods and develop models of higher accuracies in light of potentially the cheapest way to evaluate the pavement roughness. Possible improvement approaches include larger datasets, more suitable DNNs, more precise image stitching methods, and narrowing the width of the wheel path images to discard as many unnecessary pavement surfaces as possible. 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 different approaches.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

The investigated highway sections are highlighted in Figure A.1 with thick blue lines and their corresponding starting and ending coordinates are listed in Table A.1. These sections are not dispersed but relatively concentrated northeast of the city due to budget and time limitations.

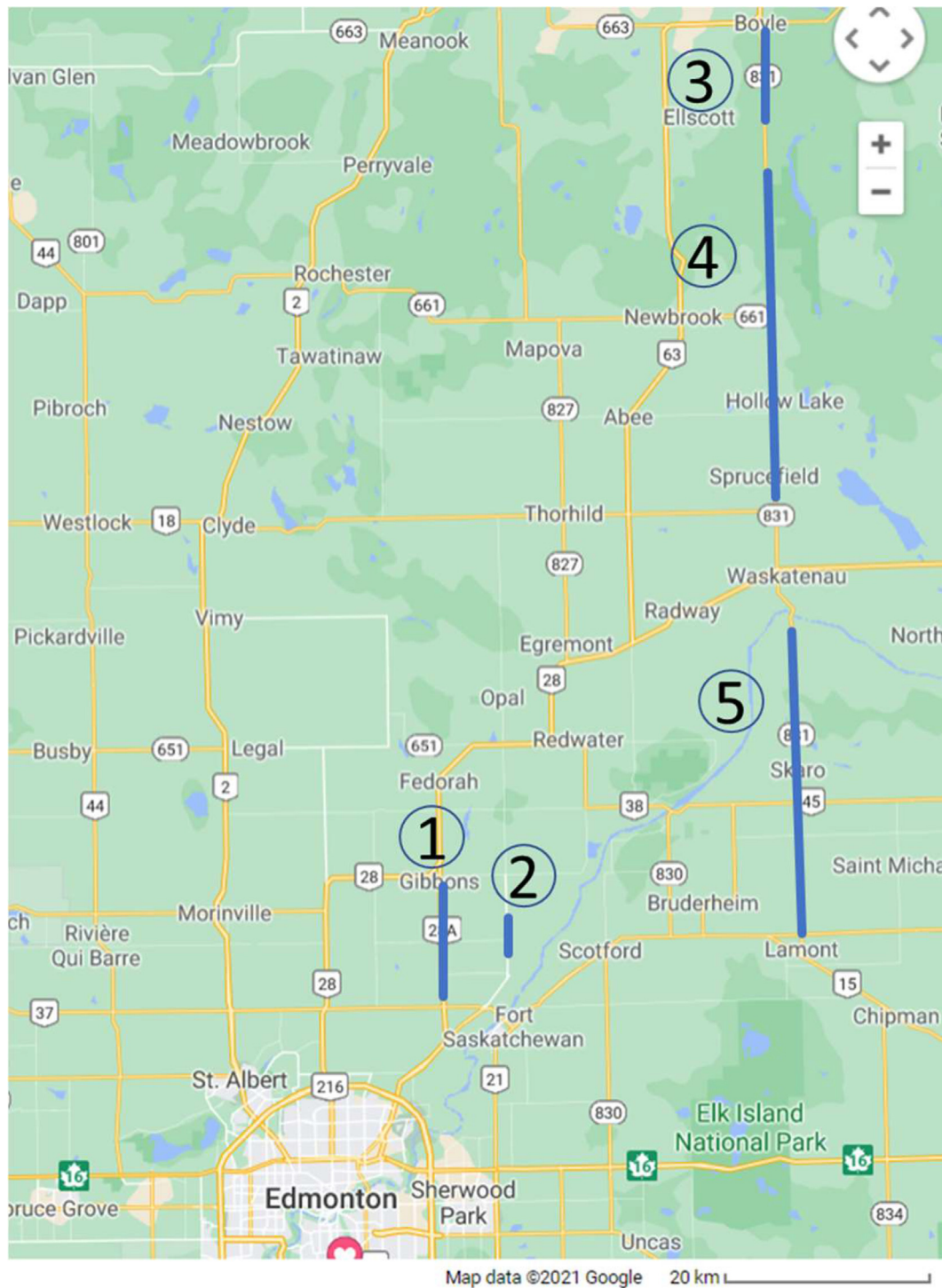


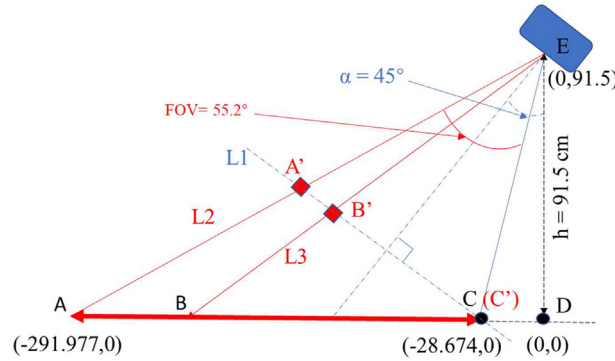
Fig. A.1. Maps of the investigated road sections (modified from (Google Maps)).

**Table A.1**  
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

**Appendix B**

Fig. 3 is modified to calculate the length of AB, as shown in Figure B.1. 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 is the line connecting B and its projecting point B'.



**Fig. B.1.** 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), 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)
- $\frac{A'B'}{B'C'} = \frac{345 \text{ pixels}}{1135 \text{ pixels}}$  (Fig. 4), 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 x-axis at B, so B (-162.062,0)
- A (-291.977,0) and B (-162.062,0), so AB = 129.9 cm.

**Appendix C**

To describe the interpolation method, an example is given in Figure C.1, where the starting point of a segment is given, 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, and the trivial deviations in the longitudinal coordinates are due to the sway of the vehicle 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
	<b>Starting point:</b>	<b>53.7063143</b>	<b>-113.3200180</b>		<b>62</b>
③	2021-07-10T18:44:03.847Z	53.7064032	-113.3200174	0.733	87.96

**Fig. C.1.** Sample calculation of using linear interpolation method get the frame number.

To find the corresponding starting frame number, first, 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 the difference between row ③ and row ②, and the ratio of the two differences:

$$\frac{53.7063143 - 53.7062536}{53.7064032 - 53.7062536} = 0.405749 \tag{13}$$

Afterward, this ratio is used to find the corresponding starting frame number:

$$43.92 + 0.405749 \cdot (87.96 - 43.92) = 61.79 \tag{14}$$

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

## Appendix D

See Table D.1.

**Table D.1**  
Summary of FC layers.

Layer Type	Output Shape	Parameter Number	Trainable
AvgPool2d	1 × 1 × 512	0	False
MaxPool2d	1 × 1 × 512	0	False
Flatten	1024	0	False
BatchNorm1d	1024	2048	True
Dropout	1024	0	False
Linear	512	524,800	True
ReLU	512	0	False
BatchNorm1d	512	1024	True
Dropout	512	0	False
Linear	16	8208	True
ReLU	16	0	False
Linear	1	17	True

## References

- Aaa, M., Gth, S., 1981. Road roughness: its evaluation and effect on riding comfort and pavement life. *Transport. Res. Rec.* 863, 41–49.
- Abohamer, H., Elseifi, M., Dhakal, N., et al., 2021. Development of a deep convolutional neural network for the prediction of pavement roughness from 3D images. *J. Transport. Eng., Part B: Pavements* 147 (4), 4021048.
- Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. Understanding of a convolutional neural network. In: *International Conference on Engineering and Technology (ICET)*. Antalya.
- Alberta Government, 26 June 2018. Physical condition of provincial highway surfaces, Alberta [Online]. Available: <https://open.alberta.ca/opendata/physical-condition-of-provincial-highway-surfaces-alberta>. (Accessed 7 November 2021).
- Al-Gaadi, K.A., 2005. Testing the accuracy of autonomous GPS in ground speed measurement. *J. Appl. Sci.* 5 (9), 1518–1522.
- Ali, R., Zeng, J., Cha, Y.-J., 2020. Deep learning-based crack detection in a concrete tunnel structure using multispectral dynamic imaging. In: *SPIE Smart Structures + Nondestructive Evaluation*.
- HERO7 field of view (FOV) information. GoPro, [Online]. Available: [https://gopro.com/help/articles/question\\_answer/hero7-field-of-view-fov-information?sf96748270=1](https://gopro.com/help/articles/question_answer/hero7-field-of-view-fov-information?sf96748270=1). (Accessed 26 August 2021).
- Deng, J., Dong, W., Socher, R., et al., 2009. ImageNet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.
- Du, Y., Liu, C., Wu, D., et al., 2014. Measurement of international roughness index by using Z-Axis Accelerometers and GPS. *Math. Probl. Eng.* 2014.
- Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* 12 (7), 2121–2159.
- Infrastructure and transportation, 2006. *Guidelines for Assessing Pavement Preservation Treatments and Strategies : Edition 2*. Edmonton, Infrastructure and Transportation, p. 4.
- El-Hakim, R.A., El-Badawy, S., 2013. International roughness index prediction for rigid pavements: an artificial neural network application. *Adv. Mater. Res.* 723, 854–860.
- Federal Highway Administration, 2017. *The Long-Term Pavement Performance Program*. FHWA, McLean.
- FHWA, 2011. Our Nation's Highways, 7th November 2014. [Online]. Available: <http://www.fhwa.dot.gov/policyinformation/pubs/hf/pl11028/chapter1.cfm>. (Accessed 17 October 2021).
- Gong, H., Sun, Y., Shu, X., Huang, B., 2018. Use of random forests regression for predicting IRI of asphalt pavements. *Construct. Build. Mater.* 189, 890–897.
- Google Maps. Google [Online]. Available: <https://www.google.com/maps>. (Accessed 26 August 2021).
- Gopalakrishnan, K., Khaitan, S.K., Choudhary, A., et al., 2017. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construct. Build. Mater.* 157, 322–330.
- Government of Alberta, 2020. *International Roughness Index and Rut Data*. Edmonton.
- Government of Alberta, "Provincial Construction Program," Edmonton.
- He, K., Zhang, X., Ren, S., et al., 2016. Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*. CVPR, pp. 770–778.
- Highway Research Board, 1962. *Special Report 61E: the AASHO Road Test, Report 5: Pavement Research*. National Research Council, D.C.
- G. Hinton, N. Srivastava and K. Swersky, rmsprop: divide the gradient by a running average of its recent magnitude.
- Hossain, M., Gopiseti, L.S.P., Miah, M.S., 2020. Artificial neural network modelling to predict international roughness index of rigid pavements. *Int. J. Pavement Res. Technol.* 13, 229–239.
- Huang, Y.H., 2004. *Pavement Analysis and Design*. Pearson Education Hall.
- Jeong, J.H., Jo, H., Ditzler, G., 2020. Convolutional neural networks for pavement roughness assessment using calibration-free vehicle dynamics. *Comput. Aided Civ. Infrastruct. Eng.* 35 (11), 1209–1229.
- Jurgens, R., Chan, J., 2005. Highway performance measures for business plans in Alberta. In: *Session of the 2005 Annual Conference of the Transportation Association of Canada*, Calgary.
- Karballeaezadeh, N., Zaremotekhas, F., Shamshirband, S., Mosavi, A., Nabipour, N., Csiba, P., Várkonyi-Kóczy, A.R., 2020. Intelligent road inspection with advanced machine learning: hybrid prediction models for smart mobility and transportation maintenance systems. *Energies* 13 (7).
- Kingma, D.P., Ba, J.L., 2014. Adam: a method for stochastic optimization. In: *The 3rd International Conference for Learning Representations*. San Diego.
- Kumar, A., Chakrapani, Kalita, D.J., Singh, V.P., 2020. A modern pothole detection technique using deep learning. In: *2nd International Conference on Data, Engineering and Applications (IDEA)*. Bhopal.
- Maeda, H., Sekimoto, Y., Seto, T., et al., 2018. Road damage detection using deep neural networks with images captured through a smartphone. *Comput. Aided Civ. Infrastruct. Eng.* 33 (13), 1127–1141.
- Mei, Q., Gul, M., 2020. A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Construct. Build. Mater.* 256.
- Milhomem, S., Almeida, T.d.S., Silva, W.G.d., et al., 2018. Weightless neural network with transfer learning to detect distress in asphalt. *Int. J. Adv. Eng. Res. Sci.* 5 (12), 294–299.
- Mirtabar, Z., Golroo, A., Mahmoudzadeh, A., et al., 2020. Development of a crowdsourcing-based system for computing the international roughness index. *Int. J. Pavement Eng.* 23 (2), 489–498.
- Mirtabar, Z., Golroo, A., Mahmoudzadeh, A., et al., 2022. Development of a crowdsourcing-based system for computing the international roughness index. *Int. J. Pavement Eng.* 23 (2), 489–498.
- Moh, A., Roy, J., 2000. *Uses & Comparison of IRI with Other Jurisdictions (Asphaltic Concrete Pavements)*. Edmonton, Alberta Infrastructure, p. 3.
- 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 November 2021).
- Nagayama, T., Miyajima, A., Kimura, S., et al., 2013. Road condition evaluation using the vibration response of ordinary vehicles and synchronously recorded movies. In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*.
- Ngwangwa, H., Heyns, P., Labuschagne, F., et al., 2010. Reconstruction of road defects and road roughness classification using vehicle responses with artificial neural networks simulation. *J. Terramechanics* 47 (2), 97–111.
- Paszke, A., Gross, S., Chintala, S., et al., 2017. *Automatic differentiation in PyTorch*. In: *31st Conference on Neural Information Processing Systems*. Long Beach.
- Power, J.D., 2021 [Online]. Available: <https://www.nadaguides.com/cars/2016/bmw/3-series/Sedan-4D-328i-14-Turbo/Pictures/Print>. (Accessed 15 July 2021).
- Qin, Y., Xiang, C., Wang, Z., et al., 2018. Road excitation classification for semi-active suspension system based on system response. *J. Vib. Control* 24 (13), 2732–2748.
- Ragnoli, A., Blasiis, M.R.D., Benedetto, A.D., 2018. Pavement distress detection methods: a review. *Infrastructures* 3 (4), 58.
- Sayer, M.W., Gillespie, T.D., Paterson, W.D.O., 1986. *Guidelines for the Conduct and Calibration of Road Roughness Measurements*. The World Bank, Washington, D.C., p. iii
- Sayers, M.W., Karamihas, S.M., 1998. *A Little Book of Profiling*. University of Michigan.
- Sayers, M.W., Gillespie, T.D., Queiroz, C.A.V., 1986. *The International Road Roughness Experiment: Establishing Correlation and a Calibration Standard for Measurements*. The World Bank, Washington.
- Shahin, M.Y., Darter, M.I., Kohn, S.D., 1977. *Development of a Pavement Maintenance Management System, vol. I. Airfield Pavement Condition Rating*.
- Smith, L.N., 2015. *Cyclical Learning Rates for Training Neural Networks*.
- Souza, V.M.A., RafaelGiusti, Batista, A.J.L., 2018. Asfalt: a low-cost system to evaluate pavement conditions in real-time using smartphones and machine learning. *Pervasive Mob. Comput.* 51, 121–137.

- Tehrani, S.S., 2014. Performance Measurement Using IRI and Collision Rates in the Province of Alberta. University of Calgary, Calgary.
- Tong, Z., Gao, J., Sha, A., et al., 2018. Convolutional neural network for asphalt pavement surface texture analysis. *Comput. Aided Civ. Infrastruct. Eng.* 33 (12), 1056–1072.
- Wang, S.C., 2003. Artificial neural network. In: *Interdisciplinary Computing in Java Programming*, vol. 743. The Springer International Series in Engineering and Computer Science, Boston, MA, p. 83.
- Wei, Y., Zhang, Y., Huang, J., Yang, Q., 2018. Transfer learning via learning to transfer. In: *35th International Conference on Machine Learning*. Stockholm.
- Wong, C., Worden, K., 2005. Generalised NARX shunting neural network modelling of friction. *Mech. Syst. Signal Process.* 21 (1), 553–572.
- Yu, D., Wang, H., Chen, P., Wei, Z., 2014. Mixed pooling for convolutional neural networks. In: *International Conference on Rough Sets and Knowledge Technology*.
- Zeida, W., Dabous, S.A., Hamad, K., et al., 2020. Machine learning for pavement performance modelling in warm climate regions. *Arabian J. Sci. Eng.* 45, 4091–4109.
- Zhang, Z., Sun, C., Bridgelall, R., et al., 2018. Application of a machine learning method to evaluate road roughness from connected vehicles. *J. Transport. Eng.* 144 (4).
- Zhao, B., Nagayama, T., Toyoda, M., et al., 2017. Vehicle model calibration in the frequency domain and its application to large-scale IRI estimation. *J. Disaster Res.* 12 (3), 446–455.