International Journal of Transportation Science and Technology 11 (2022) 298–309

Contents lists available at ScienceDirect

International Journal of Transportation Science and Technology

journal homepage: www.elsevier.com/locate/ijtst

Pavement distress detection using convolutional neural network (CNN): A case study in Montreal, Canada

Ce Zhang, Ehsan Nateghinia *, Luis F. Miranda-Moreno, Lijun Sun

Dept. of Civil Engineering and Applied Mechanics, McGill University, Montreal, QC, H3A 0C3, Canada

ARTICLE INFO

Article history:
Received 3 November 2020
Received in revised form 14 February 2021
Accepted 18 April 2021
Available online 28 April 2021

Keywords:
Pavement distress type detection and classification
Road maintenance operation
Camera-based pavement monitoring system
Convolutional neural network

ABSTRACT

Pavement distresses, including cracking and disintegration, deteriorate road user's comfort, damage vehicles, increase evasive maneuvers, and increase emissions. Transportation agencies spend a significant portion of their budget to monitor and maintain road pavements. Pavement distress can be identified through manual surveys, i.e., visual inspections of pavement images captured by an inspection vehicle. To reduce manual inspection costs, research and industry have moved quietly towards the development and implementation of automated road surface monitoring systems.

Considering the latest research developments, the objective of this work is to propose and evaluate a methodology for automated detection and classification of pavement distress types using Convolutional Neural Networks (CNN) and a low-cost video data collection strategy. In this work, pavement distress types are categorized as linear or longitudinal crack, network crack, fatigue crack or pothole, patch, and pavement marking. The models are trained and tested based on an image dataset collected from Montreal's road pavements. A sensitivity analysis is carried on for evaluating different regularization scenarios and data generation strategies especially scaling and partitioning the input image. The detection rate and classification accuracy of the proposed approach with the trained CNN model reaches 83.8% over the test set, which is promising compared with the literature. More specifically, the F1-scores for "*pothole*", "*patch*", "*marking*", "*crack-linear*" and "*crack-network*" classes are 0.808, 0.802, 0.860, 0.796, and 0.813, respectively. However, by merging linear and network crack classes, the F1-score over the merged class increases to 0.916.

Introduction

Because of exposure to abrasion from vehicle traffic, material aging, and extreme weather conditions, pavement surface conditions deteriorate. This is manifested in the form of cracks or other distress types (distortion, disintegration, etc.) (Babkov, 1975). Pavement cracks make people uncomfortable while driving, damage vehicles, and may increase evasive maneuvers which can result in a crash (Elghriany, 2016). However, the relationship between pavement conditions and crash frequency and severity has not been well studied due to the lack of collecting pavement conditions when reporting a crash.

Peer review under responsibility of Tongji University and Tongji University Press.

* Corresponding author.
E-mail address: ehsan.nateghinia@mail.mcgill.ca (E. Nateghinia).

https://doi.org/10.1016/j.ijtst.2021.04.008
2046-0430/© 2021 Tongji University and Tongji University Press. Publishing Services by Elsevier B.V.

Despite pavement distress detection through manual surveys (using visual inspections of pavement images from inspection vehicles), automated video-based road surface monitoring systems have emerged in research and practice in the last years. The use of automated monitoring solutions as a part of the pavement maintenance infrastructures is crucial to reduce costs and increase efficiency.

Compared to highway pavement cracks, urban pavement cracks can have more impacts on the comfort levels of road users especially the vulnerable road users, including cyclists and people with disabilities. Pavement cracks near intersections or crosswalks can increase the risk of road accidents (Huang, 2004).

Collecting pavement surface data is a tedious process for governments. They need to invest in human resources, special inspection vehicles, multiple sensors or cameras, and the mechanical mounting tools to provide robustness against shaking caused by acceleration/deceleration of the inspection vehicle (Jokela et al., 2009). Cities use multifunctional vehicles equipped with pavement monitoring systems that a human operator manually computes a score–*Pavement Condition Index* (PCI)–based on pavement distress type and severity of them for each block of the road (Baladi et al., 1991). The PCI ranges from 1 to 100, where 100 represents a section without any imperfection.

In the literature, many studies focused on crack detection in highways' pavement. Mahler et al. applied image processing techniques to extract crack features from highway images and analyzed those features to find the sections that need to be repaired (Mahler et al., 1991). According to the Montreal PCI report in 2018, the average PCI of 14,114 urban pavements was 61.6, and more than 30% of pavements were in bad or worse conditions (Montreal, 2019).

Studies in the literature have used machine learning and deep learning methods. However, most of the studies have focused on only one or two distress types. Ouma's research focuses on linear crack detection (Ouma and Hahn, 2016) and Siriborvornratanakul's research focuses on potholes detection (Siriborvornratanakul, 2018). Besides, the data collection module used in most studies is complicated and heavy, which must be installed on special vehicles and their cameras unit cannot provide stable images without their fully equipped vehicle.

Based on recent advances in machine learning, this research proposes and evaluates an automated methodology for pavement distress detection and type classification using convolutional neural networks and a low-cost video data collection strategy. The camera used in this research can be installed in almost every vehicle, including cars or bicycles. As part of this research, a pavement dataset with an adequate number of high-quality images is collected in the city of Montreal. The dataset is manually labeled with different pavement distress types, including linear crack, network crack, fatigue crack or potholes, patching, and pavement markings. As part of the methodology, alternative CNN models are implemented with various structures and regularizations.

Literature review

The literature on automated pavement image analysis is immense in terms of data collection and processing techniques. Automatic crack detection techniques have gained interest in the literature, and several image-processing algorithms have emerged for this task in the past few years which can be categorized into two groups: traditional image processing, and machine and deep learning approaches. Traditional image processing methods are mainly based on texture analysis which includes image thresholding or edge detection techniques.

The history of using two-dimensional (2D) image-based detection systems goes back to 1991 when Mahler et al. designed an image-based automatic crack monitoring system (Mahler et al., 1991). It included a time code generator to record the milepost data and a video camera to take grayscale photos. After developing and tuning the image-based crack classification model, they integrated crack detection, thinning, and crack-tracing algorithms into their system. Paying extra attention to the quantitative parameters of the cracks, including direction, length, and width, was their main demonstration.

The *CrackIT* toolbox is another successful crack detection approach based on image thresholding implemented on MATLAB (Oliveira and Correia, 2014). The toolbox classifies images based on brightness and grey level differences, and includes four main modules: image preprocessing, crack detection, crack classification, and evaluation. Although the image thresholding methods have been widely used, they are time-consuming, and their classification accuracy is still insufficient.

The second widely used traditional methods are based on image edge detection. Ayenu et al. implemented a Sobel edge detector to detect cracks after applying image smoothing and noise removal filters (Ayenu-Prah and Attoh-Okine, 2008). Later on, Zou et al. proposed the *CrackTree* algorithm to address the incapability problem of the edge detection algorithms (Zou et al., 2012). First, they designed an algorithm to remove the pavement shadow and then constructed a crack probability map using tensor voting (Medioni et al., 2000), which was efficient to offset noise and fragment. Afterward, they built a graph model by sampling crack seeds from the former map, then constructed the *Minimum Spanning Tree* (MST) of the graph and conducted recursive edge pruning in the MST to identify the final crack curves. Their result showed that the *CrackTree* reached the precision of 0.79 on their dataset with 206 images, which was better than *CrackIT* but still insufficient for practical application.

In 2016, Ouma et al. introduced an empirical approach for incipient linear distress identification in asphalt pavements (Ouma and Hahn, 2016). The pavement distresses included longitudinal, transverse, diagonal, block (random), and alligator (fatigue). In their study, they used a multichannel RGB camera instead of the grayscale camera. They used two different camera types in two different places and developed an algorithm to offset the partial area effect. Their system reported a detection accuracy of 83.2% over the test set.

As alternative approaches, machine learning-based algorithms became popular in transportation applications. Some papers focused on using machine learning methods to deal with automatic crack detection, including *Support Vector Machine*

(SVM) (Marques and Correia, 2012; Lin and Liu, 2010) and *Convolutional Neural Networks* (CNN) (Eisenbach et al., 2017, Bray et al., 2006). In 2010, Lin et al. proposed a recognition algorithm based on SVM with Gaussian kernel (Lin and Liu, 2010). Their system was very competitive in pothole detection. Although the SVM outperforms the traditional image processing significantly, the limitation on classifying multiple classes dataset restricted the promotion of SVM.

Despite being a black-box model, CNN is the most successful method for image classification. Zhang et al. have compared CNN with SVM in road distress classification applications (Zhang et al., 2016). Using the camera of a smartphone, they collected 500 images, with a resolution of 3264 × 2448 pixels, and then categorized them into two classes: *non-crack* and *crack*. Limited by the number of training samples, their result suggests that the CNN outperformed the SVM classifier where the F1-Score of CNN and SVM were 0.89 and 0.73, respectively.

Li et al. proposed and evaluated a nine-layers CNN (with 6 convolutional layers) model with their dataset containing 3D pavement images of 512 × 512 pixels (Li et al., 2020). The images were grouped into six classes: *non-crack*, *longitudinal crack*, *transverse crack*, *block crack*, and *alligator crack* classes. Their proposed system reached an accuracy of 94% on the 3D dataset. However, other pavement distress classes such as *patch* and *pothole* were not discussed.

In 2017, Eisenbach et al. (Eisenbach et al., 2017) presented the *German Asphalt Pavement Distress* (GAPs) dataset, which is an open-source dataset containing two-dimensional pavement distress images, being large enough for training deep neural networks. Following the specific approach for data collection called *Road Monitoring and Assessment* (RMA), they used a certified measuring vehicle. The main components of the vehicle were an inertial navigation system, laser sensors for evenness and texture measurement, a 2D laser range finder, and two cameras with a frame rate of 32 fps and a resolution of 1920 × 1080 pixels for capturing the pavement's surface. The data source in their paper is from the camera system. In total, the GAPs dataset has 1,969 gray-scale images of six distress types: *crack*, *pothole*, *inlaid patch*, *applied patch*, *open joint*, and *bleeding*. Eisenbach et al. trained several CNN with their GAPs dataset and compared them with the classical networks (Eisenbach et al., 2017). As a result, their best CNN model achieved an *F1-Score* of 0.7246. Eisenbach et al. also applied the *CrackIT* toolbox on their GAPs dataset and achieved an accuracy of 0.718 (Eisenbach et al., 2017).

Maeda et al. collected a pavement dataset using a smartphone camera mounted on the dashboard of a car (Maeda et al., 2018). The images include not only pavement surfaces, but also the surrounding environment. Then they labeled their data using multiple bounding boxes while each box encircles a specific type of pavement distress. The labeled pavement distress types include *liner crack longitudinal*, *liner crack lateral*, *alligator crack*, *marking*, and an extra class including all other distress types such as *rutting*, *pothole*, *separation*. As a result of this labeling, the regional CNN models have been used. They implemented Single Shot MultiBox Detector (SSD). The detection and classification accuracies are given as a table where it ranges from 0.77 to 0.95 for different classes.

With advances in technologies, some studies focused on 3D imaging systems. In 2009, Jokela et al. used a monochrome-stereo camera that could build a 3D image of the pavement by taking a pair of photos with a resolution of 640 × 480 pixels (Jokela et al., 2009).

Other cases of efficient 3D scanning systems applicable to pavement distress type detection are based on LiDAR technology. In LiDAR-based systems, the sensing module has a structured light projected laser and a digital area scanner camera. The camera takes images of the laser light. Then the deformations of the laser line on the object area are converted to the depth of each point (Tsai and Li, 2012). Tsai et al. built a *Laser Crack Measurement System* (LCMS) on a vehicle by installing two high-performance laser profiling units that both consisted of a laser line projector, a custom filter, and a camera (Tsai and Li, 2012). Laurent et al. also built an LCMS using 3D laser profilers, custom filters, and a camera (Laurent et al., 2012). Their system automatically detects crack types and computes crack depth and severity. As a result, the accuracy of the system reached 0.95 in general cracks classification. Although LCMSs have proven promising performance in pavement condition monitoring, it is unfeasible to use LCMS for city streets pavement monitoring. One limitation is its high manufacturing and maintenance costs, therefore not every city has one of these systems. Besides, the monitoring speed of the LCMS is slow, therefore in big cities, monitoring with one or two LCMSs takes a long time.

A considerable part of the literature focuses on highway distress detection, and most of the data collection systems are complex, complicated, and expensive. Additionally, these systems are heavy and difficult to mount on a vehicle. Therefore, it is necessary to install them on special inspection vehicles. These factors have limited the number of available inspection vehicles for urban pavement distress monitoring. Moreover, most of the available pavement distress datasets have been collected from highways, which are regularly maintained. As a result, the collected pavement datasets have images with less variety of pavement distresses. Additionally, due to the total length of the streets in a city, a pavement distress monitoring operation requires several inspection vehicles that can perform the task at a reasonably higher speed than the current systems, while the number of current inspection vehicles is limited due to their cost. Therefore, there is a particular gap surrounding efficient urban pavement distress detection and classification systems.

Methodology

The methodology consists of four steps: data collection and dataset preparation, image labeling, and building deep neural networks to learn the patterns of different distress types in pavement images. This section describes the details of each of these processes.

Data collection and dataset preparation

In this study, the road pavement condition in the city of Montreal is used as an application environment. Because of winter weather conditions, Montreal's urban pavement surfaces are likely to be distressed. The city's high latitude and good air quality expose pavements to high-level of sun radiation; besides, the substantial temperature difference between its daytime and nighttime in summer and winter accelerates the pavement aging. Moreover, Montreal has a humid continental climate with several rainy days in spring and fall, and snowy days in winter. The water weakens the pavement surface, and the winter maintenance operations cause heavy erosion.

To collect high-quality images without shaking and vibration, a sports camera, GoPro Hero 7, is used. Choosing a regular camera, such as GoPro, enables the system to be installed on any vehicle and or to replace the camera with any other cameras with the same standard resolutions. Therefore, more inspection vehicles can be employed to achieve a fast monitoring objective.

In this study, the camera resolution was set as 720 dpi, focal ratio (aperture) as f/2.8, film speed as ISO 293, and the exposure time as 1/256 seconds. The dimensions of the output images were set to 3000 × 4000 pixels, and the camera automatically records one image per second. Moreover, the camera can provide location information of the images in the format Global Positing System (GPS).

The camera was mounted on the front bumper of a vehicle, and the inspection vehicle circulated in the city and took consecutive images from the pavement for about 6 hours. Almost 12,000 images containing a variety of pavement surface types were collected in June 2019. From the 12,000 images, the 2,000 best images, containing most of the classical distress types, were selected. During the algorithm's development step, it was found out that the number of pothole images was excessively small. Therefore, the data collection was repeated for the second day, and 105 pothole images were added to the dataset.

The images were converted into grayscale since the texture features of pavement distress are significantly more important than their color features. This conversion also helps to stabilize and standardize the dataset. Then, image histogram equalization was applied to homogenize the intensity distribution of different images. Finally, a median filter was used to filter our *Salt & Pepper* noise.

Image annotation

The FHWA manual (Miller and Bellinger, 2003) has listed various pavement distress types including multiple cracking classes such as fatigue, block, edge, longitudinal, joint reflection, and transverse; in addition to patching, pothole, surface deformations, and defects. For training the deep learning models, it is necessary to have enough samples of every distress class. However, when collecting surface data from urban's pavements, some of these distress types were not observed frequently, and collecting enough samples for those classes was challenging. Therefore, some similar classes have been merged in this work. As a result, all the distress types observed in the collected dataset were labeled as *patching, pothole, crack-linear*, and *network crack*.

In this work, the "*clean*" type means there is no distress in the pavement's image (Fig. 1 (a)). The "*patching*" class, illustrated in Fig. 1 (b), means there is an area of pavement that has been replaced by new materials to repair old distresses on the pavement. The "*pothole*" class, Fig. 1 (c), corresponds to the images that on the pavement surface, there are small bowl-shaped depressions with sharp edges and vertical sides near the top of the hole.

The "*crack-linear*" class, Fig. 1 (d), has been defined as a set of longitudinal, transverse, joint reflection, or/and a block of cracks, which are all mainly comprised of long-crack lines. Moreover, the "*crack-network*" class, illustrated in Fig. 1 (e), is comprised of a series of interconnected cracks by fatigue and has many-sided sharp-angled pieces that make the pavement looks like an alligator back. Lastly, the "*pavement marking*" class, Fig. 1 (f), includes images of road surface marking (crosswalks, traffic lane marking, etc.), which are usually in white or yellow colors.

The images in Fig. 2 show the annotated images for the examples presented in Fig. 1. Different colors (intensities) have been used to annotate different distress types, and the value of the intensity is considered as the numerical value of the corresponding labels. The clean image (Fig. 2 (a)), does not have any annotation since there is no distress in the image. For the patching and pothole examples, polygons with different colors have been used (Fig. 2 (b & c)). The images with linear crack are also annotated with polygons (Fig. 2 (d)), but instead of the whole area, only cracks are filled with the narrow polygons. On the other hand, the network of cracks is annotated the same as patching and pothole since the whole area of distress needs to be repaired (Fig. 2 (e)). Lastly, the marking annotation is the same as the marking shape (Fig. 2 (f)).

For illustration purposes, only one distress type has been annotated in any of the images in Fig. 1. However, in the collected dataset, an image could have multiple distress types, and consequently, all of the distress types of that image have been annotated (see Fig. 4).

Resizing and partitioning

The size of the collected images is 3000 × 4000 pixels, and the bigger the image size is, the bigger CNN models will be. A large number of neurons demand lots of parameters to be learned which increases the learning time and complexity of the model. To limit the size of the CNN model, the original image is resized to a smaller image. Besides, to analyze different sec-

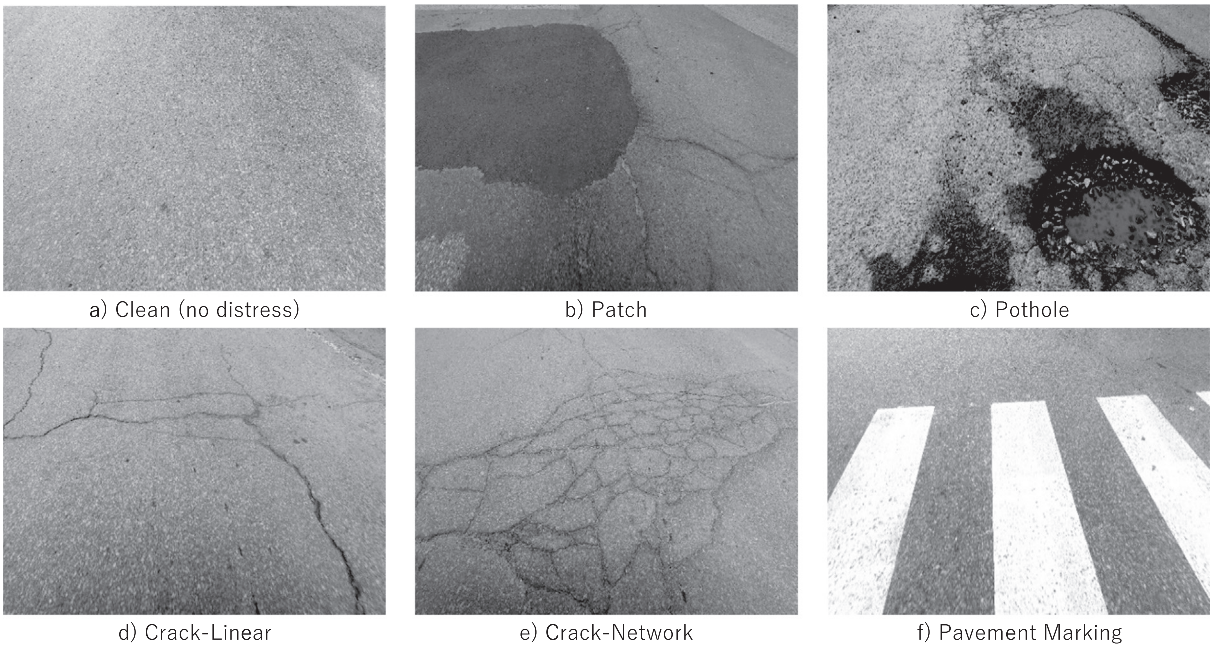


Fig. 1. Different Pavement Distress Types.

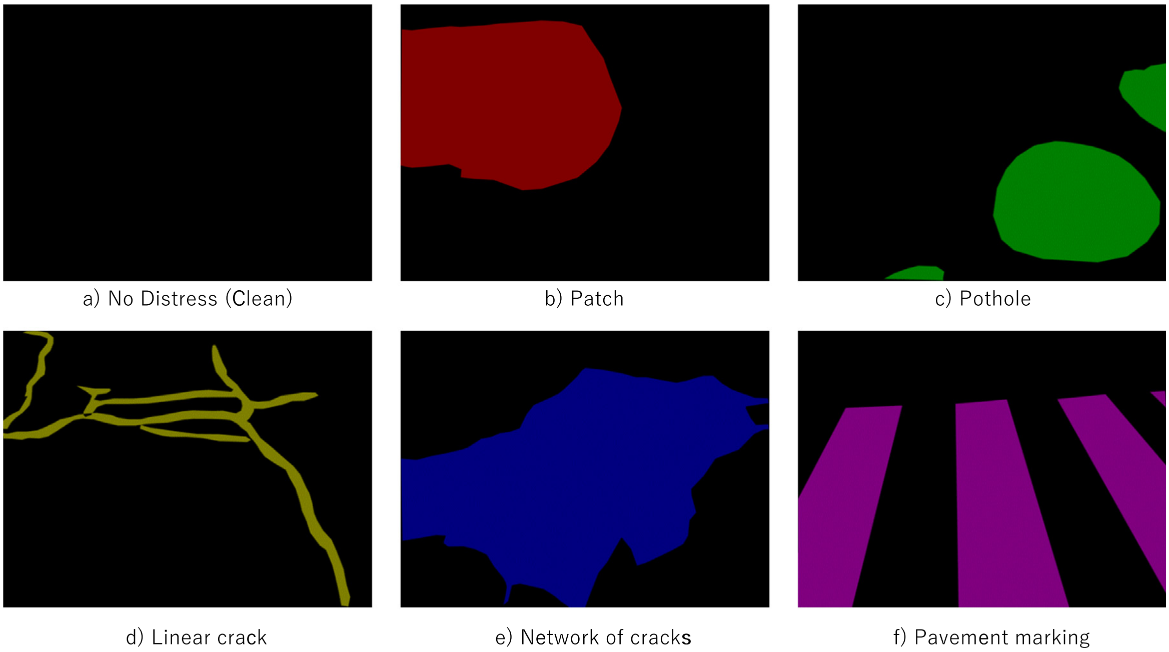


Fig. 2. Annotated Images of the Sample Images in Fig. 1.

tions of the pavement in one image, the reduced size image needs to be split into small sub-images (for example, 100 × 100), which also helps to reduce the number of parameters of the CNN model.

However, choosing the resizing and splitting factors are considered as hyper-parameters. In this research, the two resizing factors, 0.5 and 0.25, are used to convert the original image in 3000 × 4000 pixels to 1500 × 2000 pixels and 750 × 1000 pixels, respectively. In Table 1, four different scenarios are defined and implemented accordingly. In the first, second, and fourth scenarios, the resizing factor is 0.5 but the splitting factor is 100 × 100, 150 × 150, or 200 × 200. In the third scenario, the reducing factor is chosen as 0.25 and the splitting factor is 150 × 150.

In addition to the tuning of the image partitioning hyperparameters, the deep neural network also has its hyperparameters. To fulfill these two tasks, the second dataset summarized in Table 1 is chosen and used to tune the hyperparameters of the CNN model. Then, the four datasets, in Table 1**,** will be tested with the tuned CNN and the best one will be selected as the optimal resizing-splitting scenarios in terms of minimizing the error measures.

Table 2 shows the detailed description of the second dataset containing sub-images with a size of 150 × 150 pixels. The 5th and 8th columns (*percentage of class*) explain what percentage of that class has been sampled for training or test sets, respectively. The 6th and 9th columns (*percentage of training or test set*) clarifies that the sampled images of that class form what percentage of the final training or test sets, respectively.

Of 294,700 total images in the second dataset, only 91,280 images (31% of total images), are used for the learning process. To avoid having an unbiased dataset, images are un-evenly sampled from each class and un-evenly split into training and test samples. The "*clean*" class is the largest class by having 70% of the total images, and of 206,199 "*clean*" images, only 26,806 and 2,062 images (13% and 1% of the total class size) are randomly sampled for the training and test sets, respectively.

On the other hand, all the images of the "*pothole*" class (the smallest class) are sampled; where 85% of them are randomly selected for the training set, and the remaining 15% are selected for the test set. For the "*crack-linear*" and "*crack-network*" classes, 60% and 5% of total images in each of these two classes are randomly selected for the training and test sets, respectively.

Deep neural network structure

This research focuses on implementing deep neural networks to build automated distress detection and classification methodology. Accordingly, Convolutional Neural Networks (CNNs) with different structures and combinations of layers are developed and tested using the collected pavement dataset. The proposed CNN models are mainly based on VGG networks (Simonyan and Zisserman, 2014). Fig. 3 (a) illustrates an example of a VGG16 network with 13 convolutional layers, five pooling layers, and three fully-connected layers (Simonyan and Zisserman, 2014), implemented for *ImageNet Large Scale Visual Recognition Competition* (ILSVRC, 2015).

In this research, a variety of VGG-based CNN models were tested. Due to the size of the collected dataset, the deeper CNN model does not improve the classification accuracy but consumes more computation power and time. Therefore, a nine-layer deep neural network, with about 4.3 million parameters, is implemented. Fig. 3 (b) shows the structure of one of the proposed CNN models for pavement distress classification in this research while the input image size is 150 × 150 × 3 and the number of classes is six.

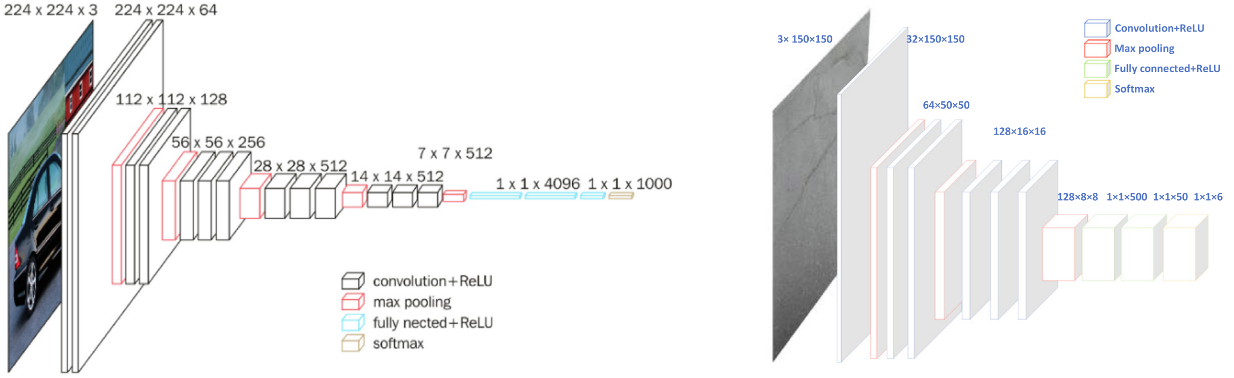
The proposed CNN with input size as 150 × 150 pixels, Fig. 3 (b), has three convolutional blocks; the first block consists of one convolutional layer with 32 channels followed by a max-pooling layer, the second block has two convolutional layers with 64 channels followed by a max-pooling layer, and the last block has three convolutional layers with 128 channels fol-

Table 1
Images Partitioning Scenarios.

Dataset Number	Original Image	Number of Original Images	Reduced Image Size	Sub-Image Size	Number of Sub-Images	Number of Image of Clean Class	Number of Images of Other Classes
1	3000 × 4000	2105	1500 × 2000	100 × 100	631,500	456,028	175,472
2	3000 × 4000	2105	1500 × 2100	150 × 150 (A)	294,700	206,199	88,501
3	3000 × 4000	2105	750 × 1050	150 × 150 (B)	73,675	49,877	23,798
4	3000 × 4000	2105	1600 × 2000	200 × 200	168,400	115,321	53,079

Table 2
Description of the Second Sub-Dataset.

Classes	Numberof Images	% of the Total Dataset	Training Set			Test Set		
			Number of Samples	Percentage of Class	Percentage of Training Set	Number of Samples	Percentage of Class	Percentage of Test Set
Clean	206,199	70%	26,806	13%	32.2%	2,062	1%	25.6%
Patch	16,025	5.4%	11,218	70%	13.5%	1,282	8%	15.9%
Pothole	2,525	0.9%	2,146	85%	2.6%	379	15%	4.7%
Crack-Linear	29,459	10%	17,675	60%	21.2%	1,473	5%	18.3%
Crack- 21.8%	Network	35,007	11.9%	21,004	60%	25.2%	1,750	5%
Marking	5,485	1.9%	4,388	80%	5.3%	1,097	20%	13.6%
Total	294,700	100%	83,237	-	100%	8,043	-	100%



(a) An example of VGG 16 while the input image size is 224×224×3 (Simonyan and Zisserman, 2014)

(b) The proposed CNN while the input image size is 150×150×3

Fig. 3. The Structures of the Reference CNN and the proposed CNN.

lowed by a max-pooling layer. The max-pooling layers reduce the size of the feature map by picking the maximum value of a 2×2 window of neurons at the previous layer and inserting it in the reduced feature map.

In the proposed CNN, the *ReLU* (Rectified Linear Unit) function (1) is used as the activation function for all the convolutional layers and the first two layers of the fully connected network (Krizhevsky et al., 2012).

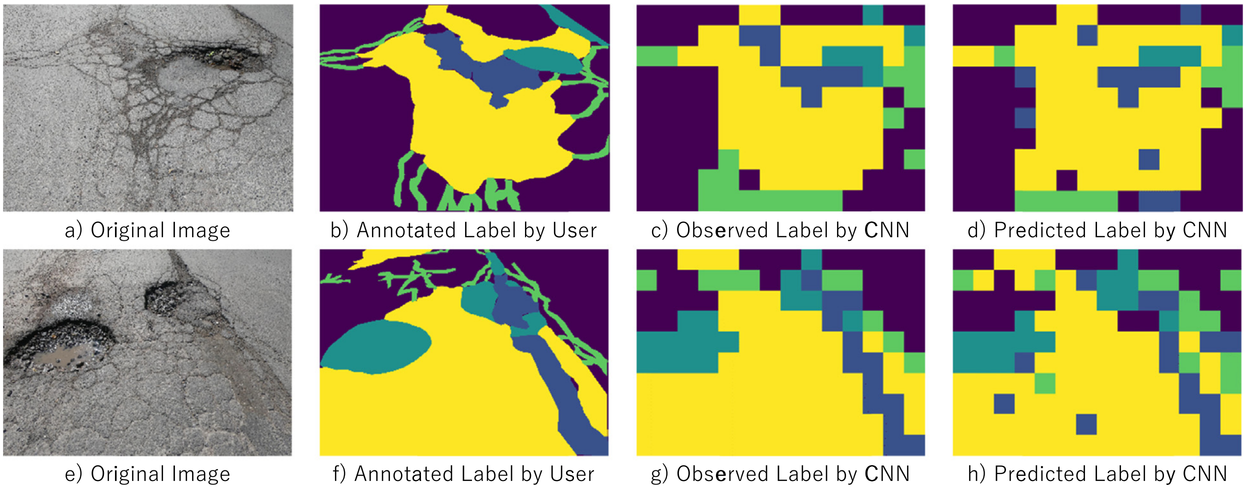
$$f(x_i) = \max(0, x_i) \tag{1}$$

where x_i is the input to the activation unit of i^{th} neuron and $f(x_i)$ is the output of the same neuron.

The output of the last convolutional block is flattened making a vector of 8192 elements. Then this one-dimensional vector is given to a fully connected network with three dense layers. The first and second dense layers of the fully connected network have 500 and 50 neurons, respectively. The output layer has six neurons, and the *SoftMax* function (2) is used as its activation function. Each *SoftMax* function calculates the probability of its corresponding class. Then the class with the maximum probability is chosen as the decision or label for any sample pavement image given to the CNN model.

$$P(Y = i|x_j) = g(x) = \frac{exp(x_i)}{\sum_{j=0}^{K} exp(x_j)} \tag{2}$$

Table 3 describes the proposed CNN model in detail. The CNN model has 4,341,824 parameters that need to be learned by using the training set of the collected dataset.


a) Original Image b) Annotated Label by User c) Observed Label by CNN d) Predicted Label by CNN

e) Original Image f) Annotated Label by User g) Observed Label by CNN h) Predicted Label by CNN

Fig. 4. Two samples of distress type detection.

Table 3
The Proposed Deep Neural Networks with Nine Layers While the Input Image Size is 150 × 150.

Layer Type	Filter Size	Number of Feature Maps	Stride	Output Size	Number of Parameters
Convolutional	3 × 3	32	1 × 1	32 × 150 × 150	320
Max-Pooling	3 × 3	-	1 × 1	32 × 50 × 50	-
Convolutional	3 × 3	64	1 × 1	64 × 50 × 50	18,496
Convolutional	3 × 3	64	1 × 1	64 × 50 × 50	36,928
Max-Pooling	3 × 3	-	1 × 1	64 × 16 × 16	
Convolutional	2 × 2	128	1 × 1	128 × 16 × 16	32,896
Convolutional	2 × 2	128	1 × 1	128 × 16 × 16	65,664
Convolutional	2 × 2	128	1 × 1	128 × 16 × 16	65,664
Max-Pooling	2 × 2	-	1 × 1	128 × 8 × 8	-
Flattening	-	-	-	128 × 8 × 8 = 8192	-
Fully Connected	500	-	-	500	4,096,500
Fully Connected	50	-	-	50	25,050
Output	6	-	-	6	306
Total Number of Parameters	-	-	-	-	4,341,824

Table 4
Regularization Scenarios.

Layer	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Convolutional	BN	BN	BN	BN	BN	-
Max-Pooling	-	-	DO (rate = 0.5)	DO (rate = 0.35)	-	DO (rate = 0.5)
Convolutional	BN	BN	BN	BN	BN	-
Convolutional	BN	BN	BN	BN	BN	-
Max-Pooling	-	-	DO (rate = 0.5)	DO (rate = 0.35)	-	DO (rate = 0.5)
Convolutional	BN	BN	BN	BN	BN	-
Convolutional	BN	BN	BN	BN	BN	-
Convolutional	BN	BN	BN	BN	-	-
Max-Pooling	-	-	DO (rate = 0.5)	DO (rate = 0.35)	-	DO (rate = 0.5)
Flattening	-	-	-	-	-	-
Fully Connected	DO (rate = 0.5)BN	DO (rate = 0.35)BN	DO (rate = 0.5)BN	DO (rate = 0.35)BN	-BN	DO (rate = 0.5)-
Fully Connected	DO (rate = 0.5)BN	DO (rate = 0.35)BN	DO (rate = 0.5)BN	DO (rate = 0.35)BN	-BN	DO (rate = 0.5)-
Output	-	-	-	-	-	-

CNN regularization scenarios

In addition to the dataset partitioning and the network structure, tuning the regularization hyperparameters of the proposed CNN is a key point for having a generalized model that can perform well on the test set. The two important and widely applied operations for the regularization of CNN are *Dropout* and *Batch Normalization*.

Dropout (DO) operation is used to avoid overfitting in neural networks by temporarily and randomly removing some of the learned parameters from the network (Srivastava et al., 2014). *Batch Normalization* (BN) is used to stop a neuron from going to the saturated mode by adjusting the input to its activation function (Ioffe and Szegedy, 2015). Furthermore, for the classes that have a limited number of images, and collecting more images is not feasible, *Image Augmentation* is used (Perez and Wang, 2017). An *Image Augmentation* technique modifies an image and inserts it into the dataset as a new image. The modification includes flipping on vertical or horizontal edges, rescaling, rotating, whitening, shifting, etc.

Six scenarios, presented in Table 4, are defined to tune and evaluate the regularization hyperparameters. In scenarios 1 and 2, the effect of different dropout rates applied to fully connected layers has been compared, while in scenarios 3 and 4, the performance of applying different dropout rates to the pooling layers has been compared. In scenarios 5 and 6, the effects of using only *Batch Normalization* or only *Dropout* have been analyzed.

Experimental results and performance evaluation

The performance of the proposed CNN model in the context of pavement distress detection and classification is evaluated by training and testing using the collected and labeled images in Montreal Pavement Dataset (MPD). In this research, *Keras*, with *TensorFlow* backend, is used to train and test different CNN structures and to tune all the hyperparameters. The average time to train one of the proposed deep neural network models on the available GPU (GeForce GTX1060 3 GB) was two days. However, testing a new sample image is executed in real-time (a batch of 91,000 images were executed in one minute). The parameters of the deep neural network are obtained by optimizing the cost function using the *Adadelta* method (Zeiler, 2012). *Adadelta* uses an adaptive learning rate method and has a lower computational cost than *Stochastic Gradient Descent* (SGD).

Distress type detection and classification

In Fig. 4 two examples of distress type detections have been illustrated. Fig. 4 (a) shows the pavement with some cracks (network and linear), a pothole, and patching. Fig. 4 (b) shows the annotated images. The yellow polygons are for "*crack-network*", the light green polygons are for *"crack-linear*", the dark blue polygons are for the "*patching*", and the dark green polygons are for "*pothole*". In Fig. 4 (c), the original labeled image is partitioned into 10 × 14 sub-images and then for each sub-image, the most frequent intensity (label) in the square segment is chosen as the label of that segment. These sub-images are used as the input data to the CNN model. Finally, Fig. 4 (d) shows the output of the proposed CNN while 10 × 14 sub-images are concatenated and make a single image. Fig. 4 (e-h) show another example of distress type detection. Comparing figures (c) by (d) and comparing (g) by (h) generates true or false predictions that can be used for computing the error measures.

Using the sub-images as the input of the CNN model, instead of using the original images, helps in decreasing the size of the CNN model, increasing the number of training samples, and reducing the computation time of training the CNN model. Besides, re-merging the labeled sub-images and building the image of predicted distress provide some information about the shape of distress as well (Fig. 4 (d & h)).

Error measures definitions

To evaluate the performance of the proposed CNN model, a variety of error measures are employed or defined in Table 5. These definitions use the concept of positivity and negativity of image classes. The clean images are considered as a negative class and all the other distress types are considered as positive classes. In this context, true negative (t_n) is the number of the clean images that are predicted correctly, false positive (f_p) is the number of clean images that are predicted as positive or having distress, and false negative (f_n) is the number of positive images with any kind of distress that are predicted as clean.

The true positive value (t_p) is split into two sub-factors: true positive-true classified (t_{p-t_c}) is the number of images that have been predicted as positive and the type of distress is detected correctly, and true positive-false classified (t_{p-f_c}) is the number of positive predicted images that are miss-classified as another distress type. This concept helps to accurately identify the source of the error. By considering these five factors, nine different error measures are defined including *Recall* or *Ture Positive Rate* (TPR), *Precision* or *Positive Predictive Value* (PPV), *F1-Score*, and *Accuracy*. The *F1-Score* is the harmonic mean of *Recall* and *Precision*. All error measures can be calculated for each class individually or of the entire data.

Evaluation of the regularization scenarios

The six regularization scenarios, mentioned in section 3.5, were simulated and results over the second partitioning scenario, splitting images to 150 × 150 (A), are reported in Table 6. The *F1-Score* of 0.814 and *Accuracy* of 0.838 show that the fourth regularization scenario, using *Batch Normalization* and *Dropout* (dropout rate equal to 0.35) layers, has the best performance over the test set. Using only *Dropout* (sixth scenario) has the worst results while adding *Batch Normalization* to the same structure improved the outcomes in the third and fourth scenarios.

Evaluation of the image resizing and partitioning scenarios

Tuning the hyperparameters for resizing and splitting the original images is a trade-off between the number of produced sub-images and the level of information that each sub-image contains. In this section, the CNN model, with the same structure as section 3.4, is trained with four datasets mentioned in section 3.3.

Table 7 shows the error measures over the training and test sets. The second dataset, resizing images by half and then splitting images into the sub-images with 150 × 150 pixels, has the best performance among the other datasets in terms of every error measure over the test set. After that, the fourth dataset 200 × 200 has the second-best performance. Between the first dataset (containing many sub-images with low-resolution) and the third dataset (containing fewer sub-images with high-resolution), the results do not suggest any preference since both perform inadequately.

Table 5
Error Measures.

Recall or True Positive Rate (TPR)	False Negative Rate (FNR)	False Classification Rate (FCR)
$\boldsymbol{\frac{t_{p-t_c}}{observedpositive}}$	$\frac{f_n}{observedpositive}$	$\frac{t_{p-f_c}}{observedpositive}$
Precision or Positive Predictive Value (PPV) $\boldsymbol{\frac{t_{p-t_c}}{predictedpositive}}$	**False Discovery Rate (FDR)** $\frac{f_p}{predictedpositive}$	**Positive Discovery-False Classification (PDFC)** $\frac{t_{p-f_c}}{predictedpositive}$
True Negative Rate (TNR) $\boldsymbol{\frac{t_n}{observednegative}}$	**F1-Score** $2 \times \frac{PPV \times TPR}{PPV+TPR} = \frac{1}{\frac{1}{2}\times\left(\frac{1}{TPR}+\frac{1}{PPV}\right)}$	**Accuracy (ACC)** $\frac{t_{p-t_c}+t_n}{totalsamples}$

Table 6
Fine Tuning of Regularization Hyper-parameters.

		Error Measures								
Set	Scenario	TNR↑	TPR↑	FNR↓	FCR↓	PPV↑	FDR↓	PDFC↓	F1-Score↑	ACC↑
Training	1	0.911	0.759	0.054	0.187	0.768	0.043	0.189	0.764	0.808
	2	0.931	0.716	0.074	0.210	0.747	0.034	0.219	0.731	0.785
	3	0.805	0.607	0.103	0.290	0.614	0.094	0.293	0.610	0.671
	4	0.928	0.832	0.032	0.136	0.831	0.034	0.136	0.831	0.863
	5	0.874	0.713	0.074	0.212	0.724	0.061	0.215	0.718	0.765
	6	0.791	0.660	0.078	0.262	0.646	0.097	0.257	0.653	0.702
Test	1	0.910	0.740	0.062	0.198	0.764	0.032	0.204	0.752	0.784
	2	0.929	0.681	0.085	0.234	0.725	0.026	0.249	0.702	0.744
	3	0.806	0.598	0.108	0.293	0.624	0.070	0.306	0.611	0.651
	4	0.930	**0.807**	0.041	0.152	0.821	0.025	0.155	**0.814**	**0.838**
	5	0.876	0.732	0.068	0.201	0.750	0.044	0.206	0.741	0.769
	6	0.796	0.652	0.083	0.266	0.660	0.071	0.269	0.656	0.689

The arrows, next to the performance measures, show if greater (↑) or lower (↓) results are better.

Performance evaluation of distress type classification

Based on the results of the two previous sections, the combination of the fourth regularization scenario and the second partitioning scenario outperforms the other models. The class-wise performance evaluation of the proposed CNN over the six different classes of the test set are discussed in Table 8. The first six columns and rows show the classification confusion matrix where the rows are the observed classes, and the columns are the predicted classes. The first element, from the first column-first row, is the number of images that are correctly classified as "*clean*" (true negative). The other elements of the first column report the number of images that have been classified as "*clean*" class (false negative) and the first row reports the number of images that belong to the "*clean*" class but are classified as others (false positive). The diagonal elements are the correct distress detection (images that are correctly classified as positive), and the rest of the elements (the non-diagonal elements excluding the first row and column) are true positive but false classified. By considering these factors, TNR, TPR (*Recall*), FNR, FCR, PPV (*Precision*), and *F1-Score* of each class have been calculated.

The TNR (only defined for the "*clean*" class) is about 93%, while the TPR of the distress classes ranges from 75.7% (*pothole*) to 84.1% (*patch*). The results over the test set show that the proposed system has an *F1-Score* around 86% for "*marking*" class, 80.2% for "*patch*", 80.8% for "*pothole*", and about 79.6% and 81.3% for the two "*crack-linear*" and "*crack-network*" classes.

The error regarding the *Recall* (TPR) value (which is $1 - TPR$), is split into two categories: *False Negative Rate* (FNR) and *False Classification Rate* (FCR). For example, the TPR of the "*crack-linear*" is 79.4%, therefore the error is 21.6% which is split into 3.1% corresponding to FNR and 17.4% corresponding to FCR. This amount of false classification rate is because of having a "*crack-network*" class where many samples of these two classes, "*crack-linear*" and "*crack-network*", have been classified as one another. The "*crack-linear*" and "*crack-network*" classes can be merged and create an additional class, called the "*crack-merged*" class. The result of this merging is presented in the last two rows of Table 8, where the detection rate (TPR) of any type of cracks is 92.6%. Merging two similar classes of cracks increases total *Recall* value from 80.7% to 86.7% and total *F1-Score* from 81.4% to 87.5%.

Due to the labeling techniques implemented in this study, which is based on pixel annotation, the CNN models have been integrated by using manual segmentation of images into sub-images. On the contrary, the RCNN models, implemented in a few other studies, require a dataset labeled using bounding boxes and cannot be applied to the collected dataset for comparison purposes. Nonetheless, in Table 9, the CNN model of this study has been compared by two CNN models of the GAPs as the reference study (Eisenbach et al., 2017).

Table 7
Prediction Results of All Four Dataset.

		Error Measures								
Set	Sub-Image Size	TNR↑	TPR↑	FNR↓	FCR↓	PPV↑	FDR↓	PDFC↓	F1-Score↑	ACC↑
Training	100 × 100	0.837	0.659	0.087	0.254	0.660	0.086	0.254	0.659	0.721
	150 × 150 (A)	0.928	0.832	0.032	0.136	0.831	0.034	0.136	0.831	0.863
	150 × 150 (B)	0.848	0.689	0.095	0.216	0.710	0.067	0.223	0.699	0.736
	200 × 200	0.837	0.719	0.061	0.220	0.711	0.071	0.218	0.715	0.755
Test	100 × 100	0.847	0.655	0.083	0.261	0.672	0.060	0.268	0.664	0.709
	150 × 150 (A)	0.930	0.807	0.041	0.152	0.821	0.025	0.155	**0.814**	**0.838**
	150 × 150 (B)	0.832	0.674	0.111	0.216	0.715	0.056	0.229	0.694	0.711
	200 × 200	0.832	0.724	0.059	0.217	0.728	0.054	0.218	0.726	0.750

Table 8
Detailed performance evaluation of the tuned CNN over the test set.

PredictedObserved	Clean	Patch	Pothole	Crack-Linear	Crack-Network	Marking	TNR/ TPR	FNR	FCR	PPV	F1-Score
Clean	1917	74	2	30	37	2	93.0%	-	-	-	-
Patch	98	1078	10	49	46	1	84.1%	7.6%	8.3%	76.7%	80.2%
Pothole	6	33	287	32	19	2	75.7%	1.6%	22.7%	86.7%	80.8%
Crack-Linear	46	49	10	1170	196	2	79.4%	3.1%	17.4%	79.9%	79.6%
Crack-Network	46	60	19	165	1454	6	83.1%	2.6%	14.3%	79.5%	81.3%
Marking	49	112	3	19	77	837	76.3%	4.5%	19.2%	98.5%	86.0%
Total - Using Crack-Linear and Network Classes	2162	1406	331	1465	1829	850	**80.7%**	4.1%	15.2%	82.1%	**81.4%**
Crack-Merged	92	109	29	2985		8	92.6%	2.9%	4.5%	90.6%	91.6%
Total – Using Crack-Merged	2162	1406	331	3294		850	**86.7%**	4.1%	9.2%	88.2%	**87.5%**

Table 9
CNN models performance comparison.

CNN Model	Scenario	TPR	FNR	FCR	PPV	FDR	PDFC	F1-Score	ACC
RCD	Separate Linear and Network Cracking Classes	58.9%	10.6%	30.5%	62.5%	5.0%	32.4%	60.7%	**65.9%**
ASINVOS		52.9%	21.1%	26.1%	64.1%	4.3%	31.6%	57.9%	62.3%
Proposed CNN		80.7%	4.1%	15.2%	82.1%	2.5%	15.5%	81.4%	**83.8%**
RCD	Merging Linear and Network Cracking Classes	73.1%	10.6%	16.3%	77.6%	5.0%	17.3%	75.3%	**76.5%**
ASINVOS		65.5%	21.1%	13.4%	79.4%	4.3%	16.3%	71.8%	71.7%
Proposed CNN		86.7%	4.1%	9.2%	88.2%	2.5%	9.3%	87.5%	**88.3%**

Although the results suggest that the proposed CNN model outperform the reference models (Eisenbach et al., 2017), in task-specific applications such as pavement distress detection and classification, the operations of the building, tuning, and training the best CNN model is unique to the base dataset, meaning that the number of classes and size of the dataset and each of its class play a significant role in the architecture and size of the implemented CNN.

However, it is worth mentioning that in this study, the camera selection and setup has fewer limitations than the other study, and images taken by any high-definition camera installed on the front bumper and focused on the pavement can be evaluated for pavement distress classification by the proposed CNN model.

Conclusion

This research proposes a methodology for pavement image distress classification using CNN models. Data was collected using low-cost high-definition cameras mounted on a vehicle. The camera captures GPS-tagged time-stamped images. This helps to cut the total cost of the integrated system by reducing the need for an additional positioning system.

The labeled dataset includes 2,105 high-definition images of four common distress types as well as marking and clean images. In the detection process, the calibrated CNN model efficiently detects and classifies the pavement distress type. The true positive rate of the model is 75.7% for "*pothole*", 84.1% for "*patch*", 76.3% for "*marking*", 79.4% for "*crack-linear*" and 83.1% for "*crack-network*". However, by merging linear and network crack classes, the accuracy over the merged "*crack*" class increases to 92.6%. Overall, this work shows that an embedded system integrated with a deep neural network can be installed in pavement monitoring vehicles and used for automated pavement distress type detection and classification in road facilities. Comparing to expensive LCMS, the system proposed in this study achieves similar accuracy while it is easily replicable and more economical. Besides, it benefits from using a regular sports camera since it can be installed on several vehicles and cover any corners of the city.

Different regularizations were attempted and compared, including *Image Augmentation*, *Dropout*, and *Batch Normalization*. It is found that the model with *Batch Normalization* and the lower rate of *Dropout* has the best performance. Additionally, dataset tuning has been placed where four different resizing and partitioning scenarios were developed and tested. The results indicate that the proper selection of resizing and splitting factors, which leads to different input image sizes, is crucial.

As future work, additional video records could be collected for improving training datasets towards the classes with fewer samples than others. A variety of road facilities during different illumination conditions can be included, such as bicycle facilities and highways. Despite the relatively large datasets used in this work, a more balanced dataset, having more *potholes* samples, could improve the detection accuracy. Moreover, although *crack-networks* and *potholes* could represent severe conditions in a way, image datasets from both regular (visual spectrum) and 3D LiDAR sensors could be combined to capture the depth and 3D shape of pavement distresses and estimate their severity.

Data availability

The Montreal Pavement Dataset used to support the findings of this study are available from the corresponding author upon request.

Funding statement

Funding for this project was provided in part by the Natural Sciences and Engineering Research Council.

CRediT author statement

The authors confirm contribution to the paper as follows: study conception and design: C. Zhang, E. Nateghinia, L. F. Miranda-Moreno, and L. Sun; data collection: C. Zhang and E. Nateghinia; model development: C. Zhang; analysis and interpretation of results: C. Zhang and E. Nateghinia; draft manuscript preparation: C. Zhang, E. Nateghinia, L. F. Miranda-Moreno, and L. Sun. All authors reviewed the results and approved the final version of the manuscript.

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.

References

Ayenu-Prah, A., Attoh-Okine, N. 2008. Evaluating pavement cracks with bidimensional empirical mode decomposition. EURASIP Journal on Advances in Signal Processing, 2008, 1–7.

BABKOV, V. F. 1975. Road conditions and traffic safety.

Baladi, G.Y., Novak, E., Kuo, W.-H., 1991. Pavement condition index—Remaining service life. Pavement management implementation, ASTM International.

Bray, J., Verma, B., Li, X., He, W., 2006. A neural network based technique for automatic classification of road cracks. The 2006 IEEE International Joint Conference on Neural Network Proceedings IEEE, 907–912.

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U. & Gross, H.-M. How to get pavement distress detection ready for deep learning? A systematic approach. 2017 international joint conference on neural networks (IJCNN), 2017. IEEE, 2039–2047.

Elghriany, A. F. 2016. Investigating Correlations of Pavement Conditions with Crash Rates on In-Service US Highways. University of Akron.

HUANG, Y. H. 2004. Pavement analysis and design.

Ioffe, S. and Szegedy, C. 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448–456). PMLR.

JOKELA, M., KUTILA, M. & LE, L. 2009. Road condition monitoring system based on a stereo camera. 2009 IEEE 5th International conference on intelligent computer communication and processing, IEEE, 423–428.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neu. Informat. Process. Syst. 1097–1105.

Laurent, John, Hébert, Jean François, Lefebvre, Daniel, Savard, Yves, 2012. In: 7th RILEM International Conference on Cracking in Pavements. Springer Netherlands, Dordrecht, pp. 157–167. https://doi.org/10.1007/978-94-007-4566-7_16.

Li, B., Wang, K.C., Zhang, A., Yang, E., Wang, G., 2020. Automatic classification of pavement crack using deep convolutional neural network. Int. J. Pavem. Eng. 21, 457–463.

LIN, J. & LIU, Y. Potholes detection based on SVM in the pavement distress image. 2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science, 2010. IEEE, 544–547.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T. & Omata, H. 2018. Road damage detection using deep neural networks with images captured through a smartphone, arXiv preprint arXiv:1801.09454.

Mahler, D. S., Kharoufa, Z. B., Wong, E. K., Shaw, L. G. J. C. A. C. & Engineering, I. 1991. Pavement distress analysis using image processing techniques. 6, 1–14.

Marques, A.G.C.S., Correia, P.L., 2012. Automatic road pavement crack detection using SVM. Lisbon, Portugal: Dissertation for the Master of Science Degree in Electrical and Computer Engineering at Instituto Superior Técnico.

Medioni, G., Lee, M.-S., Tang, C.-K., 2000. A computational framework for segmentation and grouping. Elsevier.

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

Montreal, T. C. O. 2019. Pavement condition indicators of the road network [Online]. Available: http://donnees.ville.montreal.qc.ca/dataset/condition-chaussees-reseau-routier [Accessed].

Oliveira, H. & Correia, P. L. CrackIT—An image processing toolbox for crack detection and characterization. 2014 IEEE international conference on image processing (ICIP), 2014. IEEE, 798–802.

Ouma, Y.O., Hahn, M., 2016. Wavelet-morphology based detection of incipient linear cracks in asphalt pavements from RGB camera imagery and classification using circular Radon transform. Adv. Eng. Informat. 30, 481–499.

Perez, L. & Wang, J. J. A. P. A. 2017. The effectiveness of data augmentation in image classification using deep learning.

Simonyan, K. & Zisserman, A. J. A. P. A. 2014. Very deep convolutional networks for large-scale image recognition.

Siriborvornratanakul, T. 2018. An automatic road distress visual inspection system using an onboard in-car camera. Adv. Multimedia, 2018.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1), 1929–1958.

TSAI, Y.-C. J. & LI, F. J. J. O. T. E. 2012. Critical assessment of detecting asphalt pavement cracks under different lighting and low intensity contrast conditions using emerging 3D laser technology. 138, 649–656.

Zeiler, M. D. J. A. P. A. 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

Zhang, L., Yang, F., Zhang, Y. D. & Zhu, Y. J. Road crack detection using deep convolutional neural network. 2016 IEEE international conference on image processing (ICIP), 2016. IEEE, 3708–3712.

Zou, Q., Cao, Y., Li, Q., Mao, Q. & Wang, S., 2012. CrackTree: Automatic crack detection from pavement images. Pattern Recognition Letters. 33, 227–238.