ARTICLE IN PRESS

Chinese Society of Pavement Engineering

International Journal of
Pavement Research and Technology

Journal homepage: www.springer.com/42947

An image-based system for pavement crack evaluation using transfer learning and wavelet transform

Sajad Ranjbar*, Fereidoon Moghadas Nejad, H. Zakeri

Department of civil and environmental Engineering, Amirkabir University of Technology, Tehran, Iran

Received 10 April 2020; received in revised form 13 September 2020; accepted 15 September 2020

Abstract

Automatic systems for pavement inspection can significantly enhance the performance of the Pavement Management Systems (PMSs). Cracking is the most current distress in any type of pavement. Progress of various technologies leads to a lot of effort in developing an automatic system for pavement cracking inspection. In the early image-based systems, the feature extraction process for crack classification must be done by using various image processing techniques in an expert-based system. In recent years, the new machine learning techniques such as a deep convolutional neural network (DCNN) provide more efficient models with the ability of automatic feature extracting, but these models need a lot of labeled data for training. Transfer learning is a technique that solves this problem using pre-trained models. In this research, several pre-trained models (AlexNet, GoogleNet, SqueezNet, ResNet-18, ResNet-50, ResNet-101, DenseNet-201, and Inception-v3) have been used to retrain based on pavement images using transfer learning. This study aims to evaluate the efficiency of retrained DCNNs in the detection and classification of the pavement cracking. Also, it presents a more effective algorithm based on a developed wavelet transform module with more regulizer parameters for crack segmentation. The result indicated that retrained classifier models provide reliable outputs with a range of 0.94 to 0.99 in confusion matrix-based performance, but the speed of some models is significantly higher than others. Also, the results clarified that the developed wavelet module could segment crack pixels with a high level of clarity.

Keywords: Pavement inspection; Classification; Image processing; Transfer learning; Wavelet transform

1. Introduction

Roads are one of the most important components of transportation infrastructure, providing the possibility of moving people and goods and having a direct impact on people's daily lives. Transportation agencies spend a lot of time and money for the development and maintenance of road infrastructure. Road maintenance and management increase the service life and driving experience, and enhance the road safety [1,2].

The Pavement Management System (PMS) comprises several primary phases (see Fig. 1). It plays a very important role in the road-infrastructures management system and has a direct influence on the quality and safety of roads. An efficient pavement management system leads to optimum time work planning for pavement maintenance, by a proper maintenance method, and with optimized cost. These aims become possible when pavement inspection is performed correctly [3,4]. Pavement distresses evaluation is one of the most important parts in the pavement inspection procedure. Generally, pavement distress data can be collected and assessed in three different ways that include: visual inspection, semi-automatic, and automatic systems (Fig. 1) [5].

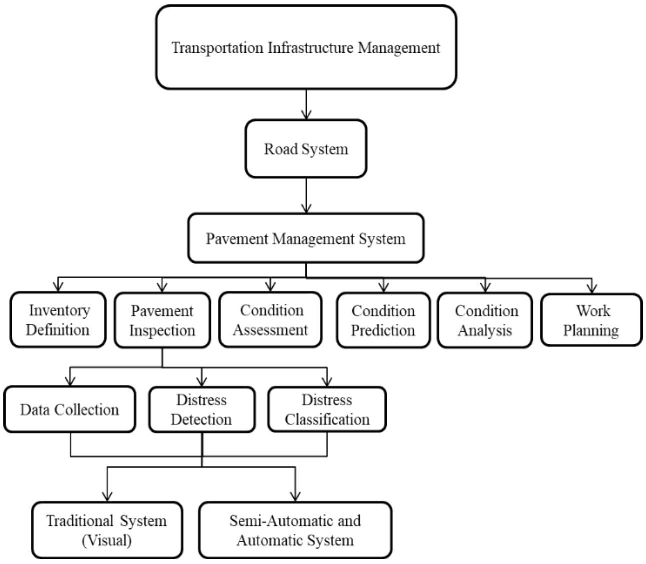

Fig. 1. Pavement inspection step in PMS.

* Corresponding author
E-mail addresses: ranjbar.sajad@aut.ac.ir (S. Ranjbara); moghadas@aut.ac.ir (F. M. Nejad,); h-zakeri@aut.ac.ir (H. Zakeri).
Peer review under responsibility of Chinese Society of Pavement Engineering.

ISSN: 1997-1400 DOI: https://doi.org/ 10.1007/s42947-020- 0098-9
Chinese Society of Pavement Engineering. Production and hosting by Springer Nature

Due to the defects of visual inspection such as high labor costs, time-consuming, unreliable results, and unsafe working conditions for laborers [5-7], most transportation departments are trying to create an automatic data collection and evaluation system. Accordingly, a lot of researches have been conducted to apply different technologies (such as image processing [5,8], laser system [9,10], ground penetration radar [11,12], fiber optic sensor [13], accelerometer sensor [14,15], ultrasonic sensors [16,17], hybrid system [18-20], and etc.) to make an automatic system for the evaluation of pavement distresses.

As can be seen in Fig. 2, pavement distresses can be divided into cracking and non-cracking distress. Cracking is the most prevalent distress on the pavement, and cracks have a significant impact on reducing the design-life [6,21,22]. The cracks can be divided into two general categories named surface cracks and linear cracks, based on their patterns [3].

In recent years, a huge volume of data (such as pavement images) is collected by transportation agencies. On the other hand, Significant growth of computing and processing power lead to developing more efficient Machine Learning (ML) approaches. Therefore, much attention has been paid to use the new ML techniques for more intelligence and efficient systems to pavement condition assessment. In the majority of ML techniques, the features are manually extracted from data. Deep learning (DL) is one of the ML technique that extract features automatically [23,24].

DCNN is one of the DL-based models that have widely been used for object detection in various applications such as pavement distress detection. Because of the high complexity of these models, they generally require large amounts of labeled data for the training process. In recent years, the transfer learning (TL) technique had been presented to solve this problem.

According to the literature, image processing is a robust tool in pavement distress evaluation. Because most of the pavement distresses appear visually such as crack, bleeding, pothole and patch. Accordingly, the use of DCNN and image processing can help transportation agencies to develop a more efficient pavement inspection system.

In this research, transfer learning-based models were developed to detect cracking and classify cracks in two classes called surface cracking and linear cracking. Also, an image processing-based approach was designed for crack segmentation by using a developed wavelet transform module. We use a pre-trained deep convolutional neural networks (PDCNN) to retrain based on our datasets by transfer learning technique. Furthermore, we use image processing techniques to segment crack pixels in the pavement image. In this image processing model, the wavelet transform is the main part of this process and other techniques such as histogram equalization (HE), image smoothing, thresholding and morphological operations are used.

2. Related work

Generally, an early image-based pavement crack evaluation system consists of three main parts: 1) segmentation, 2) feature extraction from segmented regions, and 3) detection and classification based on extracted features.

Segmentation is a method to extract the region of interest from the image. This process is vital and important for creating an efficient crack detection and classification system. There are various techniques for implementing image segmentation. Threshold-based [25,26], Matching-based [27], and region-based techniques [28] respectively are more prevalent in classical systems [5].

In the feature extraction process, significant features were extracted from image or segmented regions. These extracted features can be used for creating detection and classification models. The main problem in this section is to determine which features are more important to create a more efficient model of classification.

Spatial features [29], transform features [30], edge and boundaries features [31], shape representation features [26], and textures features [8,32] are some of the features that can be used for feature extraction step in classic crack evaluation systems. Between the mentioned feature, edge and boundaries, spatial and transform features respectively are more prevalent [5].

The last part of the classic crack assessment system is the detection and classification. Several types of ML techniques, such as supervised and unsupervised, can be used to create models for automatic crack detection and classification. In this part, supervised learning algorithms (such as neural networks [33,34], support vector machine [30,35], etc.) are more prevalent than the others.

In recent years, new systems based on DL widely considered, and DCNN is used widely for image-based pavement distress detection and classification. In these systems, feature extraction from the image and creation of the classifier model have been done

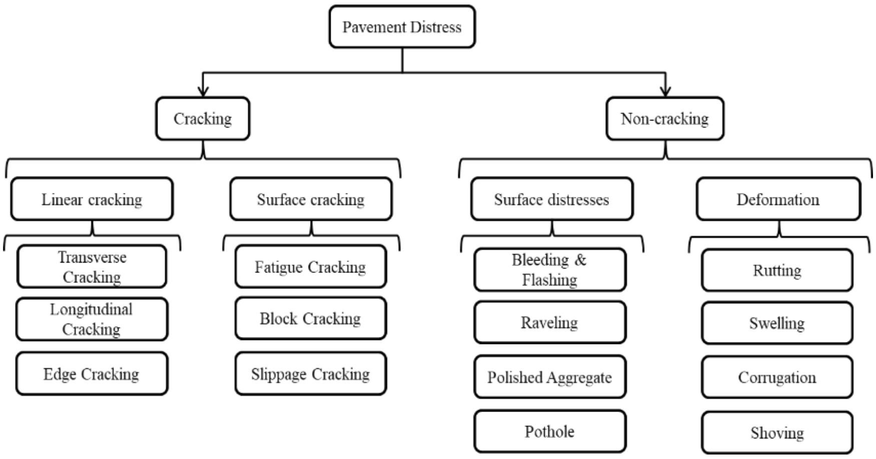

Fig. 2. General classification of Pavement distresses.

automatically by DCNN [36-40]. Generally, they don't need to do segmentation before the feature extraction process. In new systems, the segmentation is used after detection and classification for determining the important geometric parameter of distress such as length, width, area, etc.

As an important problem, DCNN-based systems need a huge dataset for the training process due to the high complexity of them. While in the majority of cases, data collection and labeling are very time consuming and costly. TL technique solves this problem and makes it possible to create DCNN based on smaller datasets [7,41,42].

According to this ability, TL-based systems have received more attention to creating an efficient image-based system for pavement crack detection and classification. On the other hand, choosing an appropriate pre-trained model is very important to create a more efficient system. Accordingly, it is necessary to evaluate the performance of the various pre-trained model.

3. Methodology

The procedure of this research consists of two main stages that include 1) Crack detection and classification, 2) Crack segmentation. As can be seen in Fig. 3, the input image is firstly analyzed by DCNN model to detect and classify the cracks. This model has been created by retraining the PDCNN using TL. In the next stage, the map of cracks is extracted by the use of wavelet transforms and other image processing techniques such as HE, image smoothing, thresholding, and morphological operations.

3.1. DCNN and TL

DCNN is a special form of deep neural networks that were designed to process data that have multiple arrays and grid-like shape such as image [43-45]. In recent years, a few research As shown in Fig. 4, the DCNN is the main part of stage1, and wavelet transform is the core of the stage2. projects (such as [41,44,46-48]) have been carried out to apply DCNN for automatic detection and classification of pavement cracking based on the pavement images.

The structure of CNNs consisted of the two main stages: feature extraction and classification. Generally, there are several convolutional and pooling layers in feature extraction stage and there are fully connected layers in classification stage.

Convolutional layers are made from several feature maps, and each unit of feature map is made from convolving a small region in input of layer. This small region is called the local receptive field.

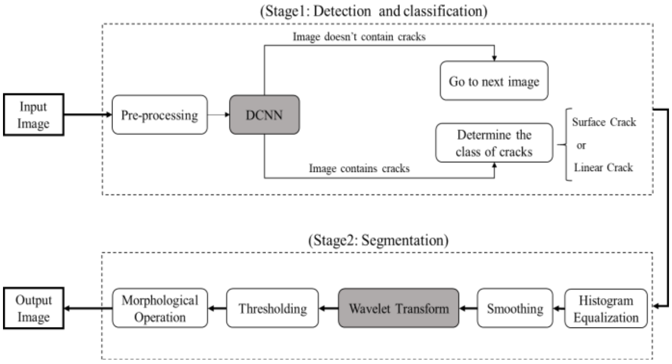

Fig. 3. Methodology of research.

As can be seen in Fig. 4, a new feature map is created by sliding a local receptive field over the input. Unlike the traditional neural network, each neuron in the layers is not connected to all the nodes (neurons) in the previous layer, but is only connected to the nodes in the local receptive field. Also, In a feature map, each unit generated with same local receptive field (weight), and this property called weight sharing [41,43,45,49].

Polling layers are commonly used after convolutional layers. These layers were generated by sliding a local receptive field on feature maps in previous layer. This operation leads to reduce the dimentionality of feature maps and to simplify the network. These layers are necessary to reduce the computational time and overfitting issues in CNN. Pooling operation can be performed in various types such as geometric average, harmonic average, maximum pooling, etc. [41,43,45,49,50] Max-pooling is one of the most prevalent pooling processes presented in Fig. 5.

Fully connected layers are placed after a sequence of convolution and pooling layers that can be one or more layers. This part of CNN is used to create a classifier based on the extracted feature from the data in the pervious stage. [41,43,45,49].

An example of the DCNN structure based on the objective of this research (classify and detect pavement cracks) has been shown in Fig. 6.

There are two methods for applying DCNN models that include: training from scratch and performing transfer learning using pre-trained models. If the first method (training from scratch) was applied for training a DCNN model, it would be necessary to use

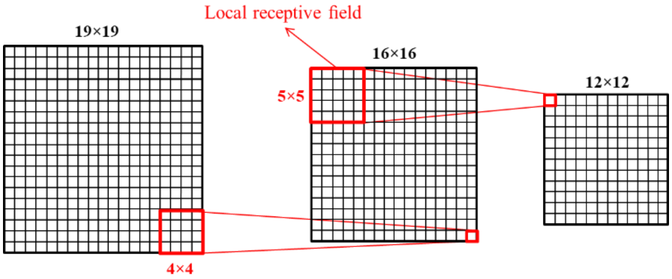

Fig. 4. Convolution process.

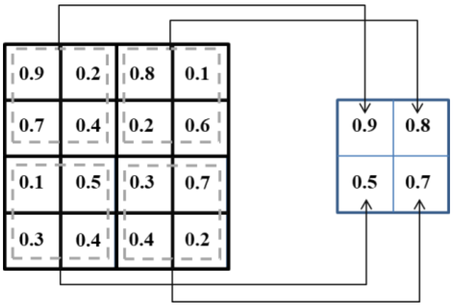

Fig. 5. An example of max-pooling operation.

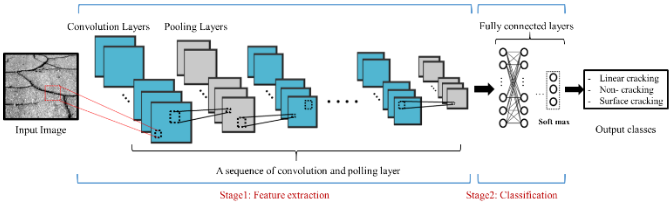

Fig. 6. CNN structure.

massive amounts of data, which are a time-consuming and costly procedure in a real application. Also, the designing of the DCNN structure to achieve proper results is a big challenge because there are many hyperparameters that affect the efficiency of DCNNs. Some of these hyperparameters includes The depth of DCNN (which includes the number of convolutional, pooling, and fully-connected layers), the number of filters, stride (step-size that the local receptive field must be moved), pooling locations and sizes, and the number of nodes in fully-connected layers [43,51].

In the other method (TL), one of the PDCNN models is being used to retrain based on new dataset. Actually in TL process, the ability of pre-trained model in feature extracting and learning the predictive rules is used to create efficient classifier model on the new dataset [7,41,52]. The details of applied transfer learning in this research have been presented in Fig. 7.

In recent years, several PDCNN is generated for various applications. The information on the more prevalent PDCNN is presented in Table 1. As shown in Table 1, the PDCNNs have different depth, number of layers and parameters, and input sizes, which have resulted from the different architecture of networks and hyper-parameters. It should be noted that the more depth and parameters lead to increase the complexity of the model, and this issue increases the overfitting probability when the trainset was not very big. Accordingly, SqueezNet and GoogleNet have lower complexity than the others. In this research, we use the mentioned pre-trained model for detection and classification of pavement cracks, and compare their performance to determine the more effective model.

3.2. Wavelet transform

The wavelet transform is a mathematical approach using localized wave like functions (wavelets) to convert the signal information into a more useful information [61]. This analysis has a wide application for image compression [62], denoising [63], and feature extraction [21]. Accordingly, extensive research has been performed for damage and crack detection [64-67] and texture analysis [8, 68-70] based on wavelet transform in civil engineering applications.

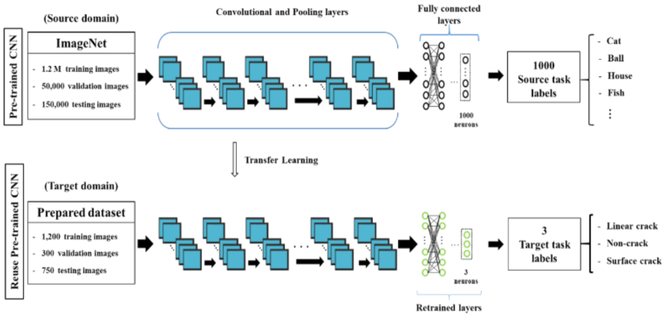

Fig. 7. TL procedure.

In the wavelet transform process, an images is decomposed into a range of frequency, and extract two sets of frequency with a high-pass and low-pass filter. In this process, an image can be decomposed in the space V_k, W_k^H, W_k^V, and W_k^D as follows [30,61,68]:

$$\begin{aligned} f_{k-1}(x,y) &= V_k + W_k^H + W_k^V + W_k^D \\ &= 2^{-k}\Big[\sum\nolimits_{m,n} LL_k(m,n)\varphi\left(2^{-k}x-m\right)\varphi\left(2^{-k}y-n\right) \\ &\quad + \sum\nolimits_{m,n} HL_k(m,n)\varphi\left(2^{-k}x-m\right)\psi\left(2^{-k}y-n\right) \\ &\quad + \sum\nolimits_{m,n} LH_k(m,n)\psi\left(2^{-k}x-m\right)\varphi\left(2^{-k}y-n\right) \\ &\quad + \sum\nolimits_{m,n} HH_k(m,n)\psi\left(2^{-k}x-m\right)\psi\left(2^{-k}y-n\right)\Big] \end{aligned} \tag{1}$$

where, k is the decomposition level, m and n are image dimensions, *φ(.)* and *ψ(.)* are orthonormal bases, and LL_k, HL_k, LH_k, HH_k, are the orthogonal projections in the orthogonal spaces that are calculated as follows:

$$\begin{aligned} LL_k(i,j) &= \sum_{m,n} l(m-2i)\,l(n-2j)\,LL_{k-1}(m,n), \\ HL_k(i,j) &= \sum_{m,n} h(m-2i)\,l(n-2j)\,LL_{k-1}(m,n), \\ LH_k(i,j) &= \sum_{m,n} l(m-2i)\,h(n-2j)\,LL_{k-1}(m,n), \\ HH_k(i,j) &= \sum_{m,n} h(m-2i)\,h(n-2j)\,LL_{k-1}(m,n). \end{aligned} \tag{2}$$

where, *l(.)* and *h(.)* are low-pass and high-pass filter, respectively. Also, LL_0 is the original image.

It should be noted that W_k^H, W_k^V, and W_k^D measure variations respectively along with the horizontal, vertical, and diagonal directions based on the high-frequency range HL_k, LH_k, HH_k in the horizontal, vertical, and diagonal directions, respectively. Also, LL_k, indicates the low-frequency range that used to determine V_k.

Consequently, three high-frequency (for horizontal, vertical, and diagonal directions) and one low-frequency sub-group were extracted by applying wavelet transform decomposition. Generally, high-frequency signals are related to details in the image such as distress (cracks) in the pavement image, and the low-frequency range of signals are related to the background of images [33,64]. In this research, a wavelet module has been developed by using the combination of horizontal, vertical, and diagonal high-frequency sub-groups to extract and segment the pavement cracks.

Table 1
PCNNs characteristics.

PDCNN	Input size	Depth (number of layers)	Number of parameters (Millions)	Size of the model (MB)	Source domain	Target Domain (number of classes)
AlexNet [53,54]	(227×227×3)	8	61	227	ImageNet	1000
SqueezNet [55]	(227×227×3)	18	1.24	4.6		
GoogleNet [56,57]	(224×224×3)	22	7	27		
ResNet-18 [58]	(224×224×3)	18	11.7	44		
ResNet-50 [58]	(224×224×3)	50	25.6	96		
ResNet-101 [58]	(224×224×3)	101	44.6	167		
Inception-V3 [59]	(299×299×3)	48	23.9	89		
DenseNet-201 [60]	(224×224×3)	201	20	77		

4. Experimental study

As shown in Fig. 4, the procedure of the experimental study consists of two main stage:1) Crack detection and classification, 2) Crack segmentation. The details of these stages have been presented in this part of the paper.

4.1. Crack detection and classification

4.1.1. Dataset

In this experiment, 1500 pavement images with a size of 900*1000 pixels have been considered for the training process, 80 percent of these images (1200 images) were used for the training , and 20 percent (300 images) for the validation. Also, 750 pavement images (quite different from training and validation sets) were selected for testing. Thus, 66.6 percent of data (1500 images) were assigned to training process and 33.3 percent of data (750 images) were assigned for testing. The detailed information about datasets has been presented in Table 2.

All of the images were cropped from the main images with a size of 3850*2764. As can be seen in Fig. 8, we crop crack regions to collect proper images for train, test, and validation data.

Also, an image enhancement process is being done on data to provide a more comprehensible image for display and makes better input for the training process. In this step, histogram equalization was performed for enhancing image contrast. As can be seen in Fig. 9, this process usually increases the local contrast of images and provides a better image for further image analysis.

In this research, block and fatigue cracking images were put in the surface cracking class, and cracks with a linear shape such as transverse and longitude cracks were put in the linear cracking class. Since that we need to distinguish cracking occurrence, we collect images from without cracking parts of the pavement and put in the non-cracking class. Based on the data in this class, the model can determine that an image related to a cracking image or not. Some of the prepared images have been presented in Fig. 10.

Table 2
Number of images in prepared datasets.

Class	Train	Validation	Test
Surface cracking	400	100	250
Linear cracking	400	100	250
Non-cracking	400	100	250
Total	1200	300	750

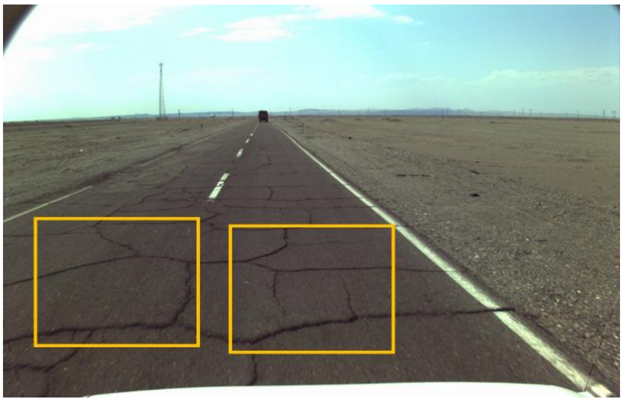

Fig. 8. A sample of captured image and cropped the cracking regions.

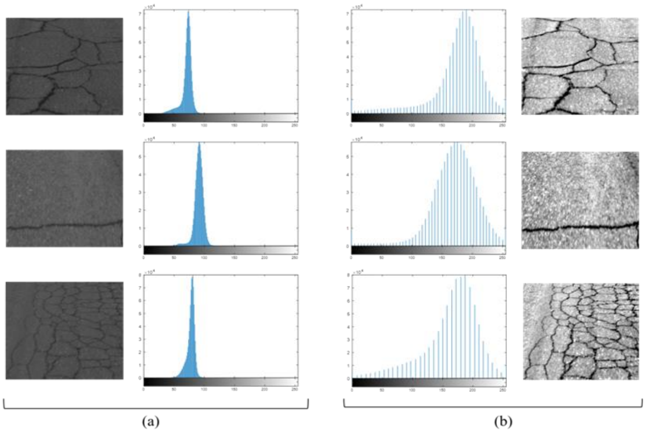

Fig. 9. (a) The original images and (b) output images after histogram equalization.

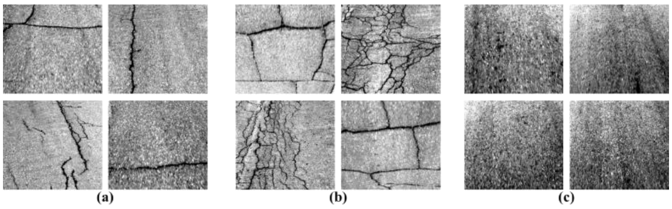

Fig. 10. Samples of the prepared dataset.

The performance of the trained models significantly depends on the quality of training datasets. For this reason, the majority of experimental works were focused on collecting and preparing the most appropriate images for the training process. Also, an attempt has been made to create datasets with a wide variety to prevent overfitting problems.

4.1.2. Retraining of pre-trained models

Before starting the training process by transfer learning, it is necessary to determine several predefined parameters that included: maximum epoch, batch size, learning rate, and momentum. These parameters have a great influence on training results and training time.

The number of epochs is defined as the number of times that the training algorithm covers all of the training data. If this number was considered too large, it could lead to an overfitted model and high training time. On the other hand, few numbers of epochs make the training process incomplete and lead to underfitted models.

In this experiment, according to the volume of datasets and processing power of the system, 15 epochs have been considered for the training process. It should be noted that the evaluation of the training progress for different PDCNNs shows that the training process was usually achieved a stable state and converged after 6 to 10 epochs (Fig. 11). Accordingly, 15 can be an appropriate number of epochs to make well-fitted models.

Another important parameter is the batch size that is defined as the number of images (data) applied for each iteration of the learning process. The value of batch size can be selected from 1 to the data set size. The high value of batch size needs high processing power. According to the processing power of the used system, the value of 15 has been considered for batch size. This means that 80 iterations of batches should be done to perform one epoch in the learning algorithm because 1200 images were used as training data (Table.2).

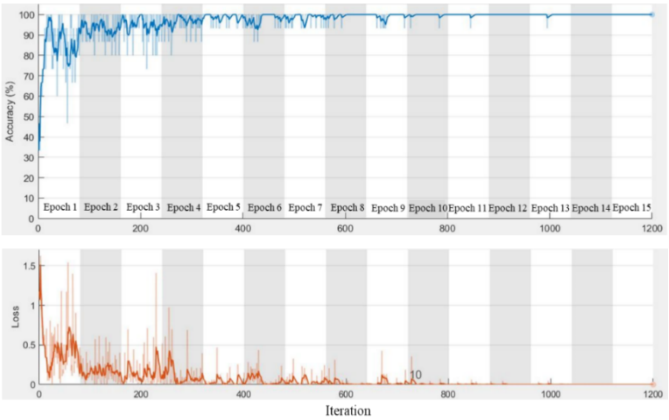

Fig. 11. A sample of training progress.

The number of iterations depends on the batch size and the number of epochs, and very large iterations lead to an overfitting model. In this experiment, 15 epochs were performed with 80 iterations of the batch sample, and 1200 iterations have been done. Other training parameters, including base learning rate and momentum, have often been considered as 0.001 and 0.9, respectively.

All of the predefined parameters have been determined equally for PDCNNs. But, for ResNet-101 in which the batch size has been determined as 12 because this model has more number of layers and parameters than other, and the limitation of processing power was led to insufficient memory space for performing the retrain process with 15 samples in each batch.

All computations were performed on a personal computer with a 64-bit operating system, 8.0 GB memory, and Intel(R) Core i7-4710HQ @ 2.50 GHz processor running a GeForce GTX 850M graphics processing unit (GPU). Also, image processing and deep learning methods were programmed and performed in MATLAB 2018b.

4.2. Crack segmentation

As shown in Fig. 3, the segmentation process consists of several steps, and in Fig. 12, an example of this process has been shown.

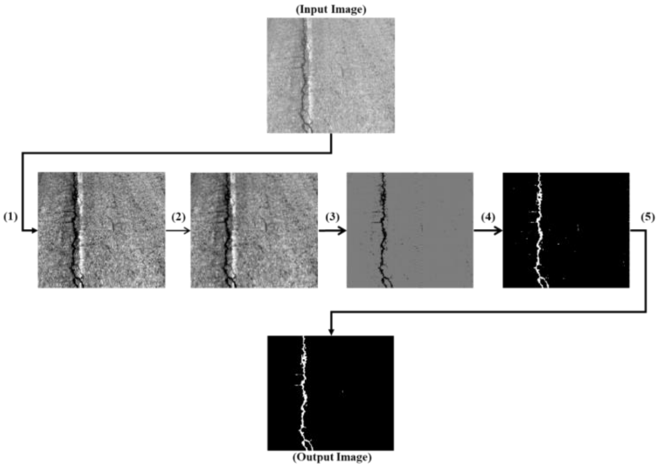

Fig. 12. Crack segmentation process; (1) histogram equalization, (2) image smoothing, (3) wavelet transform, (4) Thresholding, and (5) morphological filtering.

In the first step, the histogram equalization was done, and then image smoothing was performed by utilizing a Gaussian filter. In the image smoothing process, the intensity of smoothing is related to the value of the standard deviation of the distribution (σ) in Gaussian function in both directions (X and Y). If the value of σ was very high, the smoothing procedure led to a decrease in the image's detail, and the edges in the image were not preserved. Also, if it was very low, the reduction of noise as the aim of smoothing was insignificant. After evaluating the results of different values, 2 is the most appropriate value for both directions to smooth the images in this experiment.

In the next step, the wavelet transform has been used to enhance defects in images. In this research, a new wavelet module with three regularizer parameter has been presented as follow:

$$M_{s,k}(p,q)=\left[\left(HL_k^S(p,q)+LH_k^S(p,q)+HH_k^S(p,q)\right)\times N\right]^{\frac{1}{R}} \tag{3}$$

where, $HL_k(p,q)$, $LH_k(p,q)$, $HH_k\ (p,q)$ are high-frequency sub-bands in the horizontal, vertical, and diagonal, respectively at the position of (p, q) at the kth level. Also, S, N, and R are the regularizer parameter controlling the amount of detail in the output image.

In this experiment, the efficiency of different orthogonal wavelet bases (such as Daubechies, Coiflets, Symlets, Biorthogonal, etc.) was compared, and it was indicated that the Coiflets wavelet families had better performance than other families to highlight the local anomalies in the homogeneous surface. Also according to the literature, Coiflets wavelet filter are efficient in wavelet-base applications [71-74].

The main part of this step is determining the value of k, S, N, and R. The influence of each parameter on the results after thresholding has been presented in Fig. 13.

According to Fig. 14, several key points about regulizer parameter can be presented as follow:

1. k can be specified as a positive integer. If the number of multi-resolution levels (k) is small, in some cases, they cannot sufficiently separate distress from the background of images. However, the large number of multi-resolution levels yields the fusion effect of the anomalies and may result in false detection.
2. The parameter S can be specified as a positive integer, and the high value of this parameter can lead to a noisy and unclear result.
3. R can be specified as a positive number. The value of R has a direct relation with the amount of detail in the output image.

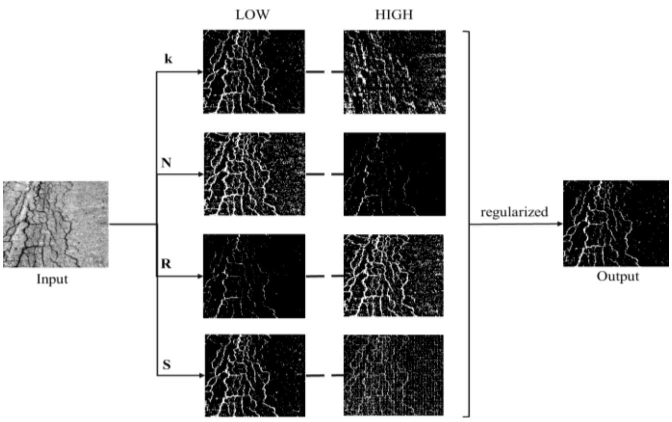

Fig. 13. Regularize the wavelet module parameter.

If the value of R is too small, the amount of separated detail is reduced and may lead to removing cracks. However, the very large value of R leads to high separated details and creates a noisy image.

4. The value of N has an inverse relation with the amount of separated detail in the output image, and different values of N have inverse effects over the effects of R. Also, N can be specified as a positive number.

After a try and error process, values of 1 to 3 for multi-resolution levels, 3 to 5 for R, 1 or 2 for S, and 2 to 4 for N are the most appropriate to segment cracks in this experiment.

In the last step, the geometric shape of defects was identified, but the output images noisy image. In the last step, the morphological filter was applied to make a clearer output by performing several tasks such as remove noise in the crack area and background.

The mathematical morphology operations are useful techniques to create more applicable output in the image processing procedure. These operations consider the images as a collection of geometric structures and process images based on shapes. In this process, first, a smaller geometrical set is defined that is known as a structure element, then each pixel in the image is adjusted based on its neighborhood and defined structure element.

Dilation and erosion are the most fundamental operations which have a different effect on images. Erosion removes pixels from the border of objects based on the structural element and shrinks the objects. With A and B as sets in Z^2, this operation is denoted by *A!B* in which A is eroded by the structuring element B as follows: [75-77].

$$A!\ B=\left\{z\,|\left(B\right)_z\subseteq A\right\} \tag{4}$$

In words, this equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

Dilation adds pixel to borders of objects based on the structural element and leads to expanded components. This operation is denoted by $A\oplus B$ in which A is dilated by the structuring element B as follows: [76,77]

$$A\oplus B=\left\{z\,|\left(\hat{B}\right)_z\cap A\neq\varnothing\right\} \tag{5}$$

$(\widehat{B})_S$ is the reflection of B about its origin that is shifting by z. The result of this operation is the set of all displacements, such that $\hat{B}$ A overlap by at least one element. Other morphological operations can be made by a combination of dilation and erosion for specific applications such as shape detection, boundary extraction, hole filling, extraction of connected components, removing small objects, and etc. [76 77]. In this research, the hole filling operation has been applied to remove holes (noises) in crack areas. This operation has been defined using dilation as follows:

$$X_k=\left(X_{k-1}\oplus B\right)\bigcap A^c \qquad k=1,2,3,... \tag{6}$$

that X_0 is a point in a hole, B is a symmetric structuring element and $A^c$is the complement of A. The algorithm terminates when $X_k=X_{k-1}$ [76,77].

Also, the opening operation has been used to removing small objects (noises) in the background to create more obvious outputs. This operation has been defined based on dilation and erosion as follows: [76,77]

$$\mathrm{A}\circ\mathrm{B}=(\mathrm{A}\ominus\mathrm{B})\oplus\mathrm{B} \tag{7}$$

In this research, the opening operation has been applied to remove objects containing fewer than 30 pixels. In most cases, these objects appear as noise in the background. Thus, the outputs are more obvious after performing this filter.

4.3. Results and discussion

In this experiment, eight PDCNNs were applied to create classifier models by using transfer learning based on the collected dataset. The speed of the models in training and testing procedure is an important parameter for choosing the appropriate pre-trained model in pavement crack detection and classification. The information on the model's speed has been presented in Table 3.

A more obvious comparison of training and testing time has been presented in Fig. 14 and Fig. 15 based on the average time for each image in seconds.

As can be seen in Fig. 14, AlexNet, SqueezNet, GoogleNet, and ResNet-18 are significantly faster than others. These models are faster because the number of training parameters are less than other models (Table.1). In other words, the complexity of this model is less than others. Accordingly, in this experiment, SqueezNet is the fastest model, and DenseNet-201 is the lowest model.

As can see in Fig. 15, more simple models (with regard to the architecture of models, number of layers, number of parameters, etc.) are faster than other models. In the testing process, similar to training, SqueezNet is the fastest model with 0.017 seconds per image.

In addition to training and testing speed, the performance of models in the detection and classification of the pavement's cracks is very important. The performance of models has been evaluated using the confusion matrix. This matrix determines the performance indices for each class using four components that

Table 3
Time spent in training and testing process

Pre-trained CNN	Training Time	Train for each image (on average)	Testing Time	Test for each image (on average)
AlexNet	578.333	0.482	14.051	0.019
SqueezNet	460	0.383	12.898	0.017
GoogleNet	764.667	0.637	18.138	0.024
ResNet-18	773	0.644	15.851	0.021
ResNet-50	2333	1.944	30.984	0.041
ResNet-101	4587	3.823	40.462	0.054
DensaNet-201	15489	12.908	66.964	0.089
Inception-V3	3541	2.951	39.512	0.053

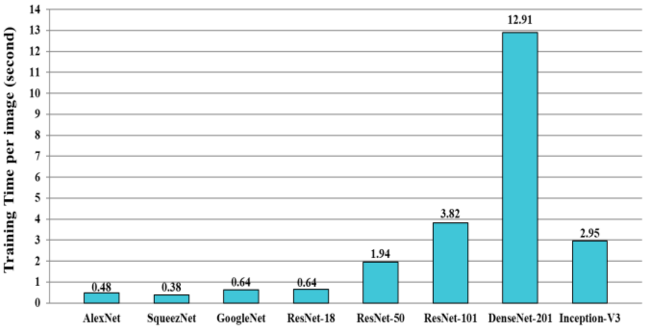

Fig. 14. Time spent in the training process.

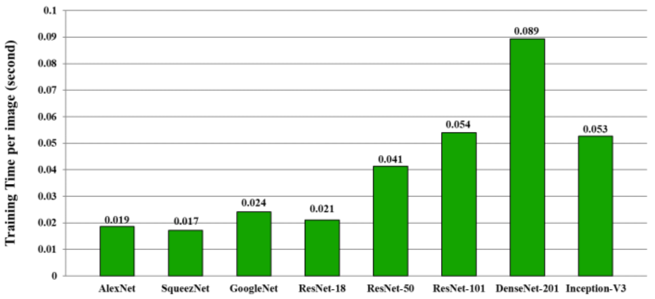

Fig. 15. Time spent in the testing process.

including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

According to confusion matrix components, different metrics can be defined for the evaluation of the models' performance. In this study, five metrics have been used to assess and compare the crack detection and classification performance, including accuracy, sensitivity, specificity, precision, and F-score. The value of these metrics has been calculated for each class in the models and has been presented in Table 4.

Accuracy is a prevalent parameter to judge the performance of classification models, and indicate how often a classifier model is correct. This parameter is calculated based on the true positive and true negative ratio of the total as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{8}$$

As can be seen in Fig. 16, the accuracy of classes is in the range of 0.957 to 1 for A series of the model. Also, the non-cracking class has the highest value in all of the models, and the linear cracking class has the lowest accuracy. The general accuracy value of models is in the range of 0.972 to 0.991.

By comparing the general accuracy of the trained model, it is determined that SqueezNet and GoogleNet models have better performance than other models with 0.991 and 0.989 value of accuracy, respectively, and ResNet-101 has the lowest accuracy with 0.972 value of accuracy.

Sensitivity or recall or true positive rate is one of classifier model performance metrics that indicates when an image is actually in a specified class, and what is the probability of classifying in the correct class. This parameter is calculated as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

Table 4
Performance of the classifier model in crack detection

Models	Class	Accuracy	Sensitivity	Specificity	Precision	Fscore
AlexNet	Linear-Cracking	0.968	0.951	0.977	0.953	0.952
	Non-Cracking	0.996	0.999	0.995	0.991	0.995
	Surface-Cracking	0.969	0.951	0.978	0.957	0.953
	General	0.978	0.967	0.983	0.967	0.967
SqueezNet	Linear-Cracking	0.986	0.989	0.985	0.970	0.980
	Non-Cracking	0.999	0.999	0.999	0.997	0.998
	Surface-Cracking	0.987	0.969	0.995	0.991	0.980
	General	0.991	0.986	0.993	0.986	0.986
GoogleNet	Linear-Cracking	0.984	0.980	0.985	0.971	0.975
	Non-Cracking	0.999	0.997	1.000	1.000	0.999
	Surface-Cracking	0.984	0.973	0.990	0.980	0.977
	General	0.989	0.984	0.992	0.984	0.984
ResNet-18	Linear-Cracking	0.968	0.964	0.971	0.943	0.953
	Non-Cracking	0.997	0.999	0.996	0.992	0.995
	Surface-Cracking	0.972	0.943	0.986	0.971	0.957
	General	0.979	0.968	0.984	0.969	0.968
ResNet-50	Linear-Cracking	0.961	0.932	0.976	0.951	0.941
	Non-Cracking	0.996	1.000	0.994	0.988	0.994
	Surface-Cracking	0.965	0.952	0.972	0.944	0.948
	General	0.974	0.961	0.981	0.961	0.961
ResNet-101	Linear-Cracking	0.957	0.948	0.962	0.926	0.937
	Non-Cracking	0.996	0.988	1.000	1.000	0.994
	Surface-Cracking	0.961	0.936	0.974	0.947	0.942
	General	0.972	0.957	0.979	0.958	0.957
DenseNet-202	Linear-Cracking	0.976	0.964	0.982	0.964	0.964
	Non-Cracking	1.000	1.000	1.000	1.000	1.000
	Surface-Cracking	0.976	0.964	0.982	0.964	0.964
	General	0.984	0.976	0.988	0.976	0.976
Inception-V3	Linear-Cracking	0.977	0.972	0.980	0.960	0.966
	Non-Cracking	1.000	1.000	1.000	1.000	1.000
	Surface-Cracking	0.977	0.960	0.986	0.972	0.966
	General	0.985	0.977	0.989	0.977	0.977

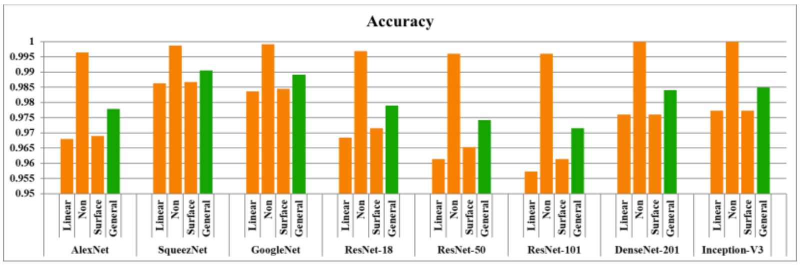

Fig. 16. Performance of models according to accuracy criterion.

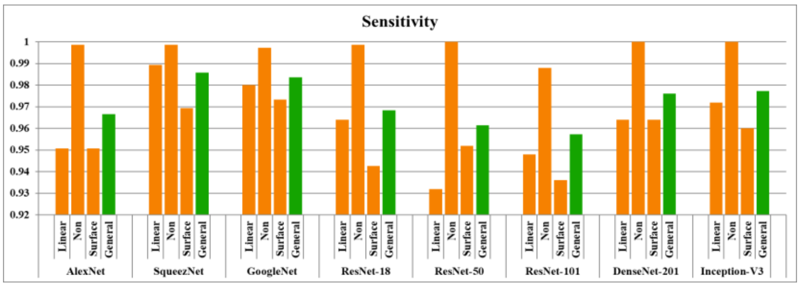

Fig. 17. Performance of models according to sensitivity criterion.

According to Fig. 17, the sensitivity of classes in the trained models are in the range of 0.932 to 1, and the non-cracking class has the highest value of sensitivity in all of the models with a range of 0.988 to 1. The evaluation of the other two classes (linear and surface cracking) shows various results

The general sensitivity of models is in the range of 0.957 to 0.986. The SqueezNet and GoogleNet have the highest value of sensitivity with 0.986 and 0.984, respectively. The ResNet-101 has the lowest sensitivity with 0.957.

Specificity or true negative rate has a similar concept to sensitivity, but for the negative state of model answers. This parameter determines the probability of the correct answer when the image is actually not in the specified class and is calculated as follows:

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \tag{10}$$

As shown in Fig. 18, the specificity is in the range of 0.962 to 1. In all of the models, the non-cracking class has the highest value of specificity. Also, in most of the models, the specificity of the surface cracking class is higher than the linear class.

Precision or positive predictive value is one of the important parameters for evaluating the performance of classifier models. This parameter indicates that when the model classifies an image in a specific class, what is the probability that the result is correct. In fact, this parameter indicates the reliability of model results, and can be calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

As can be seen in Fig. 19, the precision of the non-cracking class is higher than other classes in most models (except DenseNet-201). This means that when the result of classifier models is non-cracking, the result can be accepted with more confidence. Also, in most models except for the ResNet-50, the precision of the surface cracking class is higher than the linear cracking class. According to precision, the SqueezNet has the best performance for detection surface and linear cracking with 0.991 and 0.97, respectively.

F-score, F1-score, or F-measure is defined as the weighted harmonic mean of precision and recall. This criterion provides a more realistic criterion for model performance evaluation. This parameter is calculated as follows:

$$\text{F} - \text{score} = \frac{2}{\left(\frac{1}{\text{Precision}} + \frac{1}{\text{Sensitivity}}\right)} \tag{12}$$

Fig. 20 illustrates that similar to previous performance metrics, the non-cracking class has the highest F-score in all of the models with a range of 0.989 to 1. Also, in most models, the performance of surface crack detection is better than linear crack detection or have the same performance, except Inception-V3. The best performance on detection of linear and surface cracking images is observed in SqueezNet with an F-score value of 0.979, and the worst performance is related to ResNet-18.

According to all of the performance metrics, it can be said that the performance of models with lower complexity such as SqueezNet and GoogleNet are better than other models because the simpler structure and less training parameter lead to better training procedures when the datasets are limited.

By considering all of the important parameters that indicate the efficiency of the models such as training time, testing time, and value of the F-score as a realistic criterion for crack detection performance, a comprehensive evaluation has been provided in Fig. 21. In this figure, the size of disks represents the testing time and axis X and Y related to training time per image and general F-score of the models, respectively. Accordingly, the better model has a smaller disk, upper position in the left corner. As can be seen in Fig. 21, the performance of GoogleNet and SqueezNet are better than other models, and SqueezNet has the best performance with a value of 0.076, 0.442, and 0.986 in testing and training time per image and F-score, respectively. It should be noted that because of the week performance in training time, the DenseNet-201 has not been considered in Fig. 21.

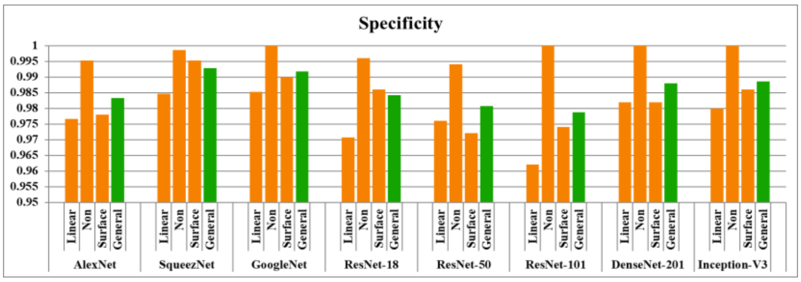

Fig. 18. Performance of models according to specificity criterion.

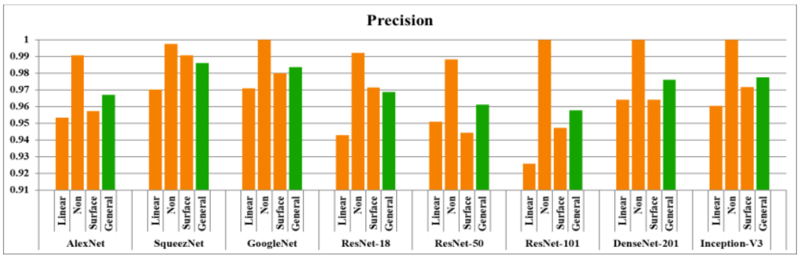

Fig. 19. Performance of models according to precision criterion.

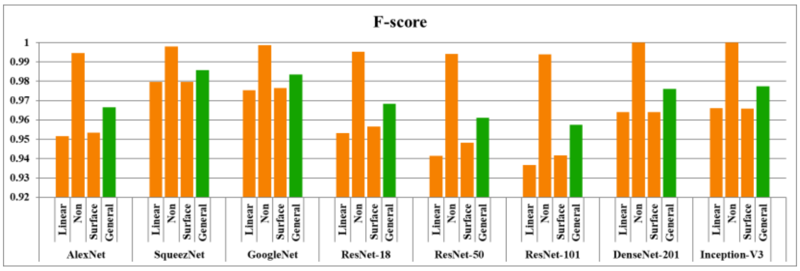

Fig. 20. Performance of models according to F-score criterion.

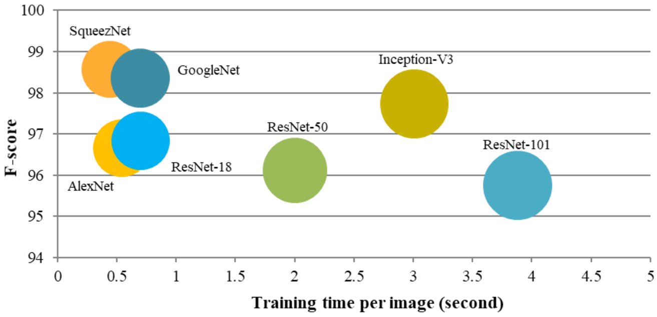

Fig. 21. The comprehensive comparison of models.

After detect and classify cracking images by using DCNN models, in the next stage, the image processing algorithm was applied to these images to segment crack pixels. Some of the results of crack segmentation have been presented in Fig. 22.

5. Conclusion and future works

The pavement management system is one of the most important parts of the road management system, and pavement inception tasks (such as distress detection) provide the main information about pavement conditions that are used in the pavement management procedure.

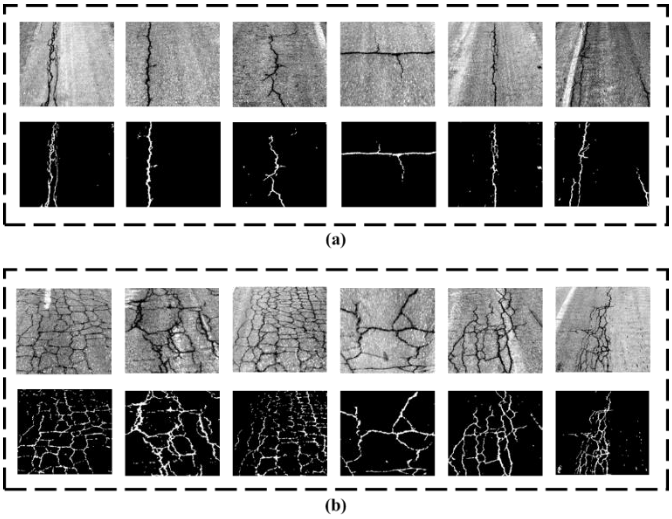

Fig. 22. Results of the crack segmentation process on the images of (a) linear cracking class and (b) surface cracking class.

In recent years, the transport agencies and researchers have focused on developing an automatic system to detect pavement distresses. On the other hand, data mining science is highly regarded because the computation power has been growing significantly and a huge volume of data was collected in the past years.

ML techniques are one of the methods that perform data mining tasks, and deep neural networks are a popular branch of it. Convolutional neural networks are a specific form of deep neural networks that have been applied for various tasks such as classification, prediction, feature extraction, etc.

In this research, eight PDCNNs (AlexNet, GoogleNet, SqueezNet, ResNet-18, ResNet-50, ResNet-101, DenseNet-201, Inception-v3) have been retrained based on pavement images with the transfer learning technique to perform classification tasks for recognition of two general types of pavement cracks, including surface cracks and linear cracks.

The main objectives of this research can be defined as follows:

Creating a TL-based model by retraining of different pre-trained models for pavement crack detection and classification

Comparing and assessing the performance of the different pre-trained model in crack detection and classification

Presenting a more efficient crack segmentation process based on a new wavelet transform module.

For creating a crack detection and classification model, a dataset was prepared, which contains three classes, including linear cracking, surface cracking, and non-cracking. However, using the transfer learning to retrain the eight PDCNNs based collected dataset. Then the performance of models was tested by the testing dataset.

The performance of models was evaluated according to training time, testing time, and crack detection performance. Based on training and testing time, AlexNet, SqueezNet, GoogleNet, and ResNet-18 have better performance than other models. In this research, five performance metrics were used to assess and compare the efficiency of crack detection models, including accuracy, sensitivity, specificity, precision, and F-score. According to performance metrics, SqueezNet and GoogleNet generally have better performance than the others. Also, the results indicated that Retraining the PDCNN by utilizing transfer learning is an efficient method for pavement crack detection and classification with a range of 0.95 to 0.99 in general models' performance.

In the second part of the research, a wavelet transform-based method was present to segment cracking regions in the pavement image. In addition to the wavelet transform, the segmentation process was performed by using various image processing algorithms such as HE, image smoothing, thresholding, and morphological operations. The presented model could segment crack pixels with a high level of clarity.

The results of this work indicate that the developed image-based system using DCNN and wavelet transformation can be used as an efficient system for detecting, classifying pavement cracks, and segmenting crack pixels in pavement images.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] C. Y. Chan, B. Huang, X. Yan, S. Richards, Investigating effects of asphalt pavement conditions on traffic accidents in Tennessee based on the pavement management system (PMS), J. Adv. Transp. 44 (3) (2010) 150-161.

[2] D. A. Noyce, H. Bahia, J. Yambo, J. Chapman, A. J. W. Bill, Incorporating road safety into pavement management: Maximizing surface friction for road safety improvements, Report Number MRUTC 04-04. Traffic Operations and Safety Laboratory, University of Wisconsin, Madison, WI, USA, 2007.

[3] M. Y. Shahin, Pavement management for airports, roads, and parking lots, Springer, NY, USA, 1994, p.2-5.

[4] F. M. Nejad, H. Zakeri, "The Hybrid Method and its Application to Smart Pavement Management," in Metaheuristics in Water, Geotechnical and Transport Engineering, ed. By X.-S. Yang, A. H. Gandomi, S. Talatahari, A. H. Alavi, Elsevier, Oxford, 2013, p. 439-484.

[5] H. Zakeri, F. M. Nejad, A. Fahimifar, Image Based Techniques for Crack Detection, Classification and Quantification in Asphalt Pavement: A Review, Archives Comput. Methods Eng. 24 (4) (2017) 935-977.

[6] V. Ananth, P. Ananthi, V. Elakkiya, J. Priyadharshini, R. Shiyamili, Automatic Pavement Crack Detection Algorithm, Inter. Innov.Res. J. Eng. Technol. 2 (1) (2017) 86-89.

[7] K. Zhang, H. Cheng, B. Zhang, Unified Approach to Pavement Crack and Sealed Crack Detection Using Preclassification Based on Transfer Learning, J. Comput. Civ. Eng. 32 (2) (2018) 04018001.

[8] B. Mataei, F. Moghadas Nejad, M. Zahedi, H. Zakeri, Evaluation of pavement surface drainage using an automated image acquisition and processing system, Autom. Constr. 86 (1) (2018) 240-255.

[9] Z. Hong, Exact extraction method for road rutting laser lines, Analysis, vol. 106070, p. 19, 2018.

[10] C. Ting, W. Weixing, Y. Nan, G. Ting, W. Fengping, Detection method for the depth of pavement broken block in cement concrete based on 3D laser scanning technology, Infrared Laser Engineering, 2 (1) (2017) 013.

[11] S. Dai and K. Hoegh, 3D step frequency GPR Asphalt pavement stripping detection: Case study evaluating filtering approaches. In Advanced Ground Penetrating Radar (IWAGPR), 9th International Workshop, Edinburgh, Scotland, 2017, pp. 1-7.

[12] S. Li, C. Yuan, D. Liu, H. Cai, Integrated processing of image and GPR data for automated pothole detection, J. Comput. Civ. Eng. 30 (6) (2016) 04016015.

[13] X. Chapeleau, J. Blanc, P. Hornych, J.-L. Gautier, J. Carroget, Use of distributed fiber optic sensors to detect damage in a pavement, 12th ISAP Conference on Asphalt pavement, Raleigh, North Carolina, USA, 2014.

[14] M. R. Carlos, M. E. Aragón, L. C. González, H. J. Escalante, F. Martínez, Evaluation of Detection Approaches for Road Anomalies Based on Accelerometer Readings--Addressing Who's Who, IEEE Transactions Intelligent Transp. Syst. 19 (10) (2018) 3334 - 3343.

[15] A. Fox, B. V. Kumar, J. Chen, F. Bai, "Multi-lane pothole detection from crowdsourced undersampled vehicle sensor data, IEEE Transactions Mobile Comput. 16 (12) (2017) 3417-3430.

[16] S. Nakashima, S. Aramaki, Y. Kitazono, S. Mu, K. Tanaka, S. Serikawa, Application of ultrasonic sensors in road surface condition distinction methods, Sensors 16 (10) (2016) 1678.

[17] R. Madli, S. Hebbar, P. Pattar, V. Golla, Automatic detection and notification of potholes and humps on roads to aid drivers, IEEE Sensors J. 15 (8) (2015) 4313-4318.

[18] J. Mehta, V. Mathur, D. Agarwal, A. Sharma, K. Prakasha, Pothole Detection and Analysis System (Pol) AS) for Real Time Data Using Sensor Networks, J. Eng. Appl. Sci. 12 (12) (2017) 3090-3097.

[19] M. Solla, S. Lagüela, H. González-Jorge, P. Arias, Approach to identify cracking in asphalt pavement using GPR and infrared thermographic methods: Preliminary findings, NDT & E Inter. 62 (1) (2014) 55-65.

[20] J. Huang, W. Liu, X. Sun, A pavement crack detection method combining 2D with 3D information based on Dempster-Shafer theory, Computer-Aided Civ. Infrast. Eng. 29 (4) (204) 299-313.

[21] Y. O. Ouma and M. Hahn, Wavelet-morphology based detection of incipient linear cracks in asphalt pavements from RGB camera imagery and classification using circular Radon transform, Adv. Eng. Informatics 30 (3) (2016) 481-499.

[22] S. Mathavan, K. Kamal, M. Rahman, A Review of Three-Dimensional Imaging Technologies for Pavement Distress Detection and Measurements, IEEE Transactions Intelligent Transp. Syst. 16 (5) (2015) 2353-2362.

[23] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature, 521 (1) (2015) 436.

[24] L. Deng, D. Yu, Deep learning: methods and applications, Foundations Trends® in Signal Process. 7 (3–4) (2014) 197-387.

[25] H. Lokeshwor, L. K. Das, S. Goel, Robust method for automated segmentation of frames with/without distress from road surface video clips, J. Transp. Eng. 140 (1) (2013) 31-41.

[26] Y. ZHANG and H. ZHOU, "Automatic pavement cracks detection and classification using radon transform, J. Infor. Comput. Sci. 9 (17) (2012) 5241-5247.

[27] Y. J. Tsai, V. Kaul, A. Yezzi, Automating the crack map detection process for machine operated crack sealer, Autom. Constr. 31 (1) (2013) 10-18.

[28] S. Varadharajan, S. Jose, K. Sharma, L. Wander, C. Mertz, Vision for road inspection. In IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, USA, 2014, pp. 115 - 122.

[29] W. Xu, Z. Tang, J. Zhou, J. Ding, Pavement crack detection based on saliency and statistical features. In IEEE International Conference on Image Processing, Melbourne, Australia, 2013, pp. 4093-4097.

[30] H. Zakeri, F. M. Nejad, A. Fahimifar, Rahbin: A quadcopter unmanned aerial vehicle based on a systematic image processing approach toward an automated asphalt pavement inspection, Autom. Constr. 72 (2) (2016) 211-235.

[31] S. Hongxun, W. Weixing, W. Fengping, W. Linchun, W. Zhiwei, Pavement crack detection by ridge detection on fractional calculus and dual-thresholds, Inter. J. Multimedia Ubiquitous Eng. 10 (4) (2015) 19-30.

[32] C. A. Lettsome, Y.-C. J. Tsai, V. Kaul, Enhanced adaptive filter-bank-based automated pavement crack detection and segmentation system, J. Electronic Imaging 21 (4) (2012) 043008.

[33] F. M. Nejad and H. Zakeri, An optimum feature extraction method based on Wavelet–Radon Transform and Dynamic Neural Network for pavement distress classification, Expert Syst. Appl. 38 (8) (2011) 9442-9460.

[34] H. Ceylan, M. B. Bayrak, K. Gopalakrishnan, Neural networks applications in pavement engineering: A recent survey, Int. J. Pavement Eng. 7 (6) (2014) 434-444.

[35] N.-D. Hoang, Q.-L. Nguyen, D. Tien Bui, Image processing–based classification of asphalt pavement cracks using support vector machine optimized by artificial bee colony, J. Comput. Civ. Eng. 32 (5) (2018) 04018037.

[36] T. Wang, K. Gopalakrishnan, O. Smadi, A. K. Somani, Automated shape-based pavement crack detection approach, Transp. 33 (3) (2018) 598-608.

[37] W. R. L. d. Silva and D. S. d. Lucena, Concrete Cracks Detection Based on Deep Learning Image Classification. In Multidisciplinary Digital Publishing Institute Proceedings, 18th International Conference on Experimental Mechanics (ICEM18), Brussels, Belgium, 2018.

[38] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata, Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone, Comput. Aided Civ. Infras. Eng. 33 (12) (2018) 1127-1141.

[39] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks, Comput. Aided Civ. Infras. Eng. 32 (5) (2017) 361-378.

[40] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: A Deep Hierarchical Feature Learning Architecture for Crack Segmentation, Neurocomput. 338 (1) (2019) 139-153. https://doi.org/10.1016/j.neucom.2019.01.036

[41] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, A. Agrawal, Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection, Constr. Build. Mater. 157 (2017) 322-330.

[42] C. V. Dung, Autonomous concrete crack detection using deep fully convolutional neural network, Automation Constr. 99 (2019) 52-58.

[43] S. Albelwi and A. Mahmood, A framework for designing the architectures of deep convolutional neural networks, Entropy 19 (6) (2017) 242.

[44] Z. Tong, J. Gao, Z. Han, Z. Wang, Recognition of asphalt pavement crack length using deep convolutional neural networks, Road Mater. Pavement Des. 19 (6) (2018) 1334-1349.

[45] A. Bhandare, M. Bhide, P. Gokhale, R. Chandavarkar, Applications of Convolutional Neural Networks, Inter. J. Computer Sci. Infor. Technol. 7 (5) (2016) 2206-2215.

[46] C. Kyriakou, S. E. Christodoulou, L. Dimitriou, Detecting and Classifying Roadway Pavement Cracks, Rutting, Raveling, Patching, and Potholes Utilizing Smartphones, In Transportation Research Board 97th Annual Meeting, Washington DC, USA, 2018.

[47] S. Gao, Z. Jie, Z. Pan, F. Qin, R. Li, Automatic Recognition of Pavement Crack via Convolutional Neural Network, In Transactions on Edutainment XIV, ed. By Z. Pan, A. D. Cheok, W. Müller, Springer, Berlin, 2018, p. 82-89.

[48] B. Li, K. C. Wang, A. Zhang, E. Yang, G. Wang, Automatic classification of pavement crack using deep convolutional neural network, Inter. J. Pavement Eng. 21 (4) (2018) 1-7, https://doi.org/10.1080/10298436.2018.1485917.

[49] M. A. Nielsen, Neural networks and deep learning. Determination press, USA, 2015.

[50] S. Dorafshan, R. J. Thomas, M. Maguire, Comparison of deep convolutional neural networks and edge detectors for

image-based crack detection in concrete, Constr. Build. Mater. 186 (2018) 1031–1045.

[51] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, In Proceedings-International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain, 2011.

[52] S. J. Pan and Q. Yang, A survey on transfer learning, IEEE Transactions Knowledge Data Eng. 22 (10) (2010) 1345-1359.

[53] O. Russakovsky et al., Imagenet large scale visual recognition challenge, Inter. J. Comput. Vision 115 (3) (2015) 211-252.

[54] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, In Advances in neural information processing systems, Harrah's Lake Tahoe, NV, USA, 2012.

[55] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size, Computer Vision and Pattern Recognition, Cornell University, USA, 2016.

[56] C. Szegedy et al., Going deeper with convolutions, IEEE conference on computer vision and pattern recognition, Boston, USA, 2015, pp. 1-9.

[57] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, IEEE international conference on computer vision, Santiago, Chile, 2015, pp. 1026-1034.

[58] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, IEEE conference on computer vision and pattern recognition, Las Vegas, USA, 2016, pp. 770-778.

[59] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA, 2017.

[60] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in IEEE conference on computer vision and pattern recognition, Las Vegas, USA, 2016, pp. 2818-2826.

[61] P. S. Addison, The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance. CRC press, 2017.

[62] P. Prasad and G. Umamadhuri, Biorthogonal Wavelet-based Image Compression, in Artificial Intelligence and Evolutionary Computations in Engineering Systems, ed. By S. Dash, P.Chandra, B.Naidu, R.Bayindir, S.Das, Springer, Singapore, 2018, pp. 391-404.

[63] P. Luo, X. Qu, X. Qing, J. Gu, CT Image Denoising Using Double Density Dual Tree Complex Wavelet with Modified Thresholding, 2nd International Conference on Data Science and Business Analytics (ICDSBA), Changsha, China, 2018, pp. 287-290: IEEE.

[64] X. Wang and X. Feng, Pavement distress detection and classification with automated image processing, 2011 International Conference on Transportation, Mechanical, Electrical Engineering (TMEE), Changchun, China, 2011, pp. 1345-1350: IEEE.

[65] B. Sun and Y. Qiu, Automatic Pavement Surface Cracking Recognition Using Wavelet Transforms Technology, Second International Conference on Transportation Engineering, Chengdu, China, 2009, pp. 2201-2206.

[66] C. Ma, W. Wang, C. Zhao, F. Di, Z. Zhu, Pavement cracks detection based on FDWT, International Conference on Computational Intelligence and Software Engineering (CiSE), Wuhan, China, 2009, pp. 1-4: IEEE.

[67] J. Zhou, P. S. Huang, F.-P. Chiang, Wavelet-based pavement distress detection and evaluation, Optical Eng. 45 (2) (2006) 027007.

[68] F. M. Nejad, N. Karimi, H. Zakeri, Automatic image acquisition with knowledge-based approach for multi-directional determination of skid resistance of pavements, Autom. Constr. 71 (2) (2016) 414-429.

[69] G. Yang, Q. J. Li, Y. J. Zhan, K. C. Wang, C. Wang, Wavelet based macrotexture analysis for pavement friction prediction, KSCE J. Civ. Eng. 22 (1) (2018) 117-124.

[70] R. Abbasnia and A. Farsaei, Corrosion detection of reinforced concrete beams with wavelet analysis, Inter. J. Civ. Eng., Transaction A: Civ. Eng. 11 (3) (2013) 160-169.

[71] A. Dixit and S. Majumdar, Comparative analysis of coiflet and daubechies wavelets using global threshold for image denoising, Inter. J. Adv. Eng. Technol. 6 (5) (2013) 2247-2252.

[72] D. Wei and A. C. Bovik, Generalized coiflets with nonzero-centered vanishing moments, IEEE Transactions on Circuits Systems II: Analog Digital Signal Process. 45 (8) (1998) 988-1001.

[73] D. Wei and H. Cheng, Representations of stochastic processes using coiflet-type wavelets, in Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing, Pocono Manor, USA, 2000, pp. 549-553.

[74] R. Nigam and S. K. Singh, Crack detection in a beam using wavelet transform and photographic measurements, Struct. 25 (2020) 436-447.

[75] V. L. Fox, M. Milanova, S. Al-Ali, Scene Analysis Using Morphological Mathematics and Fuzzy Logic, in Computer Vision in Control Systems-1, ed. By M.N. Favorskaya, L.C. Jain, Springer, Switzerland , 2015, p. 239-259.

[76] P. Soille, Morphological image analysis: principles and applications, Springer Science & Business Media, Switzerland, 2013.

[77] R. C. Gonzalez and R. E. Woods, Digital image processing, 2nd edn. Pearson Education International, London, UK, 2007.