

## Research Article

# Real-Time Road Crack Mapping Using an Optimized Convolutional Neural Network

**M-Mahdi Naddaf-Sh** <sup>1</sup>, **SeyedSaeid Hosseini** <sup>2</sup>, **Jing Zhang** <sup>3</sup>, **Nicholas A. Brake** <sup>2</sup>,  
**and Hassan Zargarzadeh** <sup>1</sup>

<sup>1</sup>Phillip M. Drayer Electrical Engineering Department, Lamar University, Beaumont, TX, USA

<sup>2</sup>Department of Civil Engineering, Lamar University, Beaumont, TX, USA

<sup>3</sup>Department of Computer Science, Lamar University, Beaumont, TX, USA

Correspondence should be addressed to Hassan Zargarzadeh; [hzargarzadeh@lamar.edu](mailto:hzargarzadeh@lamar.edu)

Received 22 February 2019; Revised 1 July 2019; Accepted 12 August 2019; Published 29 September 2019

Academic Editor: Alex Alexandridis

Copyright © 2019 M-Mahdi Naddaf-Sh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pavement surveying and distress mapping is completed by roadway authorities to quantify the topical and structural damage levels for strategic preventative or rehabilitative action. The failure to time the preventative or rehabilitative action and control distress propagation can lead to severe structural and financial loss of the asset requiring complete reconstruction. Continuous and computer-aided surveying measures not only can eliminate human error when analyzing, identifying, defining, and mapping pavement surface distresses, but also can provide a database of road damage patterns and their locations. The database can be used for timely road repairs to gain the maximum durability of the asphalt and the minimum cost of maintenance. This paper introduces an autonomous surveying scheme to collect, analyze, and map the image-based distress data in real time. A descriptive approach is considered for identifying cracks from collected images using a convolutional neural network (CNN) that classifies several types of cracks. Typically, CNN-based schemes require a relatively large processing power to detect desired objects in images in real time. However, the portability objective of this work requires to utilize low-weight processing units. To that end, the CNN training was optimized by the Bayesian optimization algorithm (BOA) to achieve the maximum accuracy and minimum processing time with minimum neural network layers. First, a database consisting of a diverse population of crack distress types such as longitudinal, transverse, and alligator cracks, photographed at multiple angles, was prepared. Then, the database was used to train a CNN whose hyperparameters were optimized using BOA. Finally, a heuristic algorithm is introduced to process the CNN's output and produce the crack map. The performance of the classifier and mapping algorithm is examined against still images and videos captured by a drone from cracked pavement. In both instances, the proposed CNN was able to classify the cracks with 97% accuracy. The mapping algorithm is able to map a diverse population of surface cracks patterns in real time at the speed of 11.1 km per hour.

## 1. Introduction

Societal economic vitality and growth is intimately tied to the state of infrastructure and its ability to safely and efficiently handle the transfer of goods from one point to another. Pavement management, preservation, and rehabilitation strategies are critical components in maintaining the viability of infrastructure and economy over the long term. Roadway networks contain millions of miles of pavements, and maintenance operations of these systems

cost upwards of \$25 billion per year [1]. As part of the maintenance operations, pavement surveys, which include both surface and subsurface assessments, are required frequently to assess the state of the pavement and help to prioritize rehabilitative and preservative action. Moreover, according to the National Highway Traffic Safety Administration, 16% of traffic crashes are produced due to roadway environmental factors mainly by poor pavement conditions [2]. Poor road conditions also lead to excessive wear on vehicles and tend to increase the number of delays

and crashes which can lead to additional financial losses [3]. Currently, manual inspection is the most common technique for identifying pavement distress road surveys [4]. Manual inspection can be, however, time-consuming, costly, and labor-intensive. Furthermore, during manual inspection operations, human visual error is possible, the operation itself can be unsafe due to the passing of nearby motor vehicles, and the operations may impede traffic flow [5]. To overcome the limitations of manual inspection, automated and/or semiautomated crack detection techniques can be developed to measure, monitor, and map the evolution of the pavement surface and subsurface structure and distress profile [6]. Semiautomated modern pavement distress mapping or diagnosis techniques need to be nondestructive, cost-effective, accurate, enabling data acquisition at high-speed, and relatively user and environmentally friendly [7]. As part of an effort to lower costs and accelerate maintenance operations, transportation departments are prioritizing the development of automated systems profiling systems for pavement distress assessment [8]. There remains a need, however, to develop automated and real-time distress mapping and assessment tools that can provide the end-user with large quantities of information related to the distress type, geometry, and distress source without manual surveillance either in situ or by proxy. The prominent solution of replacing expert inspectors with robots that can automatically gather data and analyze them has been studied or suggested extensively in recent years.

In the literature, emphasis of automated crack detection works was set on both using image processing for data analysis and developing automatic method for fast data collection like using robots or vehicles. Advantages like portability, being nondestructive, and lane closure avoidance are some of the important aspects of using vehicles for data collection in pavement distress studies, as suggested in many publications [9, 10].

Despite all benefits of automated data collection methods, it leads to vast amount of raw pavement data. Interpreting the raw data needs human expert for analysis and decision making. Regarding the importance of on time maintenance of pavements, it is impossible to process  $256 \times 256$  all raw data relying on expert human performance which has led the researchers to develop automatic intelligent algorithms for processing gathered raw data. The utilization of computer vision methods for pavement engineering applications has grown exponentially over the last few decades [11], while many challenges should be addressed to achieve a full and seamless realization due to unwanted and highly variable image noises from random variation of brightness color, camera, and the environment [12]. In recent years, many transportation and highway agencies in the US have become interested in image processing-based methods for analyzing collected raw data from highways and roads [13].

Classical image processing algorithms were suggested for pavement distress analysis. Algorithms like edge detection [14], wavelet transforms [15–18], intensity thresholding, texture analysis, etc. have been well studied [18, 19].

Although numerous suggested classical methods have helped in pavement distress analysis, some drawbacks like being prone to environment noise, not being applicable under all road conditions, being dependent to certain image quality, etc. have reduced their robustness against processing varying data. In the recent literature, machine learning-based methods, especially deep learning, show promising results in pavement distress analysis [20–24]. Unlike classical methods, machine learning-based algorithms proved to be more robust for processing different pavement distress images under noisy conditions. Several machine learning methods like neural classifiers [25] and support vector machines (SVM) [26] had been suggested for pavement distress analysis. In a comprehensive review of computer vision-based defect detection and condition assessment of asphalt pavement, Koch et al. [11] identified SVM as the most robust machine learning technique for image-based pavement distress detection in 2015. Moreover, recently, deep learning has become a popular alternative in pavement distress analysis due to its convincing performance over SVM and other methods [27].

Cha et al. [10] created a database with 40,000 images of size pixels and annotated them into crack or intact bins utilizing MatConvNet [28] for crack detection that could achieve 98% accuracy. Gopalakrishnan et al. [29] studied transfer learning on a single-layer pretrained neural network classifier for pavement distress detection. They labeled their data as crack or (“1”) and no-crack or (“0”). Also, they could gain 90% accuracy by using ImageNet’s pretrained VGG-16 as the feature extractor. Dorafshan et al. [30] prepared a database of 1,574 crack and 16,426 without crack images. They compared deep learning and edge detection methods and suggested a combination of both for improving the results in crack detection of concrete. Also, they used AlexNet architecture for feature extraction. Smartphone-based data collection is proposed in Maeda et al. [31], and they have tested several object detection systems like Faster R-CNN, YOLO, SSD, and R-FCN. Gopalakrishnan, in [27], has extensively reviewed the most recent deep learning-based methods for pavement distress detection. Besides developing crack distress detection algorithm whether it is based on the classical or deep learning method, some papers have suggested specific platforms for data collection. Due to the vastness of the roadway system, automatic pavement screening is needed. Prasanna et al. [9] has suggested an automated crack detection algorithm, called spatially tuned robust multi-featured, for monitoring concrete bridges and they implement the algorithm on the robot platform. In [32], some had studied an automatic image-based road crack detection method and a vehicle-based data collection platform is used to collect data from different locations for further processing. Among several data collection methods, vehicle mounted cameras are the most popular one.

The integration of computer vision methods using deep convolutional neural networks (CNNs) shows exceptional promise for use in crack detection applications, but require many images for the training process [10, 27, 33]. Although

utilizing CNNs improves the crack detection accuracy, there are known drawbacks in the conventional approaches for studying the cracks [10]. In the available works [10, 27], whether asphalt pavement or other surfaces, the objective is to distinguish cracked areas of pavement from uncracked ones that yields to a binary decision with two outcomes: cracked or noncracked. Due to the proven outstanding performance of deep learning in contrast with other machine learning methods like SVM, Adaboost, and random forest, still some shortage exists. For instance, in [11], although it is proved that a CNN has the capability of detecting cracks with high accuracy, authors suggest an optimization for the CNN for future improvements. Also, using transfer learning for pavement distress detection, Gopalakrishnan et al. [29] suggested to add a feature for evaluating severity of detected cracks, which shows possible further improvements for pavement distress detection. In deep learning-based methods, for pavement distress detection, the current focus is on improving the accuracy of neural network for identifying cracks. Also, most of the recent work in this application uses AlexNet, VGG-16 CNN architecture, and some transfer learning methods for pavement distress detection task. It is important to consider that most of the mentioned architectures are designed and tested on datasets that do not include pavement distress data. Although in many cases transfer learning is applicable for reducing training time and improving accuracy, the objects in datasets like MNIST, ImageNet, CIFAR-100, etc. do not share similar patterns in pavement applications, so using transfer learning with similar CNN architectures is limited.

## 2. Methodology

As reviewed above, deep learning-based methods for pavement distress detection improves false detection accuracy within noncracked pavements, whether using different CNN architectures or transfer learning or pretrained models. This paper proposes an approach to geometrically map a surface crack on asphalt pavement using a technique that involves image partitioning and crack geometrical and spatial classification. This technique allows the user to both detect the presence and map a crack on the road surface in real time using raw input images. The work extends the functionality of CNN-based classification techniques, which up to date are limited to only crack presence detection and do not provide simultaneous geometrical mapping of the object [34, 35]. Crack images are aggregated in the database and indexed according to their orientation and spatial position within a squared partitioned area of the larger raw image file, which then allows the position and orientation to be estimated heuristically using thirteen unique categories. By applying this approach, instead of predicting crack position in each frame by a marginal error that depends on the searching window, we not only are able to detect and classify the cracks, but also map the crack and avoid errors caused by the searching window.

Instead of relying on predesigned CNN architectures, we have proposed an optimized architecture for the CNN

and hyperparameters within the pavement distress detection task. Also, rather than taking the approach in the crack detection task that is focused on detecting crack from noncrack, which introduces some false positive after classification, in this work, we propose a heuristic algorithm CMA which is able to regenerate crack shape automatically. Alternatively, in this work, we propose a method for mapping a crack's shape or analyzing the results based on the pattern of a crack in an entire image or video frame that could yield a smoother crack map by eliminating smaller cracks or errors. Moreover, the proposed method is a general concept for automating the road surface crack analysis that can be later adapted with several highway agencies protocols like American Association of State Highway and Transportation Officials (AASHTO) (PP67-10 and PP44-00) or Mechanical-Empirical Pavement Design Guide (MEPDG) [36]. In other words, by changing the camera parameters like sensor size, focal length, lens type, distance of camera from pavement, etc., it is possible to achieve the minimum deficiency length that is considered crack in several protocols.

In the proposed scheme, each image is first partitioned into 300 equal square tiles. Then, a CNN is developed and trained that classifies the cracks in the tiles to predefined categories. Since the categorization of the cracks is conducted tile-by-tile, the resulted map may show discontinuities at the borders of the tiles. To mitigate such discontinuity errors, a heuristic real-time crack mapping algorithm (CMA) is introduced. The CMA processes the classification results and, based on the regional crack's information, it modifies the current segment that yields to a unified and continuous map of cracks on the road surface image. Further, the CMA has the ability to eliminate small cracks or false positive objects isolated in one partitioned tile that is continuous over multiple partitioned tiles. Since the objective of this paper is to map the crack real-in-time in image frames from a streaming video, the CNN hyperparameters (HPs) had to be optimized so the processing time and the classification error for the input images are minimized simultaneously. To that end, a Bayesian optimization algorithm (BOA) was utilized in lieu of trial and error methods. Experimental results show that the CNN installed on a portable computer can process 5 frames per seconds providing the ability to map one band of a road real-in-time at the speed of 11.1 km per hour.

The primary objective of this work is to use real-time images to map cracks on the surface of an asphalt pavement. A crack is defined as a mechanical or thermal strain-induced separation of material. This material separation allows moisture to infiltrate the pavement structure internally, leading to premature failure or accelerated deterioration. Cracks are classified by their geometric orientation, source, width, and concentration per unit length or area. Only cracks visible and distinguishable to the naked eye are considered in the distress survey.

As illustrated in Figure 1, the proposed crack mapping scheme is comprised of three stages: database preparation, training and optimizing, and real-time crack mapping. To

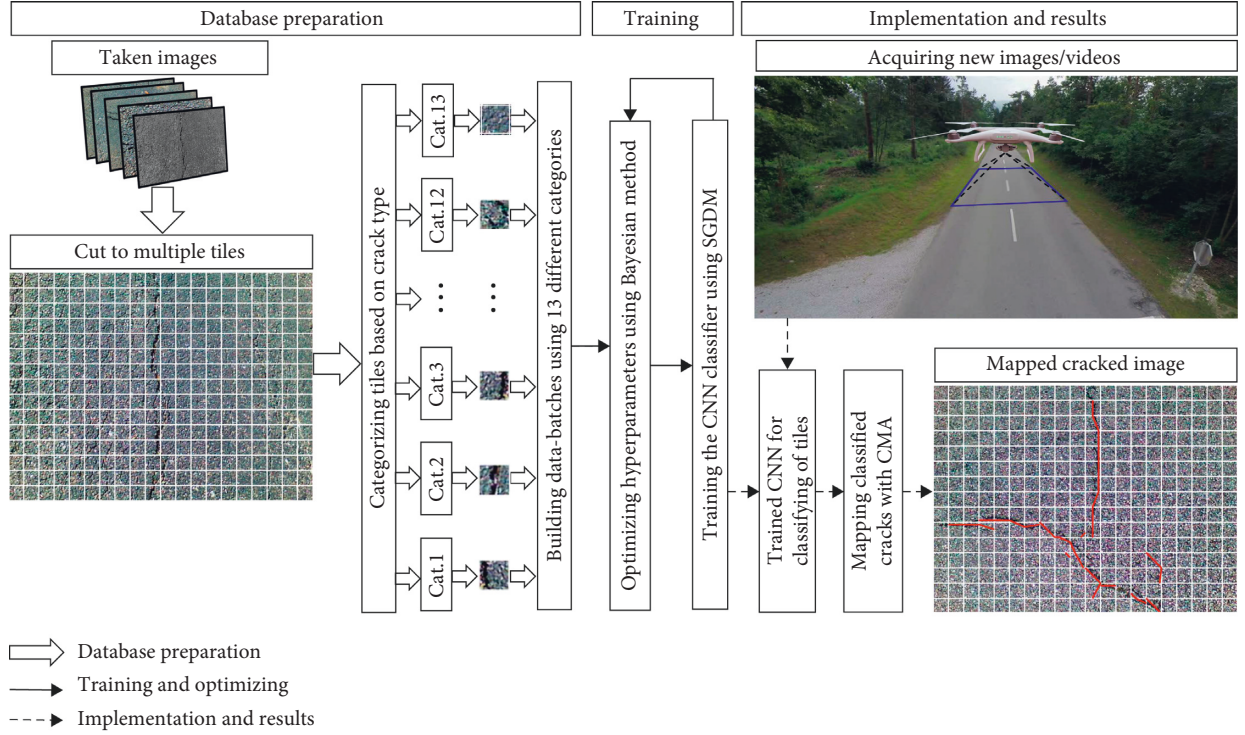


FIGURE 1: Flow chart for crack detection and mapping.

prepare the database, a descriptive approach was taken to categorize a given crack based on its relative position and geometric orientation within the image. This approach labels the cracks based on their geometric orientation. Multiple images of cracked asphalt pavements were gathered from the field, and each image was subdivided into smaller tiles (TIs). The TIs were classified into 12 different categories based on whether the TI contained a crack. If a crack is detected, it is further classified by crack position and orientation, i.e., horizontal, vertical, diagonal. TIs that did not contain a visible crack, but rather contained objects like grass, shadows, patched cracks, pavement markings, general uncracked pavement, etc., were binned into the 13<sup>th</sup> category. A CNN is then used to learn the unique features contained within each TI, and then each TI was classified  $\pm 10^\circ$  into a specific category (1–13) based on crack presence, position, and orientation. To further refine the process, BOA is used to objectively and systematically achieve optimal HPs by selecting optimal initial learning rates and momentum, which served to significantly reduced training time operations that have been shown to be tedious when tuned manually [10, 12]. The trained CNN is used for real-time crack mapping using video frames of a cracked roadway surface that were received from a camera installed on an aerial vehicle. The classified TIs are then sent to the CMA which maps the cracks in real time. The CMA is designed to enhance the decisions made by the CNN by eliminating cracks found only within one isolated TI and requiring the crack maps to be contiguous across multiple TIs to increase the smoothness and accuracy of the mapped crack field.

### 3. Experimental Study

**3.1. Database Preparation.** To develop and evaluate the CMA, 1500 images of cracked asphalt pavement surface were collected to prepare a database to train the CNN. Images were taken by a FLIR E5 camera with a  $55^\circ \times 43^\circ$  field of view and  $640 \times 480$  pixels resolution. The camera was placed between 1.5 m and 2.5 m above the road with a vertical line of sight and of tilt error. Each image was then divided into 300 equal TIs containing  $32 \times 32$  pixels. Each TI was then virtually divided into 9 equal blocks, as depicted in Figure 2(a), where the hashed blocks indicate the range of possible crack locations within a given bounded region; which is uniquely defined for each category. Figure 2(b) shows how an actual pavement surface image is aligned with a given category. Category groups {1, 2, 3}, {4, 5, 6}, and {7, ..., 12} represent horizontal, vertical, and diagonal cracks, respectively. The non-cracked category (13) is shown in Figure 2(c).

Based on the established definition of categories, a total of 6,695 TIs were handpicked from the 1500 images that fit into one of the 13 categories. For representing data, the CIFAR-10 [37] layout was used that yielded to 13 different data-batches equal to the number of crack categories. Each data-batch contains similar manually annotated TIs that represents one of the 13 categories along with an assigned label  $L_b \in \{1, 2, \dots, 13\}$  indicating TI's category number. Then, 90% of each data-batch was randomly taken along with their labels to form the training set, and the other 10% was used for the test set. A test set is then used to measure the accuracy of the trained CNN.

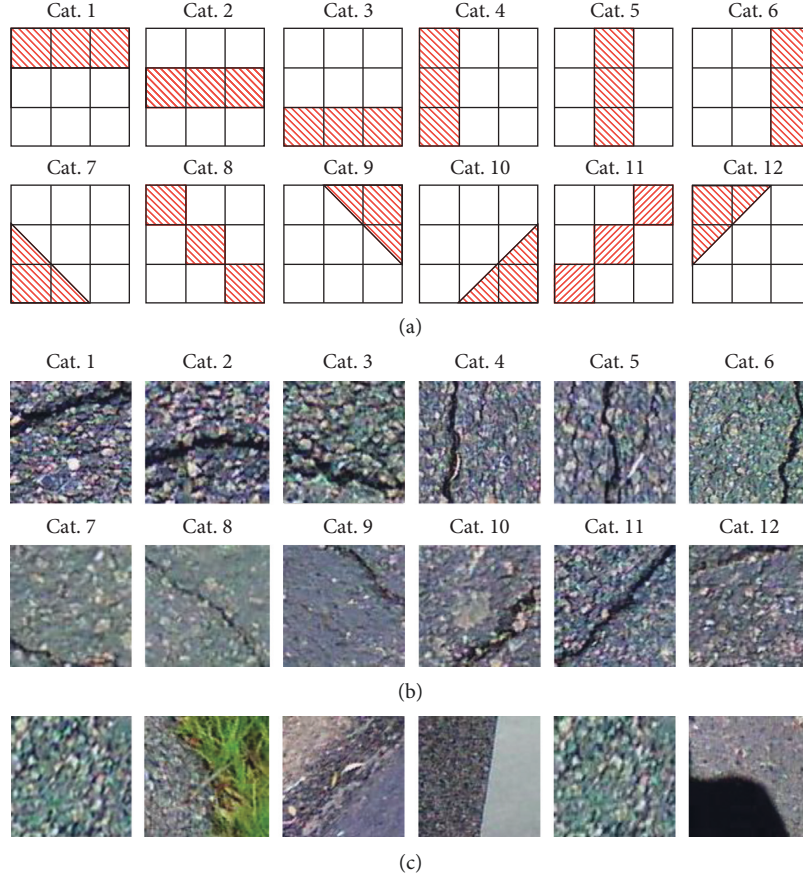


FIGURE 2: (a) Categorization method based on the position of a crack in each tile, indicating a defective section. (b) Sample of tiles in 12 categories. (c) Category 13<sup>th</sup> sample images of noncracked tiles.

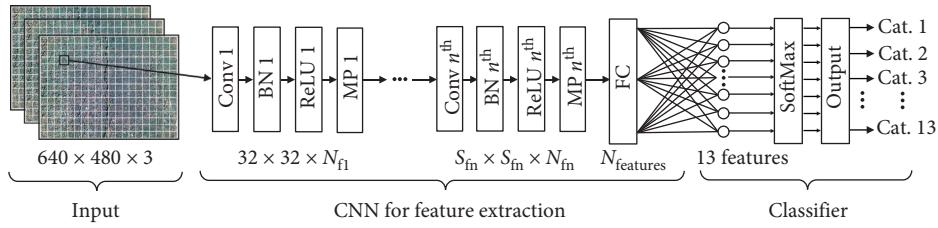


FIGURE 3: Network architecture used for crack detection. The number below each layer indicates the size of each layer. Conv: convolution layer, BN: batch normalization layer, MP: max-pooling layer, and FC: fully connected layer.

**3.2. CNN Architecture.** Figure 3 illustrates the overall CNN architecture-input and multiple convolutional layer, followed by a batch normalization layer, ReLU layer, max-pooling layer, fully connected (FC) layer, SoftMax layer, and output layer for the classification task.

The first layer is the input layer that receives the input image TIs for classification. The crack features in each data-batch were extracted using multiple convolution layers. This layer consists of various sets of neurons whose weights and biases will be updated relative to the crack features. In the convolutional layer, the neuron input consists of small sectors from the previous layer that is called the filter (kernel). The size of the filter,  $S_f$ , can be tuned from  $1 \times 1$  pixels up to the size of the input image. In the

convolution layer, the filter moves along the input and builds a convoluted feature map. To increase the number of feature maps, multiple filters should be used, and each filter has different weights and biases to be able to extract various features of the image. The stride (amount of horizontal and vertical movement of the filter on the input per convolution) is set to 2 pixels.

After the convolution layer, a batch normalization layer is used for reducing the CNN sensitivity to initial HPs values and decreasing training processing time. Following the batch normalization layer, a Rectified Linear Unit (ReLU) activation layer is added to apply a zero threshold to all negative values in the batch normalization layer; that means, the inputs from the previous layer  $b$  go through  $\max(0, b)$ . The

max-pooling layer downsamples the input by dividing it into rectangular pooling regions to compute the maximum of each region of gathered feature matrices. After designing the feature extractor, the fully connected (FC) layer is used to map the features matrix in the last layer in the form of a  $1 \times s$  vector, where  $s = 13$  is chosen equal to the number of categories in the database. For representing the probability distribution over multiple classes in the output of a classifier, a generalized model of binary logistic regression classifier (SoftMax function) is utilized after the FC [38, 39]. Considering the input of the SoftMax function as a sample tile that belongs to one of 13 categories,  $\text{TL} \in \text{Cat}_j$  where  $j \in \{1, \dots, 13\}$ ; then the category prior probability [40] is defined as  $P(\text{Cat}_j)$ , which shows the probability of  $\text{TL} \in \text{Cat}_j$  and conditional probability as  $P(\text{TL}, \bar{\phi} | \text{Cat}_j)$ , where  $\bar{\phi} = [\bar{w}, \bar{b}]$  is the parameter vector that consists of weights ( $\bar{w}$ ) and biases ( $\bar{b}$ ). The SoftMax function is described as

$$\begin{aligned} S_j(\text{TL}, \bar{\phi}) &= P(\text{Cat}_j | \text{TL}, \bar{\phi}) = \frac{P(\text{TL}, \bar{\phi} | \text{Cat}_j)P(\text{Cat}_j)}{\sum_{m=1}^j P(\text{TL}, \bar{\phi} | \text{Cat}_m)P(\text{Cat}_m)} \\ &= \frac{\exp(r_j(\text{TL}, \bar{\phi}))}{\sum_{m=1}^j \exp(r_m(\text{TL}, \bar{\phi}))}, \end{aligned} \quad (1)$$

where  $r_j(\text{TL}, \bar{\phi}) = \ln(P(\text{TL}, \bar{\phi} | \text{Cat}_j)P(\text{Cat}_j))$  and  $S_j$  is a probability distribution of the SoftMax function output, where  $0 \leq S_j \leq 1$  and  $\sum_{j=1}^{13} S_j(\text{TL}, \bar{\phi}) = 1$ .

Following the SoftMax function, the classification output layer (cross entropy function) is used to assign each input to one of the  $n = 13$  mutually exclusive categories using the loss function shown in the following:

$$l(\bar{\phi}) = - \sum_{i=1}^p \sum_{j=1}^n d_{ij} \ln S_j(\text{TL}_i, \bar{\phi}), \quad (2)$$

where  $p$  is the number of samples and  $d_{ij}$  is a matrix that shows with what probability the  $i^{\text{th}}$  sample of TL belongs to  $j^{\text{th}}$  category.

**3.3. Training.** Stochastic Gradient Descent with Momentum (SGDM) was used to train the CNN for classification. This method updates CNN's weights and biases to minimize the loss function that measures the difference between true classified and false classified Tls. The SGDM uses a subset of training data (mini-batch). The gradient derived from the data within the mini-batch is used for updating the weights and biases. Each update to the weights and biases is defined as one iteration. The gradient descent update law is described as

$$\bar{\phi}_{k+1} = \bar{\phi}_k - \lambda l(\bar{\phi}_k) + \eta(\bar{\phi}_k - \bar{\phi}_{k-1}), \quad (3)$$

where subscript  $k$  represents the iteration number, the initial learning rate is  $0 < \lambda < 1$ ,  $\bar{\phi}$  is a vector that contains the weights and biases,  $l(\bar{\phi})$  is the loss function and  $0 \leq \eta \leq 1$  is the momentum, which defines the level of contribution from

the previous step. For  $\lambda$  values close to 0, the learning process is slowed and values close to 1 lead to either diverging or suboptimal weights. Moreover, to prevent overfitting of CNN during the training process, L2 regularization [39, 41] is utilized as follows:

$$l(\bar{\phi}_{k+1}) = l(\bar{\phi}_k) + \frac{\tau(\bar{w}_k^T \bar{w}_k)}{2}, \quad (4)$$

where  $\tau$  is the regularization factor. To both prevent overfitting and feature memorization and improve the generalization of the SoftMax classifier during the training process, a modified data augmentation procedure is used during each iteration [38], where the Tls were translated randomly in the horizontal and vertical directions by a maximum of by  $\pm 4$  pixels. It is noteworthy that the Tls cannot get flipped or rotated since the classification process is dependent on crack orientation.

**3.4. Network Hyperparameters (HPs) Optimization.** HPs in the proposed CNN architecture and SGDM are the filter size  $S_f$ , number of filters  $N_f$ , the number of CNN layers ND,  $\eta$ ,  $\tau$ , and  $\lambda$ . The search range for HP was defined as  $\text{ND} \in \{1, 2, \dots, 15\}$ ,  $S_f \in \{1, 2, \dots, 12\}$ ,  $N_f \in \{1, 2, \dots, 100\}$ ,  $0 \leq \eta \leq 1$ ,  $0 \leq \tau \leq 1$ , and  $0 < \lambda < 1$ . The possible values for ND,  $S_f$ , and  $N_f$  are integers and for  $\eta$ ,  $\tau$ , and  $\lambda$  are logarithmically spaced values between 0 and 1. The classification error is the number of misclassified Tls by the classifier (SoftMax). The objective of optimization is to find optimal values for the HPs such that the classification error is minimized. So, the objective function can be considered a function with HPs as the input and the classification error as the output. Modeling of this objective function is algebraically complicated and computationally intensive. The BOA is capable of performing optimizing the HPs to minimize the classification error, while the objective function is considered as a black-box [42]. To perform the BOA, a validation set was defined that consists of 10% randomly selected Tls from the training set. The inputs of the objective function are training set and validation set. As shown in Figure 4, the objective function trains the CNN and returns the classification error on the validation set. By modeling the calculated error using Gaussian process (GP) as mentioned in [43] and in multiple iterations  $z$ , where  $z = \{1, 2, \dots, 100\}$ , the BOA finds the optimal values for HPs that minimize the classification error. The kernel function that was used for GP is the Automatic Relevance Determination (ARD) Matérn 5/2 in [44]. In addition, the acquisition function ( $q_z(\text{HP})$ ) that is used for the GP is the Expected Improvement function  $E(\cdot)$  [45], as follows:

$$q_z(\text{HP}) = \arg \max_{\text{HP}} E(\max\{0, f_{z+1}(\text{HP}) - f_z^{\max}(\text{HP})\}). \quad (5)$$

To perform the BOA, a validation set was defined that consists of 10% randomly selected Tls from the training set. The inputs of the objective function are training set and validation set. As shown in Figure 4, the objective function trains the CNN and returns the classification error on the validation set. By modeling the calculated error using

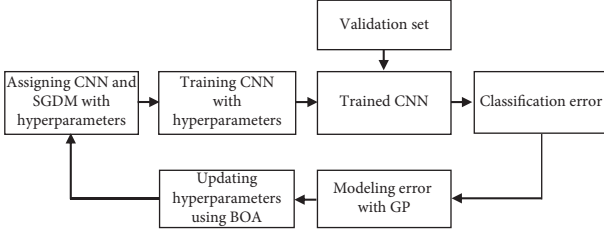


FIGURE 4: Block diagram of optimizing HPs using BOA.

Gaussian process (GP) as mentioned in [43] and in multiple iterations  $z$ , where  $z = \{1, 2, \dots, 100\}$ , the BOA finds the optimal values for HPs that minimize the classification error. The kernel function that was used for GP is the Automatic Relevance Determination (ARD) Matérn 5/2 in [44]. In addition, the acquisition function ( $q_z(\text{HP})$ ) that is used for the GP is the Expected Improvement function  $E(\cdot)$  [45], as follows:

$$q_z(\text{HP}) = \arg \max_{\text{HP}} E(\max\{0, f_{z+1}(\text{HP}) - f_z^{\max}(\text{HP})\}), \quad (6)$$

where  $f_z^{\max}(\text{HP})$  is the current maximum observed value for the objective function. The next estimation for maximizing the objective function is obtained by using the acquisition function. The GP posterior is updated in each iteration using equation (6):

$$P(f_z | u) = \frac{P(u | f_z)P(f_z)}{P(u)}, \quad (7)$$

where  $u = \{(\text{HP}_z, f_z), z = 1:100\}$ .

The extrema of  $f_z(\text{HP})$  was obtained numerically at sampled values of the function. A closed form expression of the objective function is not required within the BOA mathematical structure [46]. The objective function and acquisition function for two of SGDM HPs, i.e.,  $\eta$  and  $\lambda$ , during the optimization process are shown in Figure 5. As depicted in Figure 5(a), the observed points are demarcated by blue dots ( $f_z(\text{HP})$ ), the model mean that is obtained from the observations is depicted as the red surface, and subsequent evaluation point addition is demarcated with a black dot. Moreover, Figure 5(b) illustrates the acquisition function. The objective function is shown to reach a minimum at the 69<sup>th</sup> iteration; this point is demarcated with a black star. Figure 5(b) shows the maximum feasible value that is generated upon minimizing the classification error.

The total number of iterations was set to 100. Each iteration calculates the classification error among 600 randomly selected Tls from 13 data-batches. Figure 6 represents the estimated (expected) improvement in each iteration and the calculated improvement during optimization. The BOA was evaluated statistically using the Wald method [47] by representing the images in the test set as independent events with a known probability of success. The number of misclassified images was represented with a binomial distribution. By applying the trained CNN with optimized HPs on the test set and

computing the number of correctly classified Tls, the test error  $E_t$  is defined as follows:

$$E_t = 1 - \frac{1}{b} \sum_{i=1}^b \text{Ts}_i, \quad (8)$$

where Ts and  $b$  are the number of correctly classified Tls, and the total number of Tls in the test set, respectively. Note to evaluate the trained CNN performance on the test set without exposing the CNN to the optimization process,  $E_s$  is used to obtain the standard error. This approach helps to increase the optimization speed. The standard error is represented as follows:

$$E_s = \sqrt{\frac{E_t}{\text{Ts}}} (1 - E_t). \quad (9)$$

Moreover, as the target of this research, to obtain a  $\pm 3\%$  error margin, a confidence interval of 97% is defined to calculate the generalization error  $E_G$  defined as

$$E_G = E_t \pm 0.97 E_s. \quad (10)$$

The final HPs value for the CNN were  $\text{ND} = 4$ ,  $S_{f1} = 5$ ,  $S_{f2} = 5$ ,  $S_{f3} = 6$ ,  $S_{f4} = 2$ ,  $N_{f1} = 119$ ,  $N_{f2} = 119$ ,  $N_{f3} = 108$ , and  $N_{f4} = 96$ . Also, the optimized values for SGDM were  $\lambda = 0.0005145$ ,  $\eta = 0.040069$ , and  $\tau = 0.002859$ .

Applying the optimal HP values to the CNN and SGDM yields a CNN with 24 layers and 96.67% accuracy in 10 epochs as it is shown in Figure 7(a). Moreover, the minimized value for the loss function was 0.033 in 10 epochs, as shown in Figure 7(b). The generalized error  $E_G$  interval for the test set was  $[0.0476, 0.0184]$ . Figure 8 illustrates 24 randomly selected output of the fourth convolution layer, indicating that the features of Tls extracted by the CNN. Extracted features of the position and orientation of a crack in each Tl can be easily recognized in Figure 8. Figure 9 depicts the confusion matrix for the test set, which is obtained based on the CNN that is trained using final values. As shown in Figure 9, the highest confusion is between the two categories 5 and 6. The reason is that, from Figure 2(a), categories 5 and 6 both represent vertical cracks in the middle and right side of Tl, respectively, which intrinsically leads to a higher probability of misclassification. 24 randomly selected Tls from the test set that were selected for testing the optimized trained CNN are shown in Figure 10 along with the percentage probability that they belong to each category. The trained CNN in this section will be used to map the cracks with the CMA described in the following section.

#### 4. Real-Time Crack Mapping Algorithm (CMA)

In this section, the proposed real-time Crack Mapping Algorithm is discussed. So far, a CNN is trained that classifies the cracks in a Tl. However, if the CNN is directly used to map a crack, the resulted map will not be continuous. The discontinuity is the result of the classification errors, due to the size of the tile blocks and the limited number of categories that the CNN classifier recognizes. To mitigate this

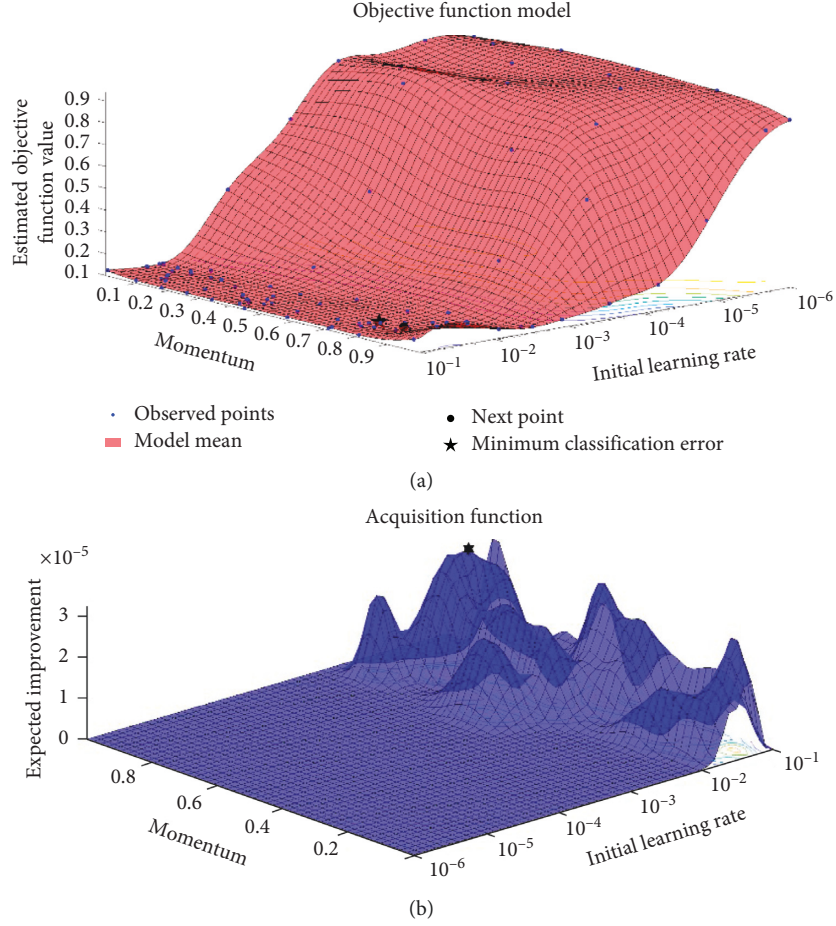


FIGURE 5: (a) The observation function model. (b) Acquisition function for two parameters of the SGDM. The starred points are the calculated optimal values at the 69<sup>th</sup> iteration for  $\lambda$  and  $\eta$ .

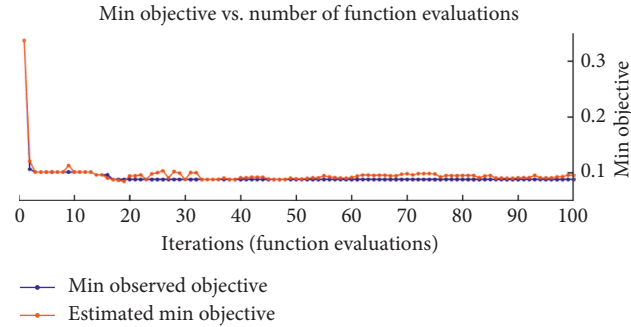


FIGURE 6: Evolution of minimum observed objective and estimated minimum objective in total 100 iterations, where the minimum calculated value is 0.07.

problem, as it is indicated in Figure 1, a CMA block is added to the mapping scheme that smoothens the resulted final map. As shown in Figure 3, an input image or video frame is divided to TIs in the same size as that of the database TIs assuming that input images have  $640 \times 480$  pixels. The classifier assigns a score to TIs related to their similarity to each category. The higher the score, the more probable that TI's crack belongs to a category. Among all assigned scores

for a TI, if a score is less than a defined threshold value of 85%, it is assumed as a noncracked TI that is the 13<sup>th</sup> category. As shown in Figure 11, the divided input image has 15 rows  $r = A, \dots, O$ , and 20 columns,  $c = 1, \dots, 20$ . As depicted in Figure 12, for each crack category, a raw mapping plot was defined. The raw mapping plot is a collection of straight-line segments that estimates crack position and orientation based on its classification. This

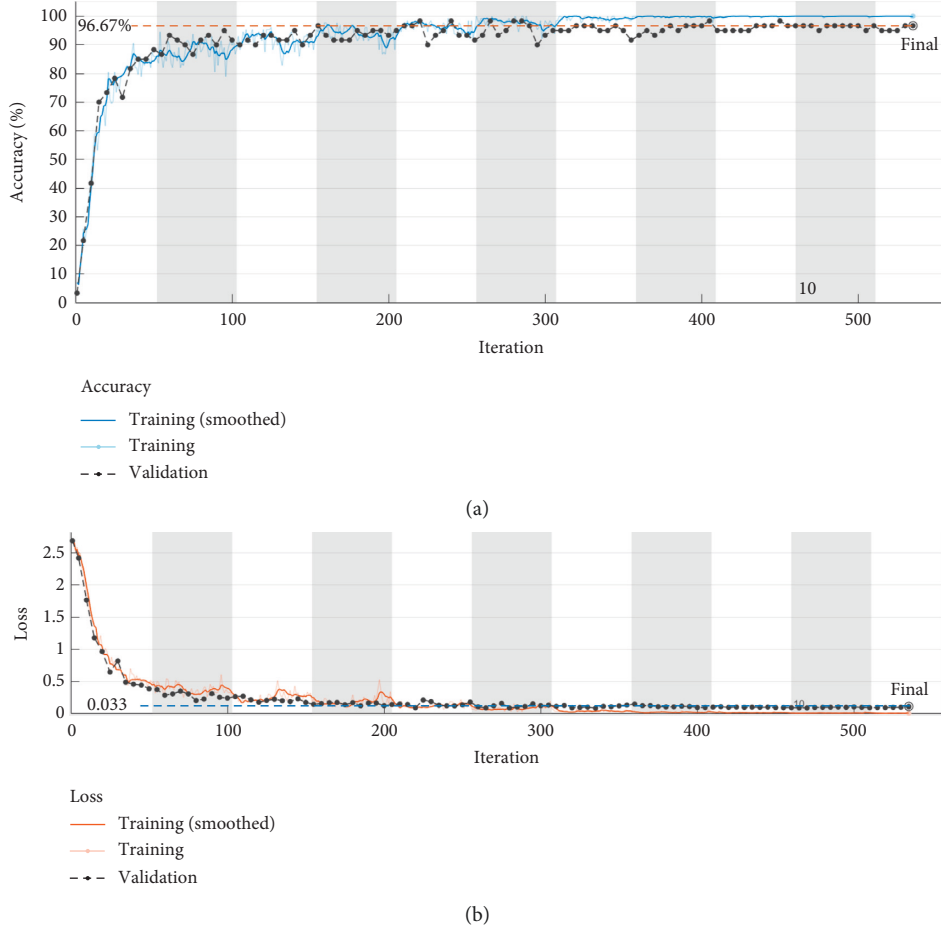


FIGURE 7: (a) Training CNN with optimized calculated value reached 97% accuracy after 10 epochs. (b) Loss function output during training, the error value is 0.033 after passing 10 epochs.

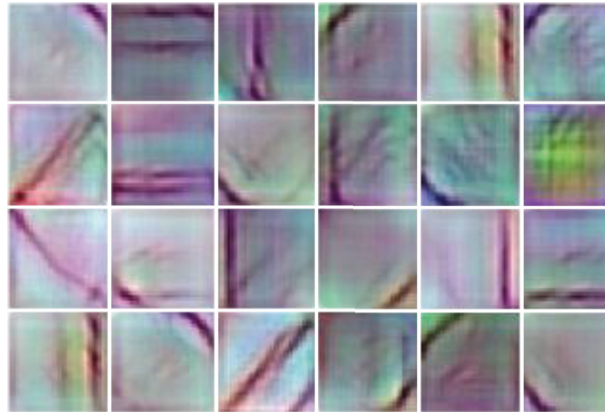


FIGURE 8: 24 randomly extracted features in the last convolution layer.

mapping will not be interconnected between tiles, which leads to two immediate deficiencies: isolated cracks and nonconnected cracks which lead to an overall mapping error.

To improve the crack connectivity between tiles, without refining the pixel dimensions of the TIs and increasing the

number of classification categories, a CMA block was created to “tie” the crack line segments between neighboring tiles. This procedure is described as follows: if a TI has any common side (Si) or corner with other, TIs it is called neighbor TI. Each TI has at least 3 neighbors at the image corner and at most 8 neighbors in the image interior. An

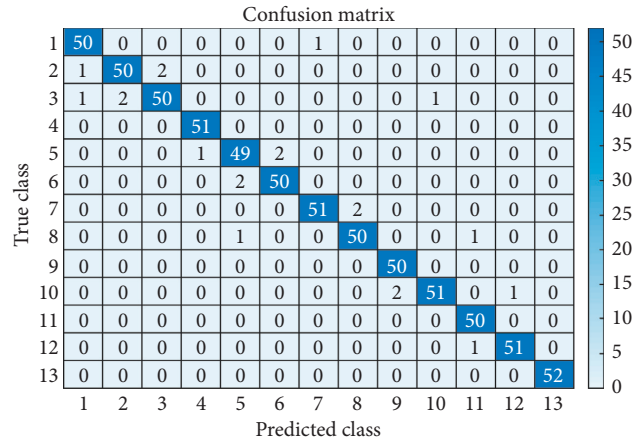


FIGURE 9: Confusion matrix between all 13 categories of the test set.

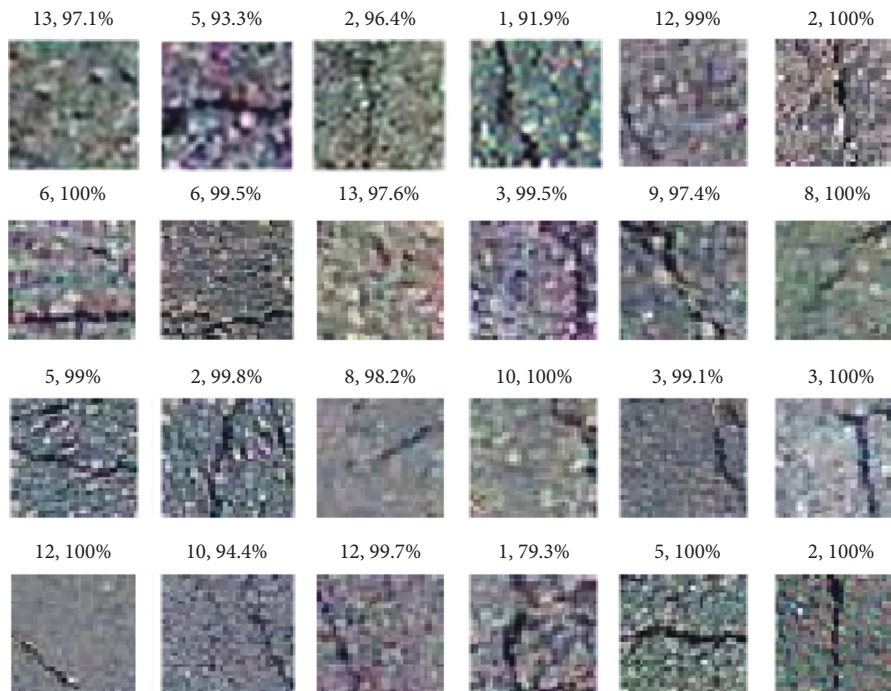


FIGURE 10: 24 randomly selected tiles from the test set. Category and its confidence are available on top of each tile.

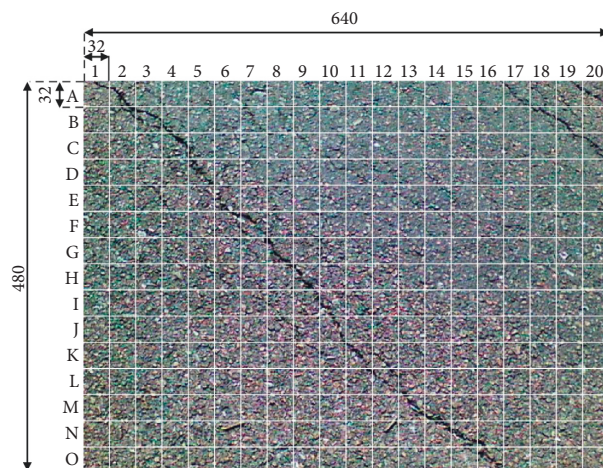


FIGURE 11: Montaging all 300 tiles.

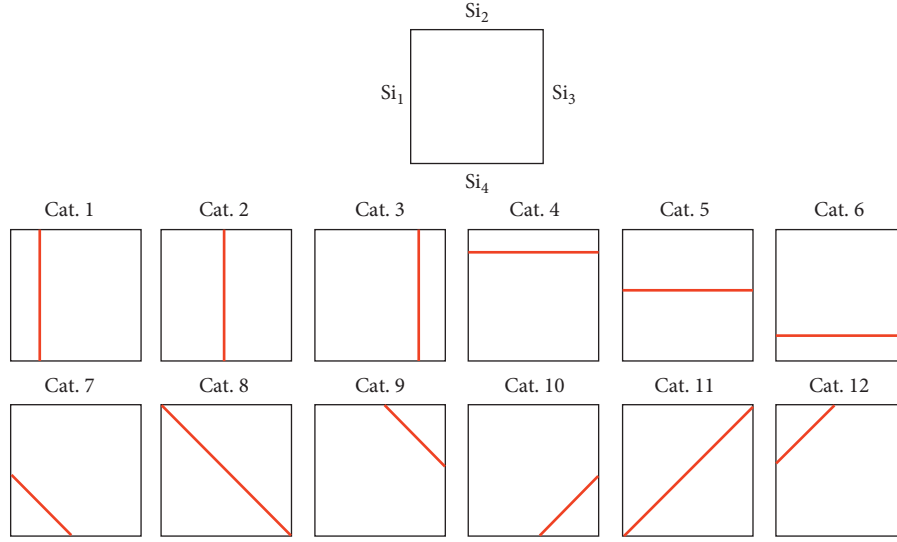


FIGURE 12: Raw plot based on the position of each crack in a single tile for all 12 categories that contain a crack.

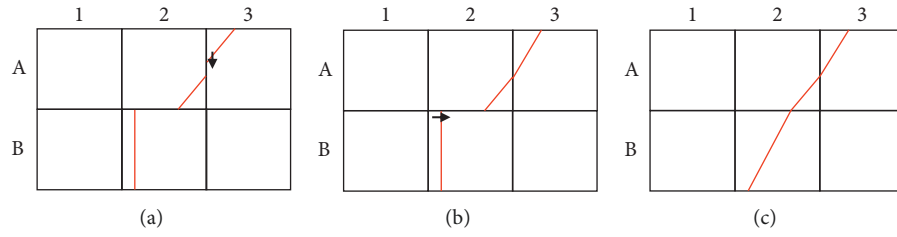


FIGURE 13: Applying the CMA, to make the crack mapping unify.

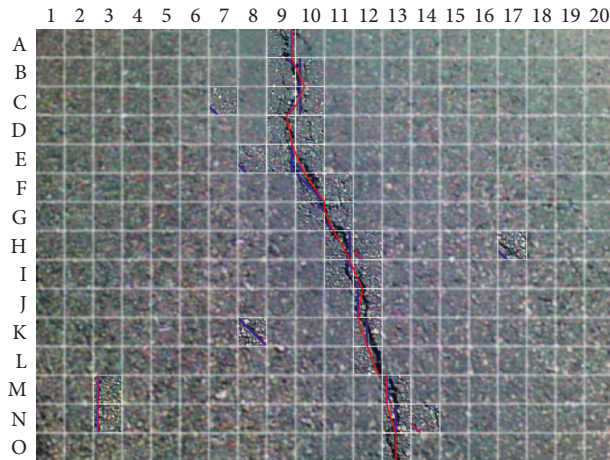


FIGURE 14: Blue lines show the raw mapping of the cracks. The crack map is not unified, and there are isolated mapped cracks available in the image. Red line shows the improved mapping the cracks using the CMA, isolated cracks were eliminated, and crack's map is unified and smoothed.

isolated Tl is a Tl with a detected crack, but no neighbor Tls have detected cracks. The CMA begins to map the input image in real time by scanning from row A from left to right. To map each row, the CMA processes right, left, and upper neighbors. While scanning a row, isolated Tls are discarded

as noncrack Tl. On the other hand, if two or more crack neighbor Tls exist, the raw linear segment plot will be translated vertically or horizontally until the raw mapping segments in the Tl and neighbor Tl intersect. As it is depicted in Figure 13(a), the cracks in Tls are mapped by a

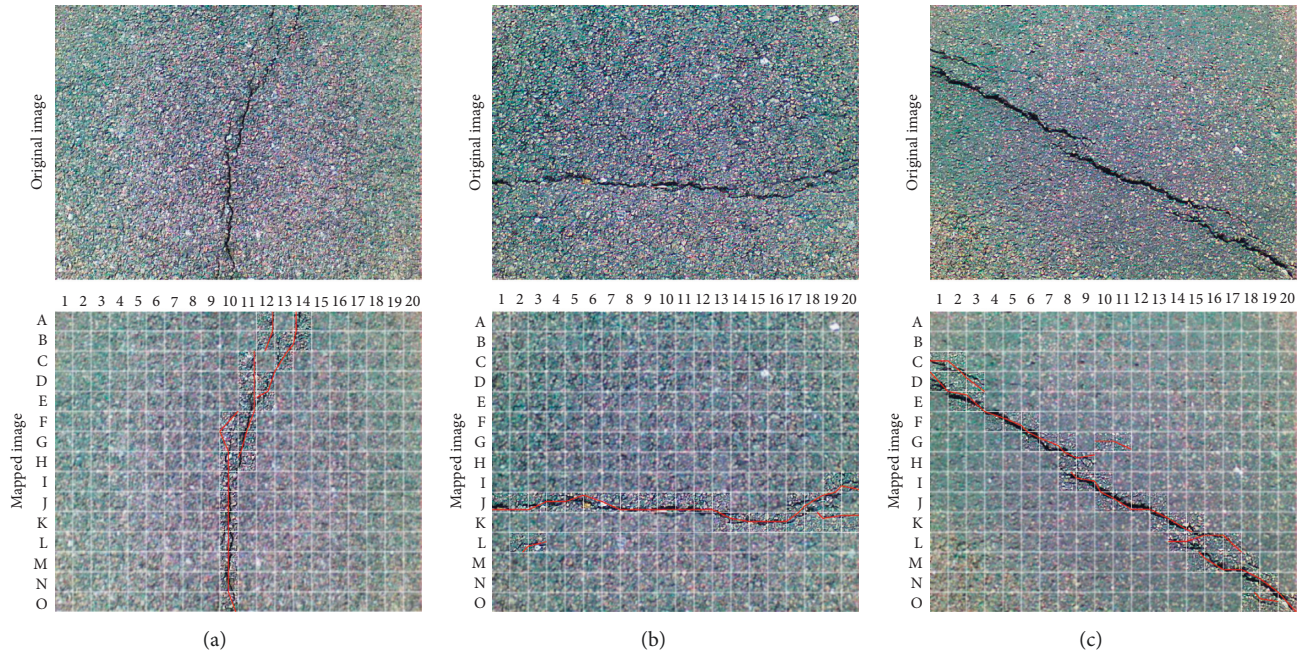


FIGURE 15: (a) Vertical, (b) horizontal, and (c) diagonal cracks used to test the crack mapping algorithm.

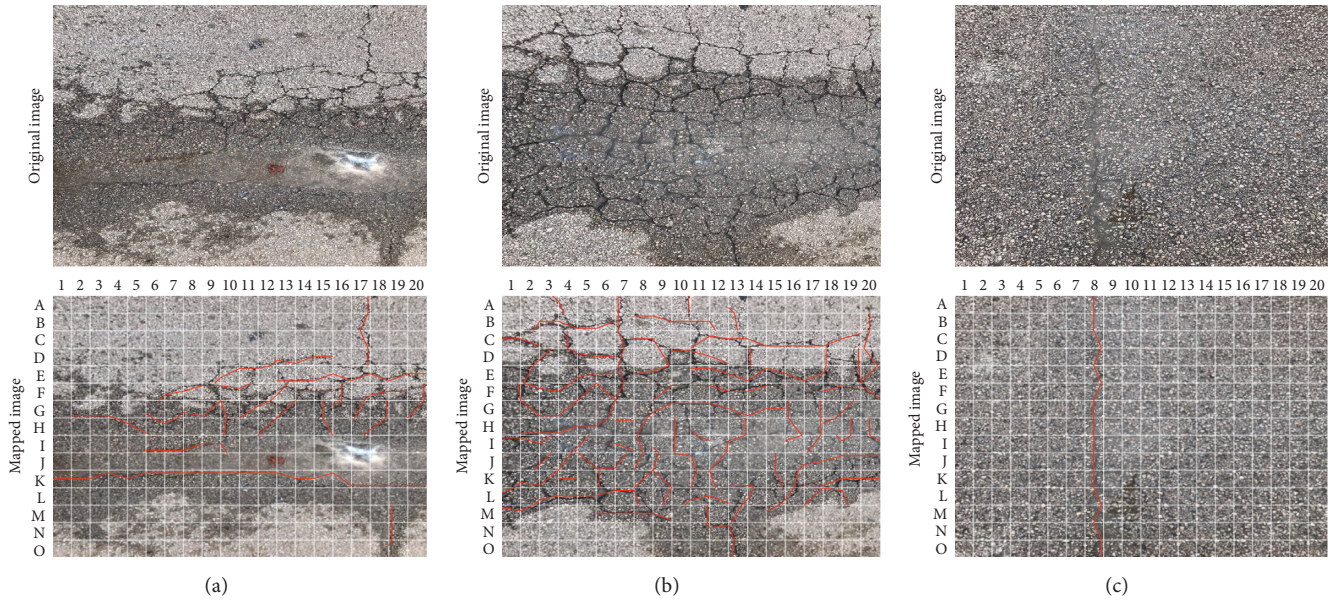


FIGURE 16: Continued.

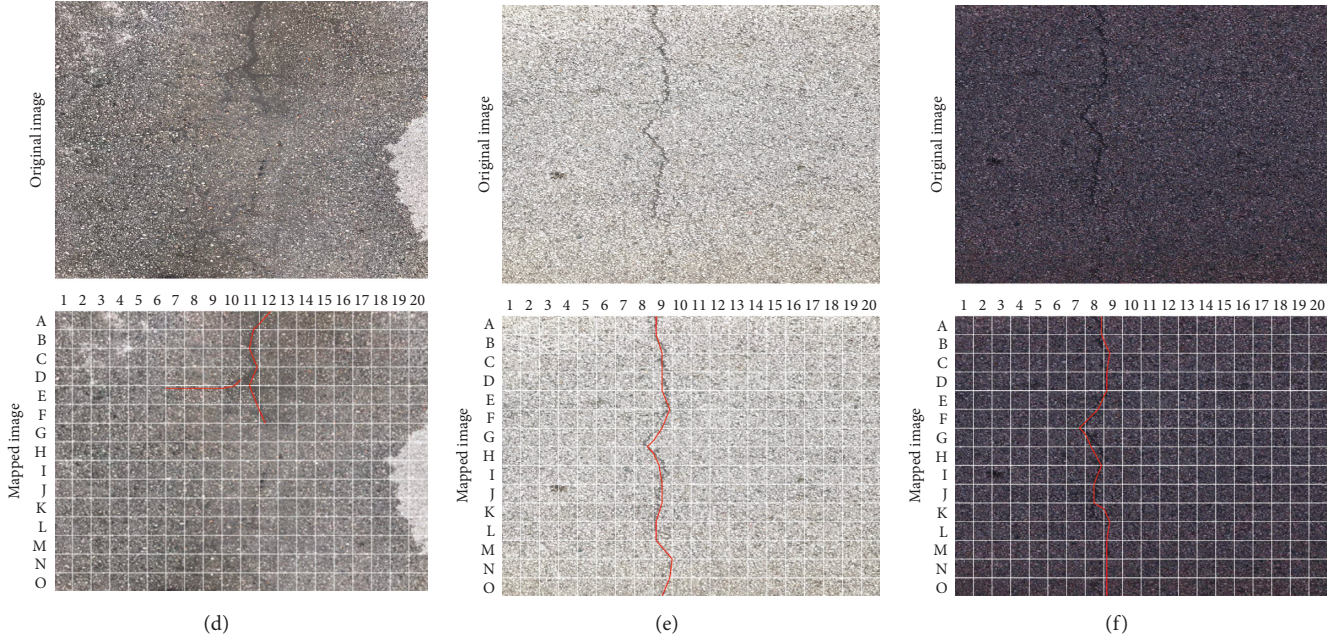


FIGURE 16: Performance of the algorithm on multiple cracks under different conditions (a), (b), (c), and (d) wet, (e) bright, and (f) dark.

linear segment and the CMA starts scanning from row A (Left to right). The tile A2 has a neighbor with a shared side on A2 and B2. The CMA translates the start of linear segment vertically downward until intersecting the end of the segment in A2, as shown in Figure 13(b). After scanning row A, the next row B is then scanned by the CMA. Tile B2 has a detected crack and detects a crack in the neighbor tile A2. As depicted in Figure 13(c), the CMA then translates the end of the line segment within B2 horizontally to the right until intersecting the start of the line segment in A2. This procedure continues until every row in the image has been processed.

Figure 14 shows the connected CMA modified mapping plots (red lines) overlaying both crack pavement image and raw mapping plots (blue lines: without segment connection). An isolated crack (blue line) is detected in tile C7 and classified into category 7. No neighbor tiles to C7 have a detected crack. More isolated cracks are detected in C7, E9, H17, and K8. The isolated cracks are eliminated within the CMA. The surviving linear segments are those that have neighbor TIs which are shown from A9, B9, B10, C9, D9 ... to O13. By defining the neighbor TI, the mapped plots from A9, B9, B10, C9, D9 ... to O13 form a pattern which can map the underneath crack. The proposed method is not limited to a certain vertical distance of the camera from the road. By having the camera's parameters like focal length, sensor size, etc., the pinhole camera model in [48] can be used to find the real dimensions of the road surface in each image. This information is required to determine the size of the crack in each image. As stated in Section 3, images were taken from distance of 1.5 to 2 meters above the cracked asphalt. For instance, for the camera used to collect the images in this work, a  $640 \times 480$  pixels image would cover a  $1036 \text{ mm} \times 915 \text{ mm}$  to  $1280 \text{ mm} \times 1097 \text{ mm}$  block on the

road. In that case, each TI would cover a  $51.8 \text{ mm} \times 61 \text{ mm}$  to  $64 \text{ mm} \times 73.1 \text{ mm}$  area.

Therefore, considering the isolated TIs that are eliminated by the CMA leads to exclusion of cracks with maximum length of 80.02 mm to 97.1 mm. On the other hand, crack width is one of the other requirements by different protocols for defining severity of deficiency. As an instance, AASHTO protocol (PP44-00 and PP67-10) has three different levels for measuring the damage severity. Level 1 is defined as cracks with width of less than 3 mm, level 2 refers to cracks with width between 3 mm and 6 mm, and level 3 is cracks with width of greater than 6 mm. Moreover, three major types of cracks in most protocols are longitudinal, transverse, and alligator cracks [36, 49]. Although converting length and width of cracks to the percentage of lane area is a straightforward task for transverse and longitudinal cracks, there is no certain way for applying the same method for alligator cracks. Hence, some protocols like HPMS, LTPP, and PP44-00 are focusing on transverse and longitudinal cracks and considering alligator cracks as a combination of those two types. With the mentioned equipment that is used in this paper, it is possible to detect cracks with minimum width of 2 mm and maximum width of 30 mm in images that are taken from 1.5 to 2 meters of the pavement surface. The proposed algorithm has no limitation over the length of crack. Moreover, the proposed algorithm is capable of detecting all three types of cracks that is defined widely accepted protocols.

## 5. Experimental Results

In this section, the test results of the trained CNN are presented in two subsections: (i) the performance of the algorithm was tested on several single images with various crack

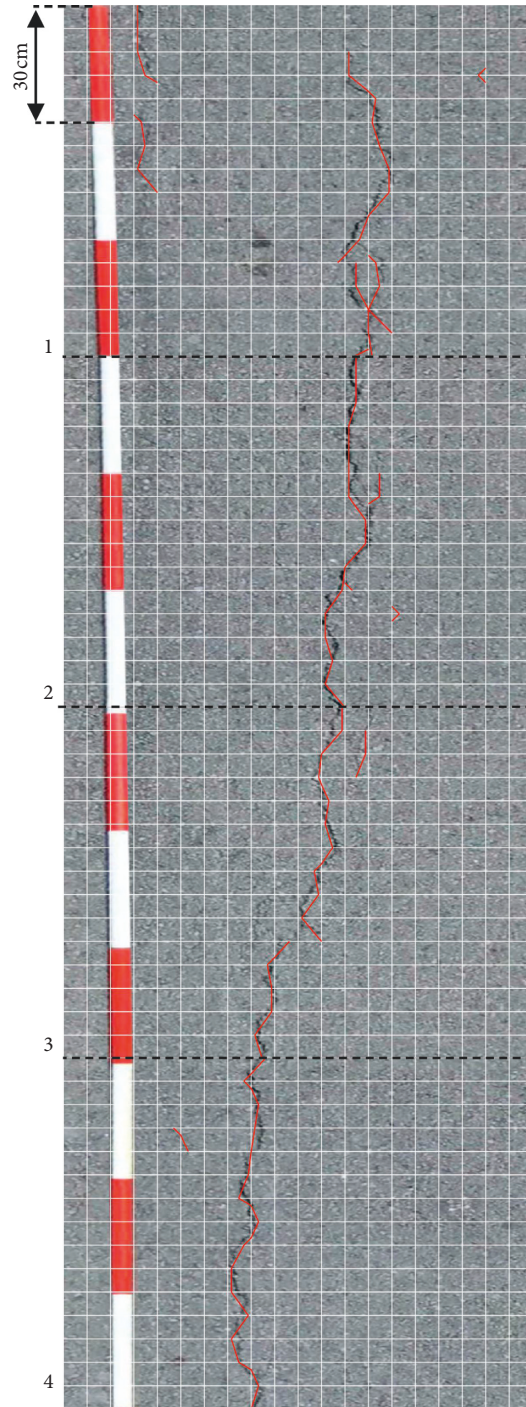


FIGURE 17: 4 processed frames with detected and mapped cracks. Mapping algorithm was able to separate all of 4 significant cracks and map them.

shapes and (ii) a real-time mapping evaluation was done on a captured video that is obtained from a randomly selected pavement roadway. The test images from the training and testing database were not used, not filtered or modified, and taken under varying light condition and camera position. Also, in this paper, MATLAB was used for training and optimizing the CNN and implementing the CMA algorithm.

*5.1. Single Image Mapping Evaluation.* As depicted in Figure 15, samples of three images having typical types of cracks with vertical (Figure 15(a)), horizontal (Figure 15(b)), and diagonal (Figure 15(c)) orientations were used to test the CMA. There are no isolated tile cracks and the connected mapped cracks cover the main crack with substantial accuracy.

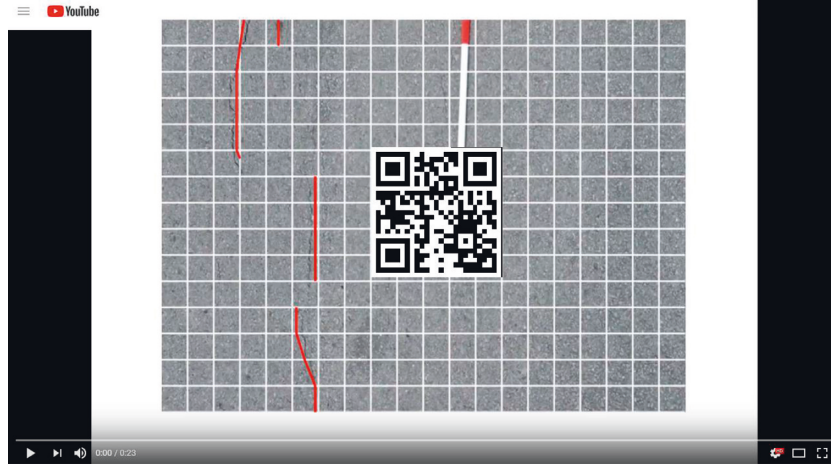


FIGURE 18: The real-time video of CMA is available at this link: <https://youtu.be/SGh03B5EEpk>.

TABLE 1: Retraining the CNN by changing the HPs to validate the calculated values by the BOA.

| Changing the HPs (%) | Accuracy (%) |
|----------------------|--------------|
| +5                   | 96           |
| -5                   | 95.9         |
| +10                  | 95.2         |
| -10                  | 95.7         |
| +20                  | 94.6         |
| -20                  | 93.1         |
| +30                  | 92.4         |
| -30                  | 89           |

Moreover, to test the algorithm's performance under different weather and illumination conditions, multiple cracks were selected and pictured in different situation. Figures 16(a)–16(f) show the wet, bright, and dark conditions respectively. The CMA could map cracks regardless of the road condition.

**5.2. Real-Time Mapping Evaluation.** Also, to evaluate the CMA performance in a real-time manner, a video was captured using a DJI Phantom 4 drone with mounted camera. The CMA was implemented on the captured video's image frames. The drone was set to an altitude of 2.5 m from the ground. The camera was perpendicular to the ground, and the video quality was set to resolution of  $640 \times 480$  and 120 fps. Since the drone was flown with a slow speed above the crack, and due to the high frame rate, the frame interval for processing was set to 60 frames. A ranging rod, 30.48 cm, with orange and white bands was set beside a road crack, and the drone was flown 3.6 meters in the forward direction. The laptop that was used for processing the captured videos was Alienware 15 R3 with NVIDIA GTX 1070 GPU and Intel Core i7 processor. As it is shown in Figure 17, the algorithm was able to detect and map deep cracks and avoid oil spills on the surface road. Referring to Figure 17, each frame was taken every 0.2 milliseconds. A total of 5 frames could be processed per second which covers 0.6 m of the road pavement. This indicates that the maximum speed of the

real-time mapping with the current hardware is 11.1 km/h. This speed, however, can be increased with the advent of lighter and more powerful graphical and processing units. Figure 18 shows a video obtained from the drone footage with real-time crack mapping segments produced from the CMA.

The proposed algorithm in this work improves upon existing work [10] by integrating crack detection with a crack mapping using image segmentation and classification within a CNN architecture. In addition, the optimized CNN architecture proposed here uses a significantly lower number of filters in the convolution layer (256) leading to reduced computational demand in both CNN training and real-time processing. As mentioned in Section 6, the BOA is used to compute the HPs. To verify the fact that the selected optimal values maximize the CNN accuracy, during the training process, all HPs were perturbed by a  $\pm 5\%$ ,  $\pm 10\%$ ,  $\pm 20\%$ , and  $\pm 30\%$  white noise. As shown in Table 1, perturbing the HPs by  $\pm 5\%$  decreases the accuracy by about 1%, while as it increases to  $\pm 30\%$ , the accuracy decreases by at most 8%.

## 6. Conclusion

In this paper, an algorithm for mapping road cracks in real time using convolutional neural networks was proposed and tested. Authors gathered the database for this work, and due to limited available resources, the size of the database was limited to 6695 images. The convolutional neural network in this work was optimized using the Bayesian optimization algorithm. A heuristic algorithm for real-time crack mapping was introduced and tested on different images with complicated crack position and orientation. Also, a video was recorded and processed for testing the real-time ability of the algorithm. Although the database was carefully selected and curated in this work, the authors attempted to include a robust population of crack images to improve the selection and classification power of the CNN. However, this study is limited to only one block size and 13 classification categories. Certainly, for commercial applications, increasing the number of images within the training and

increasing the computing power will allow users to reduce the size of the tiles and increase the number of classification categories which may further refine the smoothness of the mapping segments. The mapping results via the CMA may also be used for crack type classification and causation, analyzing what type of asphalt is more prone to cracking, what type of asphalts are more suitable for different road conditions with respect to the traffic, and how to choose the best asphalt for various conditions, and finally estimating the repair and protection costs for each individual road type. Analyzing the crack propagation patterns based on geographical information of the road using the CMA provides more analytical information in combination with other data that could help during the decision making process for road construction.

## Data Availability

The corresponding author should be contacted should the reader like to have access to the crack image database and the code sources of the crack mapping algorithm.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

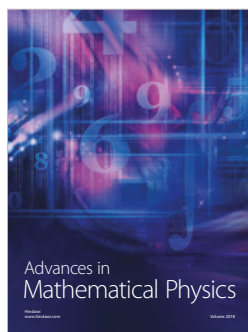
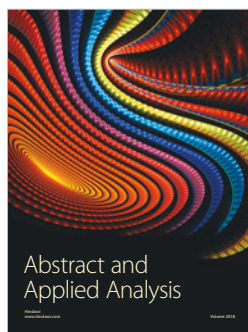
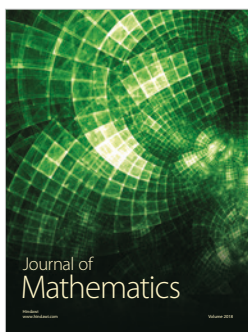
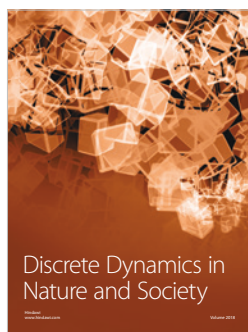
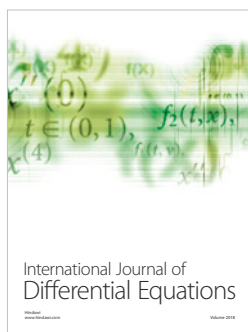
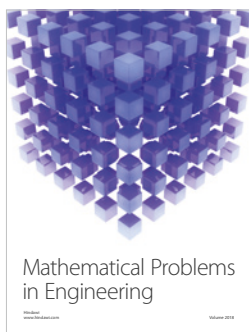
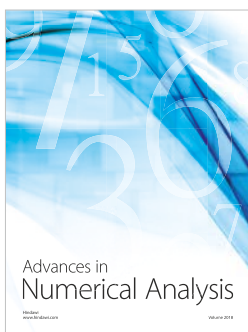
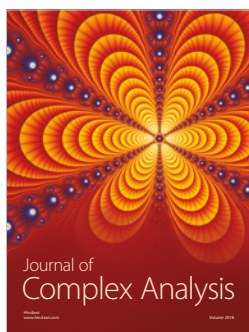
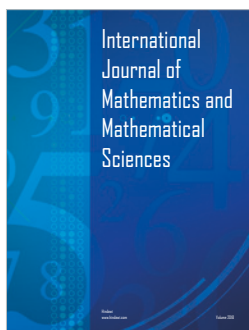
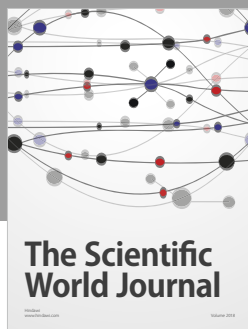
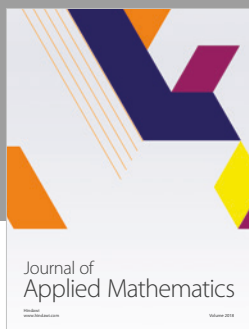
## Acknowledgments

This paper was supported by an internal grant from Lamar University, College of Engineering.

## References

- [1] J. A. Epps and M. Ardila-Coulson, *Summary of SHRP Research and Economic Benefits of Pavement Maintenance*, Federal Highway Administration, Washington, DC, USA, 1997.
- [2] National Highway Traffic Safety Administration, *National Motor Vehicle Crash Causation Survey: Report to Congress*, vol. 811, National Highway Traffic Safety Administration Technical Report DOT HS, National Highway Traffic Safety Administration, Washington, DC, USA, 2008.
- [3] L. Sun, M. Kamaliardakani, and Y. Zhang, "Weighted neighborhood pixels segmentation method for automated detection of cracks on pavement surface images," *Journal of Computing in Civil Engineering*, vol. 30, no. 2, Article ID 04015021, 2016.
- [4] A. Ouyang, C. Luo, and C. Zhou, "Surface distresses detection of pavement based on digital image processing," in *Proceedings of the International Conference on Computer and Computing Technologies in Agriculture*, pp. 368–375, Springer, Nanchang, China, October 2010.
- [5] K. Gopalakrishnan, *Advanced Pavement Health Monitoring and Management [WWW Document]*, IGI Global Videos, Hershey, PA, USA, 2016.
- [6] E. Salari and G. Bao, "Pavement distress detection and classification using feature mapping," in *Proceedings of the 2010 IEEE International Conference on Electro/Information Technology*, pp. 1–5, IEEE, Ottawa, ON, USA, September 2010.
- [7] G. Sollazzo, K. Wang, G. Bosurgi, and J. Li, "Hybrid procedure for automated detection of cracking with 3D pavement data," *Journal of Computing in Civil Engineering*, vol. 30, no. 6, Article ID 04016032, 2016.
- [8] S. Li, Y. Cao, and H. Cai, "Automatic pavement-crack detection and segmentation based on steerable matched filtering and an active contour model," *Journal of Computing in Civil Engineering*, vol. 31, no. 5, Article ID 04017045, 2017.
- [9] P. Prasanna, K. J. Dana, N. Gucunski et al., "Automated crack detection on concrete bridges," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591–599, 2016.
- [10] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
- [11] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.
- [12] P. Patidar, M. Gupta, S. Srivastava, and A. K. Nagawat, "Image de-noising by various filters for different noise," *International Journal of Computer Applications*, vol. 9, no. 4, 2010.
- [13] W. R. Vavrik, L. D. Evans, J. A. Stefanski, and S. Sargand, *PCR Evaluation—Considering Transition from Manual to Semi-Automated Pavement Distress Collection and Analysis*, Applied Research Associates, Inc., Albuquerque, USA, 2013.
- [14] A. Ayenu-Prah and N. Attoh-Okine, "Evaluating pavement cracks with bidimensional empirical mode decomposition," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, Article ID 861701, 2008.
- [15] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, "Automation of pavement surface crack detection using the continuous wavelet transform," in *Proceedings of the 2006 International Conference on Image Processing*, pp. 3037–3040, IEEE, Las Vegas, NV, USA, June 2006.
- [16] K. C. P. Wang, Q. Li, and W. Gong, "Wavelet-based pavement distress image edge detection with a trous algorithm," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2024, no. 1, pp. 73–81, 2007.
- [17] G. W. Flintsch and K. K. McGhee, *Quality Management of Pavement Condition Data Collection*, Transportation Research Board, Washington, DC, USA, 2009.
- [18] S. Liang and B. Sun, "Using wavelet technology for pavement crack detection," in *ICLEM 2010: Logistics for Sustained Economic Development: Infrastructure, Information, Integration*, pp. 2479–2484, ASCE Publications, Reston, VA, USA, 2010.
- [19] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.
- [20] H. G. Moon and J. H. Kim, "Intelligent crack detecting algorithm on the concrete crack image using neural network," in *Proceedings of the 28th International Symposium on Automation and Robotics in Construction (ISARC 2011)*, pp. 1461–1467, Seoul, South Korea, June 2011.
- [21] M. O'Byrne, B. Ghosh, F. Schoefs, and V. Pakrashi, "Regionally enhanced multiphase segmentation technique for damaged surfaces," *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 9, pp. 644–658, 2014.
- [22] M. O'Byrne, F. Schoefs, B. Ghosh, and V. Pakrashi, "Texture analysis based damage detection of ageing infrastructural elements," *Computer-Aided Civil and Infrastructure Engineering*, vol. 28, no. 3, pp. 162–177, 2013.

- [23] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme, "An innovative methodology for detection and quantification of cracks through incorporation of depth perception," *Machine Vision and Applications*, vol. 24, no. 2, pp. 227–241, 2013.
- [24] L. Wu, S. Mokhtari, A. Nazef, B. Nam, and H.-B. Yun, "Improvement of crack-detection accuracy using a novel crack defragmentation technique in image-based road assessment," vol. 30, no. 1, Article ID 04014118, 2016.
- [25] M. S. Kaseko, Z. P. Lo, and S. G. Ritchie, "Comparison of traditional and neural classifiers for pavement-crack detection," *Journal of Transportation Engineering*, vol. 120, no. 4, pp. 552–569, 1994.
- [26] N. Li, X. Hou, X. Yang, and Y. Dong, "Automation recognition of pavement surface distress based on support vector machine," in *Proceedings of the 2009 Second International Conference on Intelligent Networks and Intelligent Systems*, pp. 346–349, IEEE, Tianjin, China, November 2009.
- [27] K. Gopalakrishnan, "Deep learning in data-driven pavement image analysis and automated distress detection: a review," *Data*, vol. 3, no. 3, p. 28, 2018.
- [28] A. Vedaldi and K. Lenc, "Matconvnet: convolutional neural networks for matlab," in *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 689–692, ACM, Brisbane, Australia, October 2015.
- [29] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials*, vol. 157, pp. 322–330, 2017.
- [30] S. Dorafshan, R. J. Thomas, and M. Maguire, "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," *Construction and Building Materials*, vol. 186, pp. 1031–1045, 2018.
- [31] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiya, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," 2018, <https://arxiv.org/abs/1801.09454>.
- [32] L. Some, *Automatic Image-Based Road Crack Detection Methods*, KTH Royal Institute of Technology, Stockholm, Sweden, 2016.
- [33] Y. Kellouche, B. Boukhatem, M. Ghrici, and A. Tagnit-Hamou, "Exploring the major factors affecting fly-ash concrete carbonation using artificial neural network," *Neural Computing and Applications*, vol. 31, no. 2, pp. 969–988, 2019.
- [34] G. Xu, J. Ma, F. Liu, and X. Niu, "Automatic recognition of pavement surface crack based on BP neural network," in *Proceedings of the 2008 International Conference on Computer and Electrical Engineering*, pp. 19–22, IEEE, Phuket, Thailand, December 2008.
- [35] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3708–3712, IEEE, Phoenix, AZ, USA, September 2016.
- [36] K. C. Wang and Q. Li, *3D Laser Imaging for ODOT Interstate Network at True 1-mm Resolution*, Oklahoma Department of Transportation, Oklahoma City, OK, USA, 2014.
- [37] A. Krizhevsky and G. Hinton, *Learning Multiple Layers of Features from Tiny Images*, Technical Report TR-2009, University of Toronto, Toronto, Canada, 2009.
- [38] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.
- [39] M. B. Christopher, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York, USA, 2016.
- [40] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, pp. 369–376, ACM, Pittsburgh, PA, USA, June 2006.
- [41] K. Murphy, *Machine Learning: A Probabilistic Approach*, Massachusetts Institute of Technology, Cambridge, MA, USA, 2012.
- [42] R. Baptista and M. Poloczek, "Bayesian optimization of combinatorial structures," 2018, <https://arxiv.org/abs/1806.08838>.
- [43] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," 2014, <https://arxiv.org/abs/1403.5607>.
- [44] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning*, pp. 63–71, Springer, Berlin, Germany, 2004.
- [45] J. Mockus, *Bayesian Approach to Global Optimization: Theory and Applications*, Springer Science & Business Media, New York, USA, 2012.
- [46] E. Brochu, V. M. Cora, and N. de Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," 2010, <https://arxiv.org/abs/1012.2599>.
- [47] F. Harrell jr., "Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis," Springer, New York, NY, USA, 2015.
- [48] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-d Vision: From Images to Geometric Models*, Springer Science & Business Media, New York, USA, 2012.
- [49] S. Qiu, W. Wang, S. Wang, and K. C. P. Wang, "Methodology for accurate AASHTO PP67-10-based cracking quantification using 1-mm 3D pavement images," vol. 31, no. 2, Article ID 04016056, 2016.



Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)