



Contents lists available at SCCE

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com



Computer Vision-Based Recognition of Pavement Crack Patterns Using Light Gradient Boosting Machine, Deep Neural Network, and Convolutional Neural Network

Nhat-Duc Hoang^{1,2*}, Quoc-Lam Nguyen²

1. Lecturer, Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam

2. Lecturer, Faculty of Civil Engineering, Duy Tan University, Da Nang, 550000, Vietnam

Corresponding author: hoangnhatduc@duytan.edu.vn

 <https://doi.org/10.22115/SCCE.2023.367276.1547>

ARTICLE INFO

Article history:

Received: 25 October 2022

Revised: 07 January 2023

Accepted: 31 January 2023

Keywords:

Computer vision;

Image processing;

Pavement cracks;

Light gradient boosting machine;

Deep neural network.

ABSTRACT

The performance and serviceability of asphalt pavements have a direct influence on people's daily lives. Timely detection of pavement cracks is crucial in the task of periodic pavement survey. This paper proposes and verifies a novel computer vision-based method for recognizing pavement crack patterns. Image processing techniques, including Gaussian steerable filters, projection integrals, and image texture analyses, are employed to characterize the surface condition of asphalt pavement roads. Light Gradient Boosting Machine, Deep Neural Network, and Convolutional Neural Network are employed to recognize various patterns including longitudinal, transverse, diagonal, minor fatigue, and severe fatigue cracks. A dataset, including 12,000 samples, has been collected to construct and verify the computer vision-based approaches. Based on experiments, it can be found that all three machine learning models are capable of delivering good categorization results with an accuracy rate > 0.93 and Cohen's Kappa coefficient > 0.76 . Notably, the Light Gradient Boosting Machine has achieved the most desired performance with an accuracy rate > 0.96 and Cohen's Kappa coefficient > 0.88 .

How to cite this article: Hoang ND, Nguyen QL. Computer vision-based recognition of pavement crack patterns using light gradient boosting machine, deep neural network, and convolutional neural network. *J Soft Comput Civ Eng* 2023;7(3):21–51. <https://doi.org/10.22115/scce.2023.367276.1547>

2588-2872/ © 2023 The Authors. Published by Pouyan Press.

This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).



1. Introduction

Asphalt pavement roads are one of the most crucial components of the transportation infrastructure. Their performance and serviceability have a direct impact on people's daily lives [1,2]. Due to inclement weather conditions and increasing traffic loads, asphalt pavements quickly deteriorate over time. Among various forms of pavement distress, cracks are apparently the most widespread. Fig. 1 provides illustrations of various forms of pavement cracks. Accordingly, appropriate crack repair is crucial for ensuring the serviceability of pavements and preventing the occurrence of other more severe defects such as raveling or potholes [3].

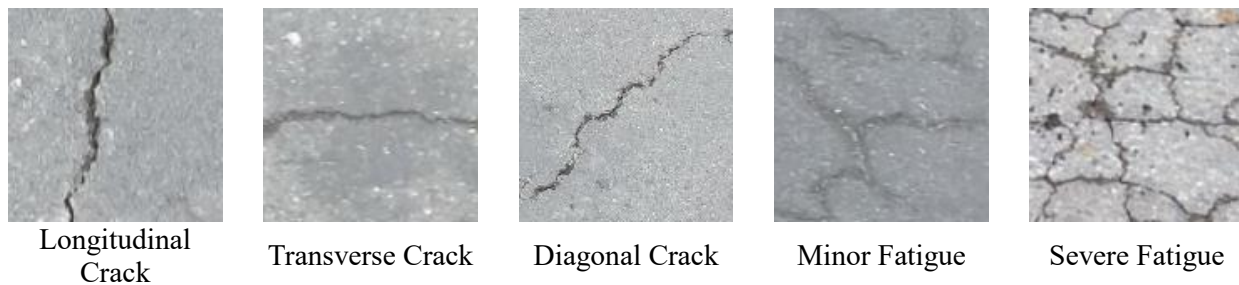


Fig. 1. Crack patterns in pavement surfaces.

Information on surface condition of asphalt pavements is particularly valuable to determine an optimal maintenance plan. In particular, information regarding the appearance of cracks as well as the type of cracks is useful for scheduling and prioritizing maintenance tasks. In order to obtain such information, periodic monitoring of pavement conditions must be performed timely and effectively. In the past decade, with the fast advancement of computer vision and machine learning techniques, various computer-based approaches for detecting and categorizing pavement distress have been proposed [4–7]. These approaches have harnessed cutting-edge machine learning and computer vision-based feature extraction approaches to obtain useful information from images of asphalt pavements. As stated by Dong and Catbas [8], the modern approaches to pavement survey provide many leverages over the conventional method, such as high productivity, safe inspection, fast data processing, and minimal interference in traffic operations.

Accordingly, previous works have been dedicated to the construction and verification of advanced models used for automatic detection and categorization of pavement cracks. Mokhtari et al. [9] carries out a comparative work that employs machine learning algorithms of decision trees (DT), k -nearest neighbors, artificial neural network (ANN), and adaptive neuro fuzzy inference system (ANFIS) for crack detection. This study shows the high performance and potential of the machine learning models that can achieve detection accuracy rates > 0.88 . However, this study only focused on crack detection and crack pattern recognition was not considered.

Cubero-Fernandez et al. [10] once again confirmed the capability of DT with an accuracy rate of 88% in detecting cracks and an accuracy rate of 80% in detecting the type of cracks. Herein, the

authors utilize edge detection and projective integrals to characterize the pavement surface condition and extract the features for the DT-based data classification process. The model proposed in [10] is capable of recognizing transverse, longitudinal, and fatigue (or alligator) cracks. Diagonal cracks and the severity of fatigue cracks were not taken into account.

A model based on Laplacian pyramid, projection integral, and least squares support vector machine has been proposed in [11] for recognizing longitudinal, transverse, diagonal, and fatigue cracks. Nevertheless, the model of interest is not capable of classifying the severity of fatigue cracks. Support vector machine coupled with steerable filters and projective integral were utilized in [12]; this method can help attain a classification accuracy rate of 87.50%. Inkoom et al. [13] put forward a model for pavement crack rating; the model employs boosted decision trees, and naïve Bayes, and k -nearest neighbors. This study observed promising performances of the employed machine learning algorithms in predicting the crack of pavement.

Besides the models that integrate image processing and machine learning-based classifiers, the deep learning method of convolutional neural network (CNN) has also been increasingly applied [14,15]. A CNN-based approach for crack classification has been introduced in [16]; this deep learning model is capable of categorizing patches cropped from pavement images. The CNN has achieved a promising classification rate of 94%. Nevertheless, the recognition of diagonal crack has not been included in this study. Hoang et al. [17] compared the capability of the CNN to that of the conventional edge detection approaches; this study found that the DL method significantly excel the image processing-based algorithms. Zhang et al. [18] recently established a pavement distress detection model based on the CNN; this deep learning method was used to detect cracks from images of the pavement surface. The method is able to recognize longitudinal, network, and fatigue cracks. Liu et al. [19] recently demonstrates the capability of the CNN in detecting crack objects from infrared images; this study emphasizes the utilization of the temperature difference between cracks and the pavement surface to construct a robust distress detection method.

Based on recent review works of Cano-Ortiz et al. [20] and Kheradmandi and Mehranfar [4], the utilization of machine intelligence in pavement performance monitoring, including crack detection, is a burgeoning trend. Therefore, there is a practical need to investigate the capability of other state-of-the-art machine learning approaches for solving the problem at hand. Furthermore, it is observable from the literature that most of the current works focus on the task of crack segmentation and crack type classification [4,21,22]. Machine-based fatigue severity recognition has rarely been investigated.

In the machine learning field, Light Gradient Boosting Machine (LightGBM), proposed in [23], is a novel gradient boosting framework based on decision trees that can be potentially used for categorizing patterns of pavement cracks. Notably, the LightGBM relies on two novel techniques of gradient-based one-side sampling and exclusive feature bundling to enhance the classification performance. These two novel techniques provide the LightGBM a significant advantage over other data classification models. In the field of pavement performance monitoring, the LightGBM has been used in [24] to estimate the pavement condition index of pavements. This

study finds that the LightGBM outperforms nonlinear regression, artificial neural networks, and random forest models. The capability of the LightGBM is also demonstrated in [25] in which cracks are automatically detected from concrete surface imagery. Nevertheless, to the best of our knowledge, none of the previous works has investigated the performance of this novel and capable machine learning method in crack pattern recognition.

In this regard, this study aims to fill the gap in the current literature by proposing an integration of the LightGBM and image processing methods to establish robust computer vision-based crack pattern recognition approach. The image processing methods, including steerable filters, projection integrals, and image texture descriptors, are used to characterize the pavement surface condition. The LightGBM is employed as a supervised learning method to categorize the image samples into six labels: non-crack, longitudinal crack, transverse crack, diagonal crack, minor fatigue crack, and severe fatigue crack. Since each type of crack indicates different damage severity and receives a different level of priority, the proposed computer vision model can be helpful for the task of pavement maintenance planning.

Additionally, an integration of deep neural network (DNN) and image processing-based feature extraction is also investigated in this study. The DNN is widely recognized as a powerful tool for pattern recognition [26]. The basic difference between a DNN and an ANN is the number of hidden layers [27]. In a DNN model, a set of multiple hidden layers acts as a hierarchical feature engineering operation and the output of one hidden layer is the input for the succeeding layer. Accordingly, the DNN possesses a high potential for analyzing multivariate and nonlinear datasets [28].

Therefore, this study constructs the computer vision-based crack pattern classifiers based on LightGBM, DNN, and CNN. Herein, the LightGBM and DNN rely on a set of extracted features obtained from the used image analysis techniques. Steerable filters and projection integrals are used to highlight the shape- and edge-based features of crack objects [12]. Meanwhile, statistical measurements of color channels and gray level co-occurrence matrices are used to account for the texture-based characteristics of pavement surface [29]. On the other hand, the CNN is able to perform the feature engineering phase automatically. Using images of asphalt pavements, this deep learning method carries out the classification of crack patterns directly. The performance of the LightGBM, DNN, and CNN is also benchmarked against that of the support vector machine (SVM). It is because the SVM was employed successfully to tackle the problem of interest in previous works [12,30,31].

To train and verify the aforementioned computer vision-models, a database, consisting of 12,000 samples and six class labels, has been acquired via surveys of road conditions in Da Nang, Vietnam. The rest of the paper is organized as follows: The second section reviews the research methodology, including the image processing techniques, LightGBM, DNN, and CNN. Results of pavement crack classifications and performance comparisons are provided in the next part of the article. The last part provides a summary of the main research findings.

2. Research method

2.1. Steerable filter and projection integral

Steerable filter (SF) coupled with projection integral (PI) is an effective tool for edge detection and shape characterization. Herein, SF [32,33] is an orientation-selective convolution kernel used to perform noise suppression and edge detection concurrently. Given a digital image within which (x, y) denotes a pixel's coordinates, a 2-dimensional Gaussian with variance σ of a pixel is described as follows [33]:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right] \tag{1}$$

The 1st order derivatives employed to compute the filters with rotation angles β of 0° and 90° are described as follows:

$$G_0 = \frac{\partial G(x, y, \sigma)}{\partial x} = -\frac{x}{2\pi\sigma^4} \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right] \tag{2}$$

$$G_{90} = \frac{\partial G(x, y, \sigma)}{\partial y} = -\frac{y}{2\pi\sigma^4} \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right] \tag{3}$$

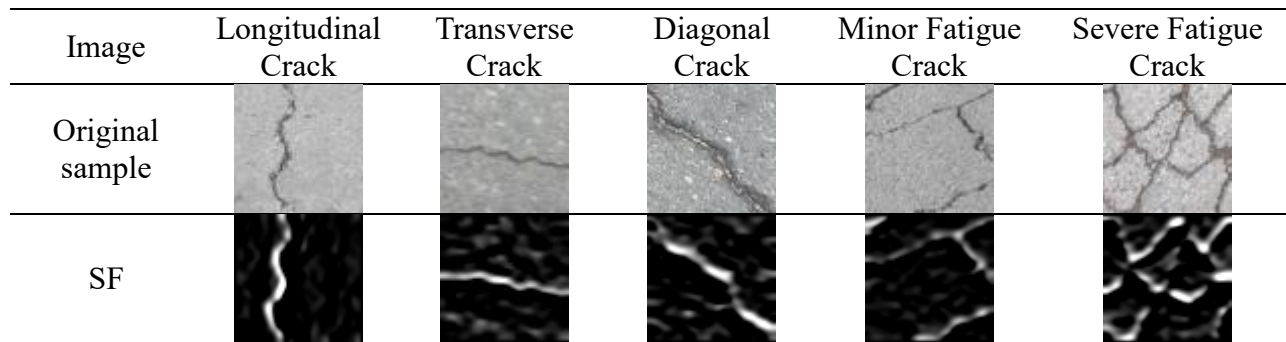


Fig. 2. Demonstrations of the SFs.

Based on the two steerable filters with $\beta = 0^\circ$ and $\beta = 90^\circ$, a PI can be constructed to characterize the shape of an object within a scene [10,34]. Demonstrations of the SFs computed for the crack patterns are provided in Fig. 2. In addition, this study relies on the horizontal PI (HPI), vertical PI (VPI), and two diagonal PIs to describe the crack patterns. The HPI and VPI are useful for recognizing transverse and longitudinal cracks [10]. The HPI and VPI are calculated as the summation of pixels' intensity along a thread x_y or y_x within an image patch as follows:

$$HPI(y) = \sum_{i \in x_y} I(i, y) \tag{4}$$

$$VPI(x) = \sum_{j \in y_x} I(x, j) \tag{5}$$

Meanwhile, the two diagonal PIs corresponding to the two projection angles of $+45^\circ$ and -45° are used for detecting diagonal cracks. These two diagonal PIs also provide helpful information for the detection of other crack patterns [35]. The diagonal PIs can be obtained by performing image

rotations (+45° and -45°) followed by a HPI calculation. Accordingly, the diagonal PIs corresponding to the two angles of rotation are denoted as diagonal PI 1 (DPI1) and diagonal PI 2 (DPI2).

2.2. Image texture descriptors

It is noted that the pavement background often contains irregular objects such as stains, blurred traffic marks, potholes, patches, etc. Therefore, using the SF coupled with PI may not be sufficient for the task of recognizing crack patterns. Accordingly, this study relies on the statistical indices of image colors and the gray-level co-occurrence matrix (GLCM) to describe the texture of the pavement surface. The mean and standard deviation (std.) of each color channel (blue, green, and red) can be useful for describing the color-based textural information [36]. The equations for computing these two indices are given by [37]:

$$\mu_C = \sum_{i=0}^N I_{C,i} \times P_C(I) \quad (6)$$

$$\sigma_C = \sum_{i=0}^N (I_{C,i} - \mu_C) \times P_C(I) \quad (7)$$

where I_C denotes an image I with respect to a color channel C . $P(I)$ represents the first-order histogram, which describes the distribution of pixel values within an image sample.

The GLCM [38] is an effective tool for characterizing the distribution of co-occurring pixel values over a patch of pavement image. In order to gain the property of rotational invariance, the GLCM is usually constructed at different values of the rotational angle α . α usually varies between 0° and 135° with an interval of 45°. The statistical indices computed from the GLCMs can be averaged to derive a set of GLCM-based features. Herein, the contrast, correlation, energy, and homogeneity of the GLCM are used to characterize the distribution of co-occurring pixel values. These indices are computed as follows [38–41]:

$$Contrast = \sum_{k=0}^{N-1} k^2 \sum_{\substack{i=1 \\ |i-j|=k}}^N \sum_{j=1}^N P(i, j) \quad (8)$$

$$Correlation = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} i \times j \times P(i, j) - \mu_X \times \mu_Y}{\sigma_X \times \sigma_Y} \quad (9)$$

$$Energy = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j) \quad (10)$$

$$Homogeneity = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{P(i, j)}{1 + (i + j)^2} \quad (11)$$

where P denotes a GLCM. $N_g = 256$ is the number of gray level values. μ_X, μ_Y, σ_X , and σ_Y represent the means and standard deviations of the marginal distribution of a GLCM.

2.3. Light gradient boosting machine

The Light Gradient Boosting Machine (LightGBM), proposed in [23], is a powerful gradient boosting framework based on classification trees. This machine learning approach combines a set of weak learners to construct a highly robust one (refer to **Fig. 3**). A LightGBM model is built sequentially by iteratively minimizing the classification error committed by previous one [42,43]. Herein, the classification error is measured by a loss function. The ensemble model $f(x)$ is established by combining a set of M decision trees as follows:

$$f(x) = \sum_{m=1}^M f_m(x) \tag{12}$$

where f_1, f_2, \dots, f_M denote individual decision trees.

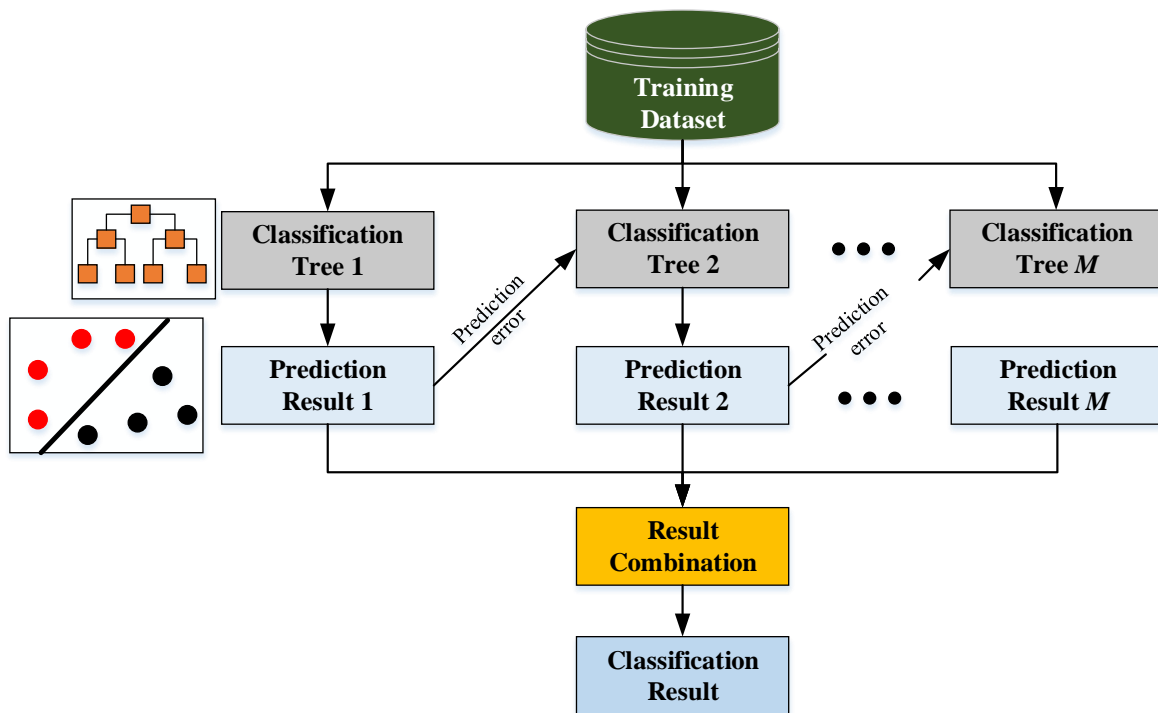


Fig. 3. The LightGBM ensemble model.

A leaf-wise algorithm is utilized by the LightGBM model to grow the trees vertically. A leaf that results in the most reduction in the loss function is chosen to split and grow the decision tree. Notably, the training performance of the LightGBM is enhanced by the Gradient-based One-Side Sampling (GOSS) to express the importance of data samples. The GOSS helps the model to focus on data samples having larger gradients and neglect ones having small gradients. It is because the samples associated with small gradients are fitted well and this results in lower classification errors. Accordingly, the LightGBM is able to steer the learning phase towards more informative data points. Moreover, the Exclusive Feature Bundling (EFB) technique is also employed to cope with sparse datasets. This technique aims to combine mutually exclusive features to concurrently achieve feature reduction and preserve the most informative predictor variables.

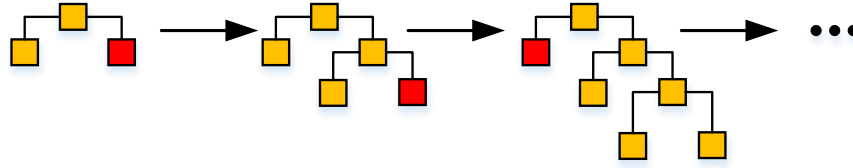


Fig. 4. The leaf-wise tree growth employed by the LightGBM.

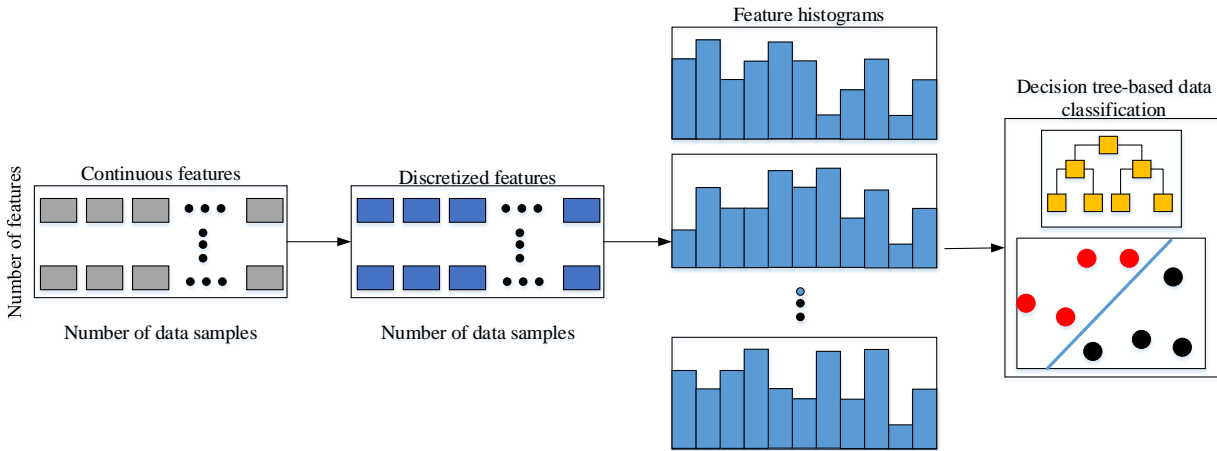


Fig. 5. The histogram-based algorithm.

Notably, the LightGBM grows individual decision trees in a leaf-wise manner [25]. The leaf-wise tree growing process is demonstrated in Fig. 4. A significant advantage of the leaf-wise tree growth is that it can effectively increase the complexity of a tree. Thus, the LightGBM is capable of modeling sophisticated mapping functions. In addition, a histogram-based algorithm (refer to Fig. 5) is employed to convert the original continuous features into a small number of bins (e.g. 255 bins). These bins can be utilized to establish the histograms that represent the distribution of the input variables. Statistical indices (e.g. the number of data instances and the sum of gradients) can be computed for each bin. The optimal split points used for training the weak learners can be effectively determined via these statistical indices.

Notably, the histogram-based algorithm is able to reduce the computational cost of the training phase because the scanning of the whole ranges of features for determining a split point is not required [23]. Additionally, this algorithm also enhances the generalization property of the constructed model because the learning phase of the LightGBM is less susceptible to noise [44].

2.4. Deep neural network

The Deep Neural Network (DNN) is a popular machine learning method for pattern recognition [45,46]. The structure of a DNN model includes an input layer, a set of hidden layers, and an output layer (refer to Fig. 6). This machine learning method typically utilizes a set of hidden layers to process the numerical variables provided by the input layer [47]. The input nodes in the first layer transmit the input signals $x = \{f_1, f_2, \dots, f_D\}$ is a D -dimensional vector with f_j as the feature and $j \in \{1, 2, \dots, D\}$. These signals are the features computed by the aforementioned image processing techniques.

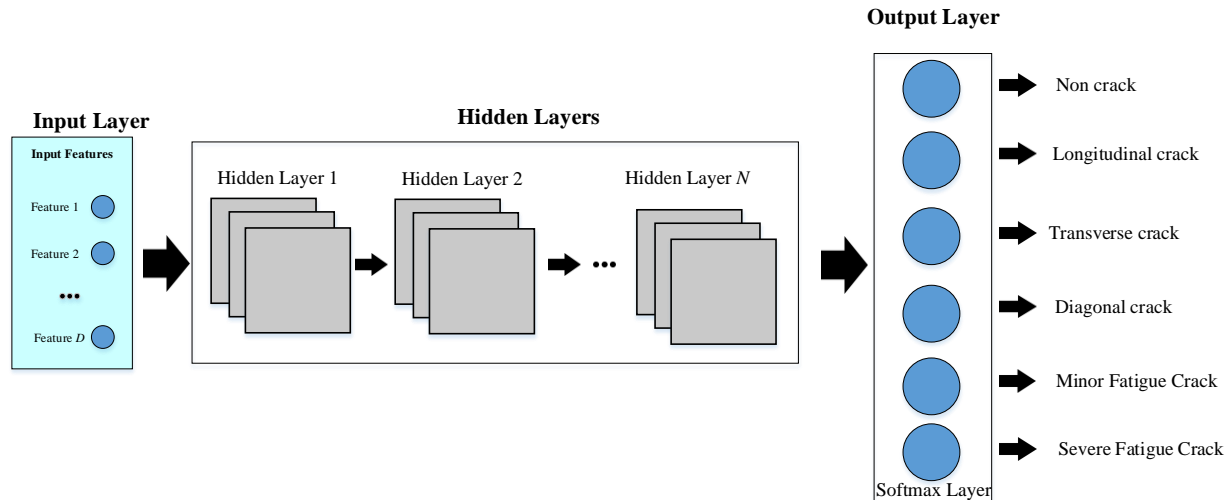


Fig. 6. The general structure of a deep learning model used for pavement fatigue classification.

The stacked hidden layers act as a sophisticated feature engineering operator. This operator analyzes the input signal of the preceding layer, creates more informative features, and transmits them to the subsequent layer. Finally, the output layer uses a softmax function to calculate the probabilities of the class labels. Herein, the problem at hand includes six class labels as demonstrated in Fig. 6. Notably, to alleviate the problem of vanishing gradient, the Rectified Linear Unit (ReLU) activation function should be used in the hidden layer [27]. In addition, to adapt the DNN's weights according to the collected dataset, this work resorts to the state-of-the-art adaptive moment estimation (Adam) algorithm [48].

2.5. Convolutional neural network

The Convolutional Neural Network (CNN) is a popular deep learning method used for classifying image datasets [49–52]. Different from the LightGBM and DNN, the CNN is able to perform the feature computation process autonomously without the need of image processing-based feature extraction. Therefore, the input of a CNN model is a color image with the size $r \times c$ and a depth of three representing the color channels [53]. The advantage of the CNN is the ability to learn the data representations via a hierarchical organization of multiple convolutional layers. These layers have the role of extracting higher-level features directly from image samples.

The structure of a CNN model used for crack pattern recognition is depicted in Fig. 7. The structure of a CNN model typically includes of a set of convolutional layers; each layer consists of kernels for computing the various features of the input images such as edges, shapes, and textures [54,55]. A pooling layer is often put after a convolutional layer to decrease the spatial size of the image. The output of the final pooling layer is transmitted to a fully-connected layer to compute the probability of each class label. In this study, the Adam [48] algorithm is also used to optimize the parameters of the CNN used for pavement crack detection and classification.

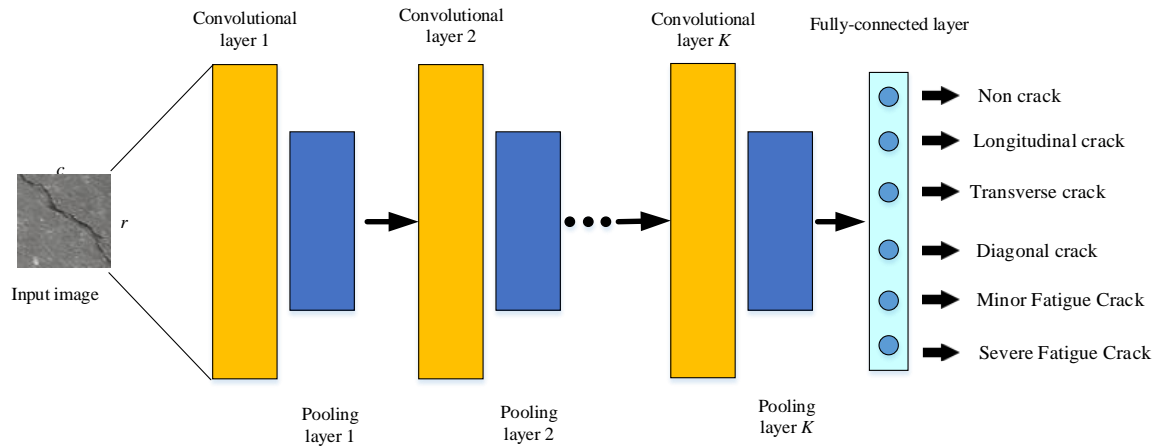


Fig. 7. The CNN model structure.

3. Comparison and results

This section of the study reports the performance of the newly developed computer vision approaches for classifying different patterns of the pavement cracks. It is noted that the data classification processes of the LightGBM and DNN rely on the image processing techniques of SF, PI, as well as the aforementioned texture descriptors. Meanwhile, the computation of the features that are relevant for the categorization task is performed automatically by the CNN. In this study, the LightGBM model is built with the help of the Python library provided in [56]. This study relies on the scikit-learn library [57] to construct the DNN model. In addition, the MATLAB deep learning toolbox [58] is employed to build the CNN model.

The LightGBM, DNN, and CNN are used to categorize input image samples into six distinctive classes of non-crack (C0), longitudinal crack (C1), transverse crack (C2), diagonal crack (C3), minor fatigue crack (C4), and severe fatigue crack (C5). It is noted that different patterns of crack result from different forms of pavement failures [59]. In addition, each cracking patterns may require a different approach of rehabilitation [60]. For instance, longitudinal and transverse cracks can be easily repaired with sealant. Meanwhile, to recover an area suffered from fatigue cracks, a full depth patch-up is usually required. The flowchart of the proposed approach for the task of crack pattern classification is summarized in Fig. 8. To train and test the aforementioned computer vision-based approaches, field surveys in Da Nang city (Vietnam) have been carried out to collect an image dataset of pavement images.

The image samples have been captured by the 16.2-megapixel resolution Nikon D5100 and the 18-megapixel resolution Canon EOS M10 at a distance of about 1.2 m above the road surface. Each class of interest contains 2000 samples. Therefore, the total number of image samples is 12000. In addition, to enhance the speed of the image texture computation and data classification, the image sample size is set to be 64x64 pixels. It is noted that the ground truth labels of image samples are determined by human inspectors. It is noted that the collected image dataset has been randomly divided into a training set (70%) and a testing set (30%). The former is utilized for training the machine learning models; the latter is reserved for assessing the generalization of the trained models. Illustrations of the collected image dataset are provided in Fig. 9.

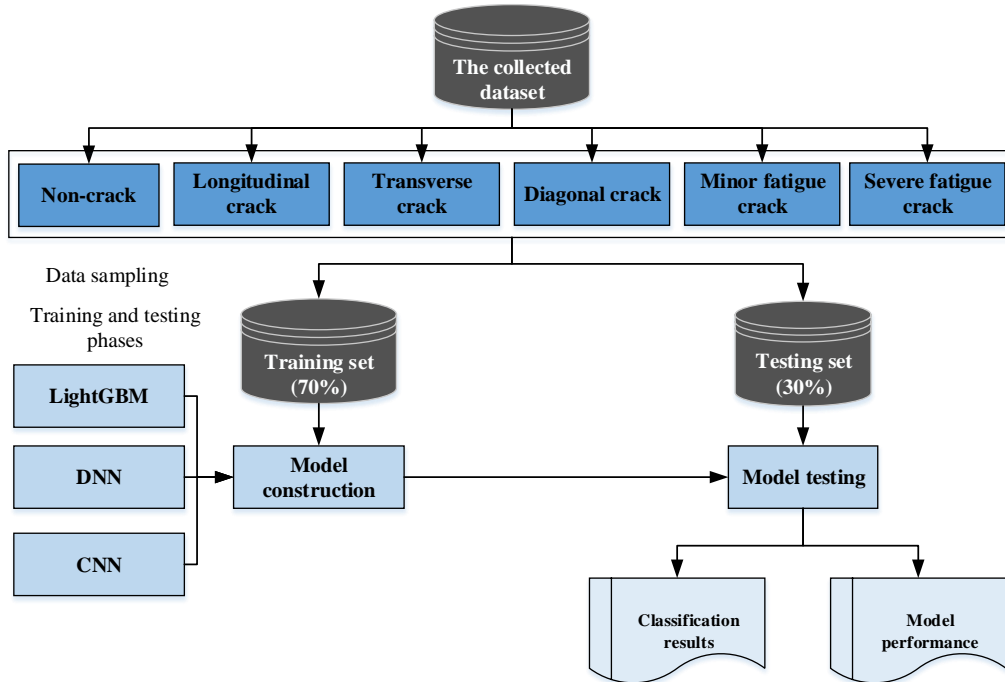


Fig. 8. Flowchart of the model training and testing phases

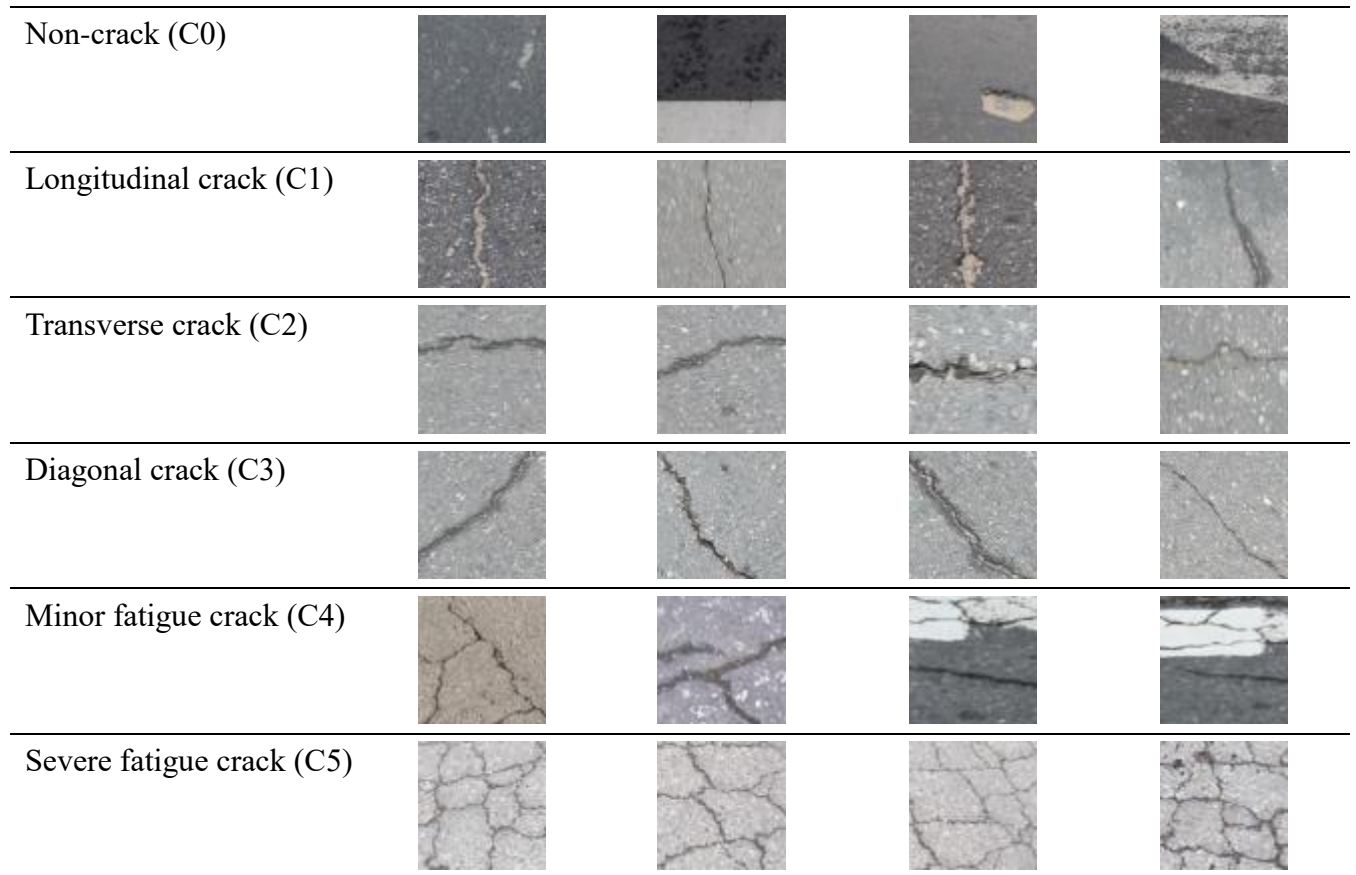


Fig. 9. The collected image dataset.

Table 1
Statistical descriptions of the variables in the dataset.

| Feature | Description | Min | Mean | Std | Skewness | Max |
|---------|-----------------------|--------|---------|--------|----------|---------|
| X1 | Minimum value of VPI | 0.000 | 0.184 | 0.123 | 1.630 | 0.902 |
| X2 | Mean of VPI | 0.015 | 0.305 | 0.158 | 1.238 | 1.118 |
| X3 | Std. of VPI | 0.007 | 0.071 | 0.038 | 1.384 | 0.313 |
| X4 | Skewness of VPI | -2.315 | 0.374 | 0.627 | 0.281 | 3.981 |
| X5 | Maximum value of VPI | 0.050 | 0.465 | 0.211 | 0.998 | 1.598 |
| X6 | Minimum value of HPI | 0.000 | 0.134 | 0.084 | 1.424 | 0.673 |
| X7 | Mean of HPI | 0.015 | 0.252 | 0.123 | 1.169 | 0.958 |
| X8 | Std. of HPI | 0.006 | 0.072 | 0.041 | 1.706 | 0.435 |
| X9 | Skewness of HPI | -1.832 | 0.464 | 0.671 | 0.517 | 3.143 |
| X10 | Maximum value of DPI1 | 0.041 | 0.420 | 0.190 | 0.988 | 1.364 |
| X11 | Mean of DPI1 | 0.052 | 0.265 | 0.121 | 1.222 | 0.873 |
| X12 | Std. of DPI1 | 0.029 | 0.174 | 0.107 | 1.840 | 0.867 |
| X13 | Skewness of DPI1 | -1.441 | 0.663 | 0.691 | 1.076 | 3.629 |
| X14 | Maximum value of DPI1 | 0.116 | 0.734 | 0.450 | 1.971 | 3.847 |
| X15 | Mean of DPI2 | 0.049 | 0.262 | 0.120 | 1.171 | 0.885 |
| X16 | Std. of DPI2 | 0.020 | 0.172 | 0.109 | 1.778 | 0.963 |
| X17 | Skewness of DPI2 | -1.484 | 0.641 | 0.691 | 0.988 | 3.643 |
| X18 | Maximum value of DPI2 | 0.093 | 0.726 | 0.457 | 1.933 | 3.984 |
| X19 | Mean of blue channel | 65.008 | 141.082 | 27.603 | 0.138 | 237.421 |
| X20 | Std. of blue channel | 1.247 | 10.078 | 6.844 | 2.496 | 60.006 |
| X21 | Mean of green channel | 53.071 | 141.210 | 28.005 | 0.075 | 240.535 |
| X22 | Std. of green channel | 1.273 | 10.518 | 7.574 | 2.471 | 61.579 |
| X23 | Mean of red channel | 41.010 | 140.787 | 29.068 | 0.027 | 240.384 |
| X24 | Std. of red channel | 1.442 | 10.692 | 8.230 | 2.675 | 61.882 |
| X25 | GLCM's contrast | 0.811 | 27.907 | 25.528 | 2.711 | 274.452 |
| X26 | GLCM's correlation | 0.715 | 0.893 | 0.045 | -0.261 | 0.997 |
| X27 | GLCM's energy | 0.023 | 0.072 | 0.028 | 1.388 | 0.251 |
| X28 | GLCM's homogeneity | 0.103 | 0.332 | 0.095 | 0.865 | 0.718 |

As mentioned in the previous section, the GSF-based PIs, the statistical measurement of the color channels, and the properties of the GLCM are employed as feature extractors for the LightGBM and DNN models. The GSF-based PIs compute four PIs: VPI, HPI, DP1, and DP2. Each of the VPI and HPI yields five statistical indices of minimum, mean, standard deviation (std.), skewness, and maximum. In addition, since the minimum of the DPI1 and DPI2 is always zero, each of the diagonal projection integrals yields four statistical indices of mean, standard

deviation (std.), skewness, and maximum. In total, the PI-related feature extractor computes 18 crack pattern's influencing factors. These factors help delineate the crack patterns within an image patch. Furthermore, to take into account various colored objects such as traffic marks, oil stains, etc. existing on the road surface, the statistical measurements (mean and std.) of three color channels (blue, green, and red) are calculated. Hence, the color-related features include six crack pattern's influencing factors. Finally, the properties of contrast, correlation, energy, and homogeneity are computed from the GLCM of each image sample. Accordingly, the total number of the extracted features is 28. The statistical descriptions of the extracted variables are summarized in Table 1. Table 2 provides an illustration of the computed dataset. The distributions of the extracted features with respect to different labels are depicted in Fig. 10.

Table 2
The collected dataset.

| Sample | X1 | X2 | X3 | X4 | X5 | ... | X24 | X25 | X26 | X27 | X28 | Label |
|--------|-------|-------|-------|--------|-------|-----|--------|--------|-------|-------|-------|-------|
| 1 | 0.128 | 0.191 | 0.049 | 0.018 | 0.268 | ... | 8.524 | 1.333 | 0.992 | 0.123 | 0.662 | 0 |
| 2 | 0.095 | 0.219 | 0.062 | -0.005 | 0.338 | ... | 9.722 | 11.773 | 0.929 | 0.062 | 0.339 | 0 |
| 3 | 0.132 | 0.217 | 0.037 | -0.015 | 0.287 | ... | 12.245 | 10.588 | 0.944 | 0.055 | 0.334 | 0 |
| ... | ... | ... | ... | ... | ... | ... | | | | | | ... |
| 2001 | 0.071 | 0.163 | 0.044 | 0.163 | 0.263 | ... | 4.336 | 7.522 | 0.881 | 0.087 | 0.397 | 1 |
| 2002 | 0.118 | 0.242 | 0.088 | 0.604 | 0.435 | ... | 9.676 | 19.138 | 0.919 | 0.072 | 0.307 | 1 |
| 2003 | 0.062 | 0.210 | 0.157 | 1.012 | 0.552 | ... | 8.494 | 18.745 | 0.910 | 0.106 | 0.411 | 1 |
| ... | ... | ... | ... | ... | ... | ... | | | | | | ... |
| 4001 | 0.263 | 0.361 | 0.047 | -0.148 | 0.446 | ... | 7.818 | 20.156 | 0.902 | 0.071 | 0.344 | 2 |
| 4002 | 0.171 | 0.269 | 0.050 | 0.371 | 0.378 | ... | 6.681 | 19.018 | 0.857 | 0.060 | 0.282 | 2 |
| 4003 | 0.240 | 0.330 | 0.051 | 0.402 | 0.444 | ... | 7.871 | 22.783 | 0.902 | 0.065 | 0.331 | 2 |
| ... | ... | ... | ... | ... | ... | ... | | | | | | ... |
| 6001 | 0.380 | 0.480 | 0.071 | 0.897 | 0.682 | ... | 17.028 | 73.989 | 0.897 | 0.048 | 0.219 | 3 |
| 6002 | 0.113 | 0.190 | 0.041 | 0.373 | 0.279 | ... | 5.330 | 13.600 | 0.868 | 0.077 | 0.335 | 3 |
| 6003 | 0.279 | 0.390 | 0.064 | 0.938 | 0.591 | ... | 14.379 | 30.809 | 0.939 | 0.059 | 0.268 | 3 |
| ... | ... | ... | ... | ... | ... | ... | | | | | | ... |
| 8001 | 0.164 | 0.257 | 0.047 | 0.084 | 0.353 | ... | 5.010 | 15.221 | 0.838 | 0.074 | 0.332 | 4 |
| 8002 | 0.168 | 0.260 | 0.064 | 0.770 | 0.422 | ... | 7.168 | 15.412 | 0.901 | 0.088 | 0.362 | 4 |
| 8003 | 0.128 | 0.218 | 0.051 | 0.438 | 0.321 | ... | 5.989 | 17.377 | 0.864 | 0.074 | 0.338 | 4 |
| ... | ... | ... | ... | ... | ... | ... | | | | | | ... |
| 11998 | 0.385 | 0.541 | 0.130 | 1.003 | 0.862 | ... | 28.658 | 49.262 | 0.967 | 0.027 | 0.196 | 5 |
| 11999 | 0.259 | 0.351 | 0.058 | 0.180 | 0.479 | ... | 8.797 | 16.925 | 0.908 | 0.054 | 0.310 | 5 |
| 12000 | 0.297 | 0.475 | 0.103 | 0.034 | 0.639 | ... | 10.820 | 56.517 | 0.866 | 0.042 | 0.206 | 5 |

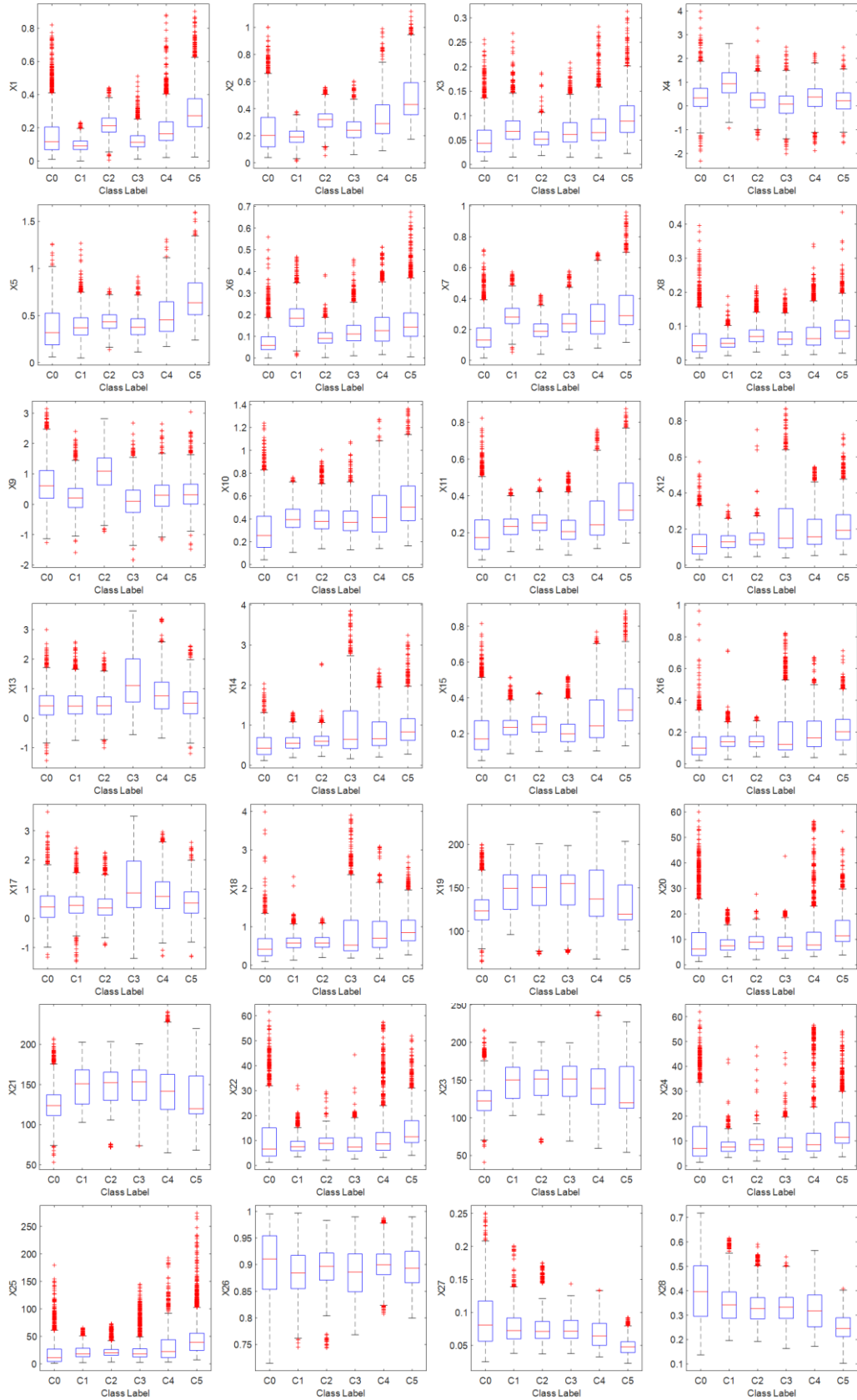


Fig. 10. Boxplots of the variables with respect to different class labels.

It is noted that the feature extractors related to the GSF-based PIs and the statistical measurement of three color channels are coded in Python. Meanwhile, the GLCM-based features are calculated with the assistance of the scikit-image library [61]. As can be observed from Fig. 10, the input variables have different ranges. Hence, to standardize the input ranges, this study has used the Z-score equation. The formula of this data standardization is given by:

$$X_Z = \frac{X_D - M_X}{STD_X} \tag{13}$$

where X_Z and X_D are the normalized and the original variable, respectively. M_X and STD_X denote the mean and the std. of the original variable, respectively.

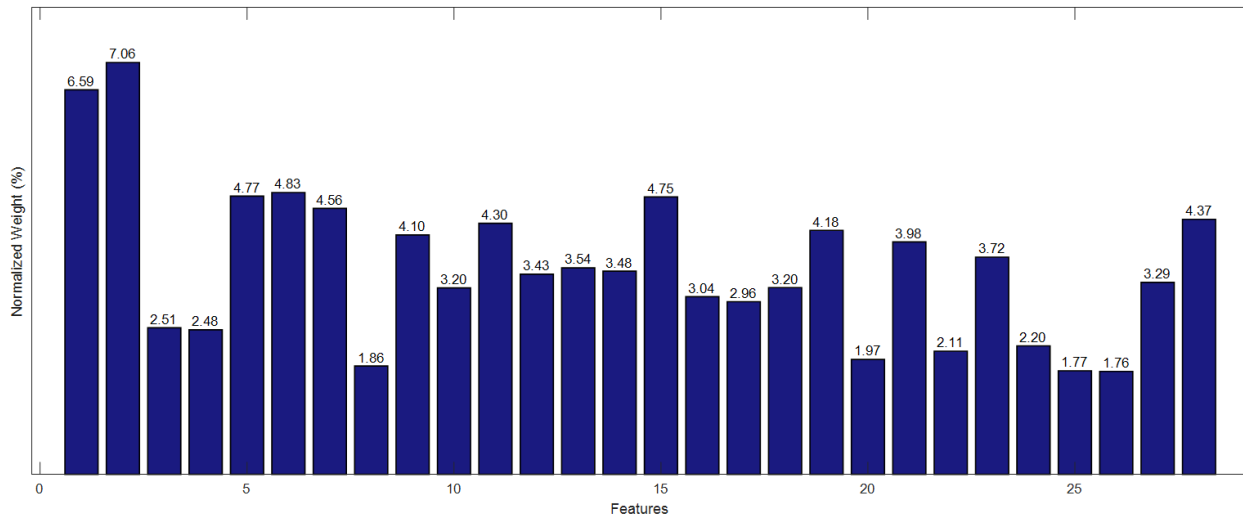


Fig. 11. ReliefF based feature ranking.

After the dataset has been computed, it is beneficial to inspect the relevancy of each input variable with respect to the class labels. Herein, the ReliefF algorithm [62,63] is employed to compute a feature weight of each variable. These feature weights indicate the relevancy of the crack pattern’s influencing factors. The higher the weight is, the more relevant the input variable is. The ReliefF is selected in this study because it is capable of modeling interactions among variables, dealing with noisy data, and handling multi-pattern recognition problems [64]. The feature weights of the extracted variables are presented in Fig. 11. Observably, $X_1, X_2, X_5, X_6, X_7,$ and X_{15} have the high importance weights. Meanwhile, the feature weights of $X_{25}, X_{26}, X_8,$ and X_{20} are comparatively lower than those of other factors. However, because all of the feature weights are greater than zeros, all of the features should be used in the pattern classification phase.

Moreover, to evaluate the performance of the computer vision-based approaches, this study relies on the indices of classification accuracy rate (CAR), precision, recall, F1 score, area under the receiver operating characteristic curve (AUC), and Cohen’s Kappa coefficient. For the construction of the receiver operating characteristic curve, readers are guided to the previous works of [65,66]. The equations used to calculate other indices are given by [67,68]:

$$\text{CAR} = \frac{N_C}{N_A} 100\% \quad (14)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (15)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (16)$$

$$\text{F1 Score} = \frac{2TP}{2TP + FP + FN} \quad (17)$$

$$\text{Kappa} = \frac{2 \times (TP \times TN - FN \times FP)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)} \quad (18)$$

where N_C and N_A are the numbers of correctly predicted samples and the total number of samples, respectively. FN , FP , TP , and TN are the false negative, false positive, true positive, and true negative samples, respectively.

Table 3

The CNN model configuration.

| CNN layers | Convolutional layers | | Pooling layers |
|------------|----------------------|-------------|----------------|
| | Number of filters | Filter size | Filter size |
| 1 | 64 | 16 | 2 |
| 2 | 128 | 8 | 2 |
| 3 | 128 | 6 | 2 |
| 4 | 256 | 4 | 2 |

Based on the dataset consisting of 12000 instances, this study constructs the data classifiers that are based on the LightGBM, DNN, and CNN. It is noted that to specify the hyper-parameters of the LightGBM and DNN, five-fold cross validation processes are used. The suitable hyper-parameters of the LightGBM, including the number of leaves, the number of estimators, and the maximum depth, are found to be 21, 100, and 6, respectively.

The appropriate setting of the DNN is as follows: the number of hidden layers = 4 and the number of neurons in each hidden layer = 40. In addition, the appropriate setting of the CNN model is evaluated via recommendation of previous works [50,69,70] and trial runs with the collected image dataset. The employed hyper-parameters of the CNN including the number of layers, the number of filters in a layer, and the filter size are shown in Table 3. Herein, the CNN includes four convolutional layers. The maximum number of training epochs for the CNN is 5000 and the batch size is selected to be 64. In addition, the results of the LightGBM, DNN, and CNN are also benchmarked to that of the Support Vector Machine (SVM). It is because the SVM has been successfully applied for crack detection and categorization in previous works. This

study also utilizes the scikit-learn library [57] to build the SVM model. It is noted that the hyperparameters of the SVM model, including the penalty coefficient and the radial basis kernel function's coefficient, are determined via the five-fold cross validation process.

As mentioned earlier, the collected dataset, consisting of 12000 samples and 28 features, is randomly separated into a training set (70%) and a testing set (30%). The former set is used for model training. The latter set is used to inspect the generalization capability of the model. In addition, to alleviate the effect of random data sampling, this study has repeated the model training and testing phases 20 times. In each time, 30% of the data samples are randomly drawn from the original dataset to form a testing set. The statistical indices including mean and standard deviation (Std.) of the employed measurement metrics (CAR, precision, recall, F1 score, AUC, and Kappa coefficient) are reported in Table 4. The result comparisons are graphically shown in Fig. 12.

In Table 4, the model performance with respect to each class label is presented. As can be seen from the experimental results, the LightGBM has achieved the most desired outcomes for all class labels. Herein, the Cohen's Kappa coefficient is the focusing performance measurement index because this coefficient provides a robust measure that takes into account both true positives and true negatives [71]. The LightGBM has attained the Cohen's Kappa coefficients of 0.8828, 0.9677, 0.9628, 0.9666, 0.9150, and 0.9046 for the six class labels of interest. For detecting non-crack, longitudinal crack, diagonal crack, minor fatigue crack, and severe fatigue crack, the DNN is the second best model, followed by the CNN model. In the task of detecting transverse cracks, the CNN (Kappa coefficient = 0.9430) outperforms the DNN (Kappa coefficient = 0.9326). In addition, the performance of the SVM is worse than that of the LightGBM, DNN, and CNN in terms of most performance measurement metrics.

The Kappa coefficients of the LightGBM are higher than 0.9 in all classes except C0 (non-crack). It is understandable because this category contains a large instances of diverse objects such as potholes, raveling, sealed crack, stains, etc. The complex texture of the pavement background in this class causes a higher miss-classification rate of the model. In other classes, the Kappa coefficients of the LightGBM are all higher than 0.9; this fact indices that the numbers of false positive and false negative cases predicted by the LightGBM are desirably low. With Kappa coefficients > 0.9 , the DNN is also highly capable of detecting instances containing the longitudinal crack, transverse crack, and diagonal crack. The CNN also shows good performances in classifying data instances from the classes of longitudinal crack (Kappa coefficients = 0.9431) and transverse crack (Kappa coefficients = 0.9430).

In addition, this study has employed the Wilcoxon signed-rank test [72] to reliably assess the models' predicted outcomes. The Wilcoxon signed-rank test is a non-parameter test widely used for pairwise comparison of model performances [17]. Herein, the data obtained from 20 independent runs of the employed models is subject to this hypothesis test. Moreover, the significant level (p -value) of the test is selected to be 0.05. The test is applied for pairwise comparison between the LightGBM and other benchmark approaches. With p -values = 0.0001, it is able to reject the null hypothesis of equal performance and confirm the superiority of the LightGBM. To better demonstrate the classification performance of the LightGBM, its average confusion matrix obtained from 20 independent runs is shown in Fig. 13.

Table 4
Model performance comparison.

| Class | Indices | LightGBM | | DNN | | CNN | | SVM | |
|------------------------------|-----------|----------|--------|--------|--------|--------|--------|--------|--------|
| | | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| C0 (Non-crack) | CAR | 0.9680 | 0.0030 | 0.9542 | 0.0050 | 0.9416 | 0.0132 | 0.9186 | 0.0038 |
| | Precision | 0.9204 | 0.0120 | 0.8718 | 0.0348 | 0.8280 | 0.0389 | 0.7475 | 0.0144 |
| | Recall | 0.8843 | 0.0134 | 0.8537 | 0.0251 | 0.8199 | 0.0468 | 0.7706 | 0.0188 |
| | F1 Score | 0.9019 | 0.0086 | 0.8617 | 0.0119 | 0.8237 | 0.0412 | 0.7587 | 0.0113 |
| | AUC | 0.9914 | 0.0016 | 0.9810 | 0.0027 | 0.9651 | 0.0120 | 0.9504 | 0.0046 |
| | Kappa | 0.8828 | 0.0103 | 0.8343 | 0.0148 | 0.7888 | 0.0491 | 0.7098 | 0.0131 |
| C1 (Longitudinal Crack) | CAR | 0.9910 | 0.0014 | 0.9868 | 0.0021 | 0.9842 | 0.0047 | 0.9717 | 0.0029 |
| | Precision | 0.9707 | 0.0067 | 0.9577 | 0.0141 | 0.9497 | 0.0141 | 0.9090 | 0.0137 |
| | Recall | 0.9756 | 0.0063 | 0.9653 | 0.0142 | 0.9555 | 0.0156 | 0.9219 | 0.0130 |
| | F1 Score | 0.9731 | 0.0041 | 0.9613 | 0.0055 | 0.9526 | 0.0140 | 0.9153 | 0.0087 |
| | AUC | 0.9986 | 0.0006 | 0.9975 | 0.0012 | 0.9950 | 0.0036 | 0.9934 | 0.0012 |
| | Kappa | 0.9677 | 0.0049 | 0.9534 | 0.0068 | 0.9431 | 0.0168 | 0.8983 | 0.0104 |
| C2 (Transverse Crack) | CAR | 0.9898 | 0.0011 | 0.9813 | 0.0023 | 0.9840 | 0.0028 | 0.9504 | 0.0030 |
| | Precision | 0.9684 | 0.0087 | 0.9425 | 0.0130 | 0.9430 | 0.0109 | 0.8719 | 0.0164 |
| | Recall | 0.9695 | 0.0058 | 0.9456 | 0.0148 | 0.9624 | 0.0093 | 0.8294 | 0.0161 |
| | F1 Score | 0.9689 | 0.0033 | 0.9439 | 0.0070 | 0.9526 | 0.0081 | 0.8499 | 0.0101 |
| | AUC | 0.9977 | 0.0010 | 0.9948 | 0.0017 | 0.9947 | 0.0018 | 0.9799 | 0.0021 |
| | Kappa | 0.9628 | 0.0039 | 0.9326 | 0.0084 | 0.9430 | 0.0098 | 0.8203 | 0.0118 |
| C3 (Diagonal Crack) | CAR | 0.9906 | 0.0015 | 0.9833 | 0.0033 | 0.9659 | 0.0115 | 0.9564 | 0.0035 |
| | Precision | 0.9719 | 0.0061 | 0.9505 | 0.0209 | 0.8948 | 0.0381 | 0.8952 | 0.0135 |
| | Recall | 0.9727 | 0.0060 | 0.9500 | 0.0139 | 0.9019 | 0.0328 | 0.8344 | 0.0181 |
| | F1 Score | 0.9723 | 0.0043 | 0.9500 | 0.0095 | 0.8982 | 0.0336 | 0.8636 | 0.0115 |
| | AUC | 0.9990 | 0.0003 | 0.9962 | 0.0012 | 0.9895 | 0.0066 | 0.9844 | 0.0021 |
| | Kappa | 0.9666 | 0.0052 | 0.9400 | 0.0114 | 0.8777 | 0.0405 | 0.8377 | 0.0135 |
| C4 (Minor Fatigue Crack) | CAR | 0.9762 | 0.0030 | 0.9609 | 0.0046 | 0.9329 | 0.0100 | 0.9135 | 0.0052 |
| | Precision | 0.9180 | 0.0114 | 0.8704 | 0.0220 | 0.7986 | 0.0335 | 0.7311 | 0.0207 |
| | Recall | 0.9410 | 0.0110 | 0.8960 | 0.0289 | 0.8000 | 0.0310 | 0.7584 | 0.0171 |
| | F1 Score | 0.9293 | 0.0092 | 0.8824 | 0.0139 | 0.7991 | 0.0293 | 0.7443 | 0.0144 |
| | AUC | 0.9953 | 0.0010 | 0.9875 | 0.0027 | 0.9668 | 0.0085 | 0.9474 | 0.0051 |
| | Kappa | 0.9150 | 0.0109 | 0.8590 | 0.0164 | 0.7588 | 0.0353 | 0.6923 | 0.0174 |
| C5 (Severe Fatigue Crack) | CAR | 0.9733 | 0.0031 | 0.9623 | 0.0036 | 0.9505 | 0.0057 | 0.9364 | 0.0041 |
| | Precision | 0.9177 | 0.0143 | 0.8966 | 0.0149 | 0.8619 | 0.0178 | 0.7990 | 0.0185 |
| | Recall | 0.9239 | 0.0125 | 0.8749 | 0.0231 | 0.8377 | 0.0249 | 0.8268 | 0.0172 |
| | F1 Score | 0.9207 | 0.0088 | 0.8853 | 0.0129 | 0.8494 | 0.0181 | 0.8125 | 0.0131 |
| | AUC | 0.9947 | 0.0008 | 0.9882 | 0.0024 | 0.9816 | 0.0036 | 0.9658 | 0.0038 |
| | Kappa | 0.9046 | 0.0107 | 0.8628 | 0.0148 | 0.8199 | 0.0214 | 0.7742 | 0.0153 |

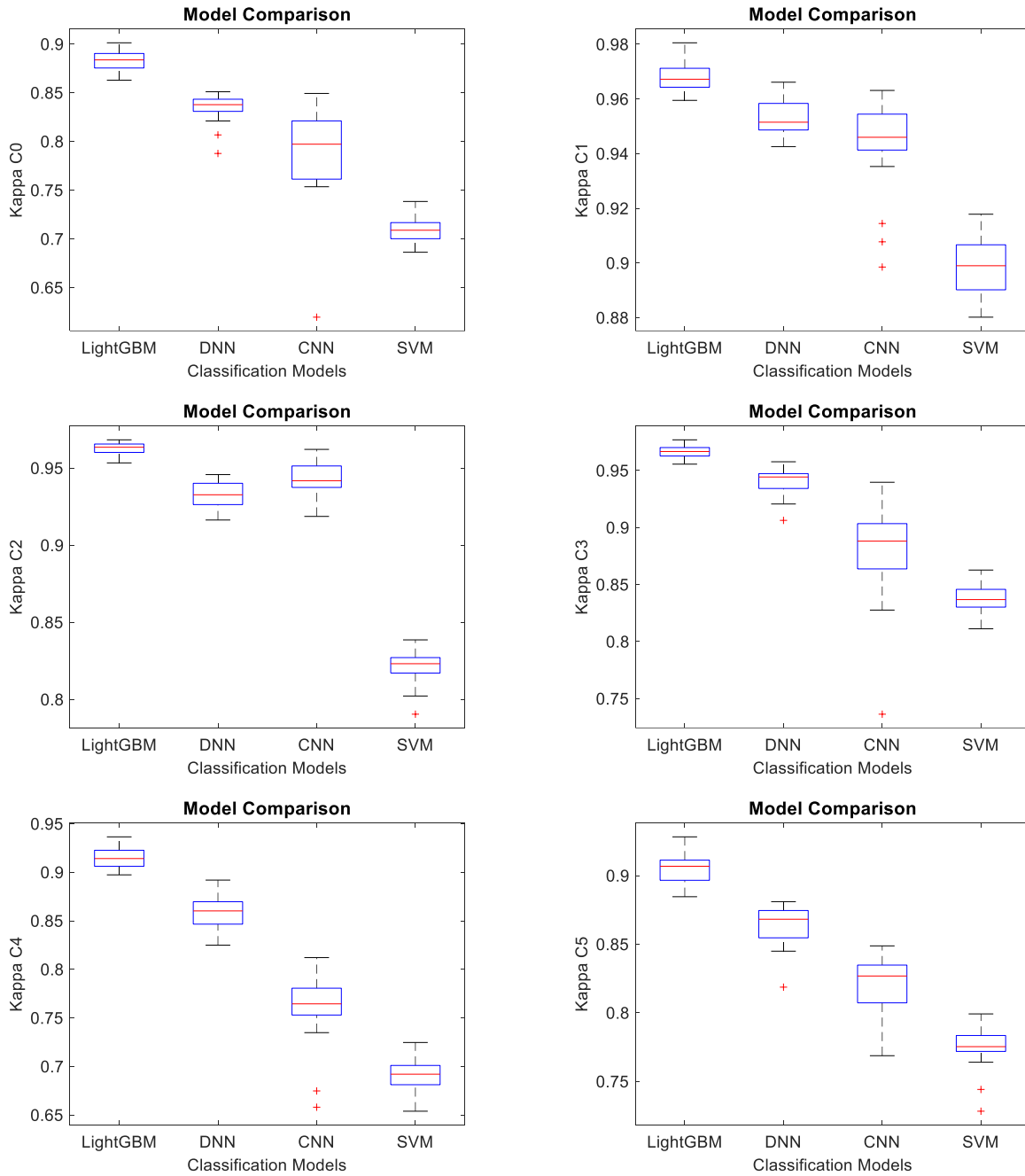


Fig. 12. Boxplots of model performance.

| | | Class | | | | | |
|-------|---|--------|--------|--------|--------|--------|--------|
| | | 0 | 1 | 2 | 3 | 4 | 5 |
| Class | 0 | 529.20 | 7.05 | 7.80 | 6.55 | 14.30 | 33.55 |
| | 1 | 4.05 | 584.90 | 1.75 | 3.40 | 4.00 | 1.40 |
| | 2 | 5.80 | 2.20 | 572.85 | 2.55 | 2.60 | 4.95 |
| | 3 | 5.75 | 3.80 | 1.85 | 591.35 | 4.90 | 0.35 |
| | 4 | 14.80 | 3.40 | 3.15 | 4.05 | 563.85 | 9.95 |
| | 5 | 15.45 | 1.25 | 4.15 | 0.55 | 24.60 | 557.90 |

Fig. 13. Average confusion matrix of the LightGBM obtained from 20 independent runs.

In addition, demonstrations of the LightGBM used for classifying data in the six classes of interest are shown in Fig. 14, Fig. 15, Fig. 16, Fig. 17, Fig. 18, and Fig. 19. As can be observed from these figures, the proposed LightGBM-based method is capable of correctly classifying image samples under various circumstances (low/excessive lighting conditions) and with the appearances of various irregular objects (e.g. traffic marks, potholes, stains, patches, raveling, shade, etc.).

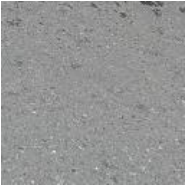





| Image sample | Actual Class | Classes' probability | Note |
|---|--------------|--|------------------------------|
|  | C0 | P(C0) = 0.9141 P(C1) = 0.0047 P(C2) = 0.0195 P(C3) = 0.0025 P(C4) = 0.0589 P(C5) = 0.0004 | Regular cases |
|  | C0 | P(C0) = 0.9952 P(C1) = 0.0000 P(C2) = 0.0022 P(C3) = 0.0000 P(C4) = 0.0006 P(C5) = 0.0020 | Traffic mark |
|  | C0 | P(C0) = 0.6649 P(C1) = 0.0140 P(C2) = 0.0110 P(C3) = 0.2092 P(C4) = 0.0879 P(C5) = 0.0129 | Blurred traffic mark |
|  | C0 | P(C0) = 0.5156 P(C1) = 0.0045 P(C2) = 0.0410 P(C3) = 0.0398 P(C4) = 0.0948 P(C5) = 0.3042 | Coupled with stain |
|  | C0 | P(C0) = 0.8466 P(C1) = 0.0000 P(C2) = 0.1266 P(C3) = 0.0003 P(C4) = 0.0024 P(C5) = 0.0240 | Irregular lighting condition |
|  | C0 | P(C0) = 0.7830 P(C1) = 0.0011 P(C2) = 0.0039 P(C3) = 0.0018 P(C4) = 0.0420 P(C5) = 0.1682 | Patches |

Fig. 14. Demonstrations of the classification for the data samples in the C0 class.







| Image sample | Actual Class | Classes' probability | Note |
|---|--------------|--|------------------------------|
|  | C1 | $P(C0) = 0.0222$ $P(C1) = 0.9699$ $P(C2) = 0.0009$ $P(C3) = 0.0064$ $P(C4) = 0.0005$ $P(C5) = 0.0001$ | Regular cases |
|  | C1 | $P(C0) = 0.0032$ $P(C1) = 0.9841$ $P(C2) = 0.0002$ $P(C3) = 0.0043$ $P(C4) = 0.0080$ $P(C5) = 0.0001$ | Parallel cracks |
|  | C1 | $P(C0) = 0.0343$ $P(C1) = 0.7664$ $P(C2) = 0.0713$ $P(C3) = 0.0814$ $P(C4) = 0.0462$ $P(C5) = 0.0005$ | Thin crack |
|  | C1 | $P(C0) = 0.0000$ $P(C1) = 0.9996$ $P(C2) = 0.0000$ $P(C3) = 0.0002$ $P(C4) = 0.0002$ $P(C5) = 0.0000$ | Excessive lighting condition |
|  | C1 | $P(C0) = 0.0001$ $P(C1) = 0.9903$ $P(C2) = 0.0000$ $P(C3) = 0.0007$ $P(C4) = 0.0088$ $P(C5) = 0.0000$ | Coupled with minor raveling |
|  | C1 | $P(C0) = 0.0019$ $P(C1) = 0.9373$ $P(C2) = 0.0017$ $P(C3) = 0.0026$ $P(C4) = 0.0492$ $P(C5) = 0.0074$ | Traffic mark |

Fig. 15. Demonstrations of the classification for the data samples in the C1 class.







| Image sample | Actual Class | Classes' probability | Note |
|---|--------------|--|------------------------------|
|  | C2 | $P(C0) = 0.0352$ $P(C1) = 0.0027$ $P(C2) = 0.9252$ $P(C3) = 0.0072$ $P(C4) = 0.0294$ $P(C5) = 0.0003$ | Regular cases |
|  | C2 | $P(C0) = 0.0007$ $P(C1) = 0.0001$ $P(C2) = 0.9715$ $P(C3) = 0.0003$ $P(C4) = 0.0094$ $P(C5) = 0.0180$ | Coupled with minor raveling |
|  | C2 | $P(C0) = 0.0089$ $P(C1) = 0.0005$ $P(C2) = 0.9483$ $P(C3) = 0.0011$ $P(C4) = 0.0278$ $P(C5) = 0.0135$ | Thin crack |
|  | C2 | $P(C0) = 0.1368$ $P(C1) = 0.0209$ $P(C2) = 0.7656$ $P(C3) = 0.0023$ $P(C4) = 0.0511$ $P(C5) = 0.0233$ | Coupled with stain |
|  | C2 | $P(C0) = 0.1300$ $P(C1) = 0.0276$ $P(C2) = 0.6712$ $P(C3) = 0.0905$ $P(C4) = 0.0589$ $P(C5) = 0.0217$ | Parallel cracks |
|  | C2 | $P(C0) = 0.0261$ $P(C1) = 0.0011$ $P(C2) = 0.9570$ $P(C3) = 0.0014$ $P(C4) = 0.0116$ $P(C5) = 0.0028$ | Irregular lighting condition |

Fig. 16. Demonstrations of the classification for the data samples in the C2 class.





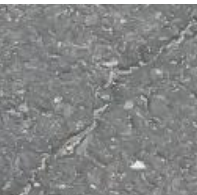

| Image sample | Actual Class | Classes' probability | Note |
|---|--------------|--|------------------------------|
|  | C3 | $P(C0) = 0.0018$ $P(C1) = 0.0002$ $P(C2) = 0.0009$ $P(C3) = 0.8944$ $P(C4) = 0.0995$ $P(C5) = 0.0032$ | Regular cases |
|  | C3 | $P(C0) = 0.0000$ $P(C1) = 0.0000$ $P(C2) = 0.0000$ $P(C3) = 0.9994$ $P(C4) = 0.0005$ $P(C5) = 0.0000$ | Coupled with traffic mark |
|  | C3 | $P(C0) = 0.0037$ $P(C1) = 0.0281$ $P(C2) = 0.0059$ $P(C3) = 0.8445$ $P(C4) = 0.1061$ $P(C5) = 0.0117$ | Excessive lighting condition |
|  | C3 | $P(C0) = 0.0000$ $P(C1) = 0.0001$ $P(C2) = 0.0000$ $P(C3) = 0.9986$ $P(C4) = 0.0001$ $P(C5) = 0.0012$ | Thin crack with patch |
|  | C3 | $P(C0) = 0.0009$ $P(C1) = 0.0436$ $P(C2) = 0.0011$ $P(C3) = 0.9417$ $P(C4) = 0.0127$ $P(C5) = 0.0000$ | Disconnected segments |
|  | C3 | $P(C0) = 0.0004$ $P(C1) = 0.0000$ $P(C2) = 0.0016$ $P(C3) = 0.9782$ $P(C4) = 0.0192$ $P(C5) = 0.0006$ | Coupled with stain |

Fig. 17. Demonstrations of the classification for the data samples in the C3 class.







| Image sample | Actual Class | Classes' probability | Note |
|---|--------------|--|------------------------------|
|  | C4 | $P(C0) = 0.0328$ $P(C1) = 0.0009$ $P(C2) = 0.0066$ $P(C3) = 0.0016$ $P(C4) = 0.9578$ $P(C5) = 0.0002$ | Regular cases |
|  | C4 | $P(C0) = 0.0049$ $P(C1) = 0.0000$ $P(C2) = 0.0415$ $P(C3) = 0.0060$ $P(C4) = 0.8895$ $P(C5) = 0.0582$ | Excessive lighting condition |
|  | C4 | $P(C0) = 0.0104$ $P(C1) = 0.0002$ $P(C2) = 0.0002$ $P(C3) = 0.0131$ $P(C4) = 0.9628$ $P(C5) = 0.0133$ | Irregular lighting condition |
|  | C4 | $P(C0) = 0.0270$ $P(C1) = 0.0001$ $P(C2) = 0.0043$ $P(C3) = 0.0003$ $P(C4) = 0.9675$ $P(C5) = 0.0008$ | Coupled with patch |
|  | C4 | $P(C0) = 0.0065$ $P(C1) = 0.0006$ $P(C2) = 0.0636$ $P(C3) = 0.0343$ $P(C4) = 0.8477$ $P(C5) = 0.0473$ | Coupled with stain |
|  | C4 | $P(C0) = 0.0384$ $P(C1) = 0.0007$ $P(C2) = 0.0012$ $P(C3) = 0.0167$ $P(C4) = 0.8751$ $P(C5) = 0.0678$ | Coupled with traffic mark |

Fig. 18. Demonstrations of the classification for the data samples in the C4 class.







| Image sample | Actual Class | Classes' probability | Note |
|---|--------------|--|--------------------------------|
|  | C5 | $P(C0) = 0.0529$ $P(C1) = 0.0000$ $P(C2) = 0.0206$ $P(C3) = 0.0012$ $P(C4) = 0.0359$ $P(C5) = 0.8894$ | Regular cases |
|  | C5 | $P(C0) = 0.0313$ $P(C1) = 0.0001$ $P(C2) = 0.0067$ $P(C3) = 0.0002$ $P(C4) = 0.0089$ $P(C5) = 0.9529$ | Coupled with pothole |
|  | C5 | $P(C0) = 0.0024$ $P(C1) = 0.0020$ $P(C2) = 0.0838$ $P(C3) = 0.0014$ $P(C4) = 0.0080$ $P(C5) = 0.9024$ | Coupled with dirt |
|  | C5 | $P(C0) = 0.1079$ $P(C1) = 0.0004$ $P(C2) = 0.0132$ $P(C3) = 0.0136$ $P(C4) = 0.3767$ $P(C5) = 0.4883$ | Coupled with minor raveling |
|  | C5 | $P(C0) = 0.0219$ $P(C1) = 0.0001$ $P(C2) = 0.1825$ $P(C3) = 0.0001$ $P(C4) = 0.0110$ $P(C5) = 0.7843$ | Coupled with pothole and stain |
|  | C5 | $P(C0) = 0.0233$ $P(C1) = 0.0000$ $P(C2) = 0.0002$ $P(C3) = 0.0001$ $P(C4) = 0.0011$ $P(C5) = 0.9752$ | Irregular lighting condition |

Fig. 19. Demonstrations of the classification for the data samples in the C5 class.

4. Concluding remarks

This paper has developed and verified the computer vision-based approaches for detecting and categorizing pavement crack patterns. The LightGBM, DNN, and CNN are employed to categorize samples of the pavement images into six categories: non-crack, longitudinal crack, transverse crack, diagonal crack, minor fatigue crack, and severe fatigue crack. In addition, image processing approaches, including SF, PI, and texture descriptors, are employed to compute the features of the pavement surface that are relevant to the categorization of the crack patterns. These features are employed by the LightGBM and the DNN to carry out the data classification phases. On the other hand, the CNN is able to perform the feature extraction and pattern recognition tasks automatically.

A dataset, consisting of 12,000 image data points, has been acquired to construct and verify the aforementioned computer vision-based approaches. Based on this image data, a set of 28 features has been computed by the image processing techniques. Accordingly, a numerical dataset has been constructed to develop the LightGBM and DNN models. By experiments, it can be shown that the LightGBM, DNN, and CNN outperform the SVM method that is widely used for crack detection and crack pattern recognition. Moreover, the Wilcoxon signed-rank test also confirms the superiority of the LightGBM over the DNN, CNN, and SVM models. Thus, the newly developed computer vision based on the LightGBM integrated with the feature extraction approach can be a promising alternative to enhance the accuracy and productivity of the pavement surveying process. Future extensions of the current work may include the following directions: (i) the applications of other advanced texture descriptors for better representing the characteristics of the pavement surface; (ii) the utilization of other potential gradient boosting machines (e.g., XGBoost [73]) for enhancing the classification accuracy; (iii) the investigation of the capability of advanced deep transfer learning in automatic feature extraction; (iv) the employment of state-of-the-art metaheuristic methods for optimizing the performance of machine learning models; (v) the applications of sophisticated image processing techniques for crack segmentation and accurate measurements of crack objects.

Supplementary material

The dataset and Python codes used to support the findings of this study have been deposited in the repository of GitHub at https://github.com/NhatDucHoang/LightGBM_PaveCrackPatterns.

Funding

This research received no external funding.

Conflicts of interest

The authors declare no conflict of interest.

Authors contribution statement

NHD: Conceptualization; Methodology; Software; Writing – original draft; Writing – review & editing. QLN: Data processing; Software; Writing – review & editing.

References

- [1] Ranjbar S, Nejad FM, Zakeri H. An image-based system for pavement crack evaluation using transfer learning and wavelet transform. *Int J Pavement Res Technol* 2021;14:437–49. <https://doi.org/10.1007/s42947-020-0098-9>.
- [2] Ali AA, Heneash U, Hussein A, Ali SIA, Khan S. Models Development for Asphalt Pavement Performance Index in Different Climate Regions Using Soft Computing Techniques. *J Soft Comput Civ Eng* 2023;7:20–42. <https://doi.org/10.22115/scce.2022.357135.1512>.
- [3] Tang Y, Zhang AA, Luo L, Wang G, Yang E. Pixel-level pavement crack segmentation with encoder-decoder network. *Measurement* 2021;184:109914. <https://doi.org/https://doi.org/10.1016/j.measurement.2021.109914>.
- [4] Kheradmandi N, Mehranfar V. A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Constr Build Mater* 2022;321:126162. <https://doi.org/https://doi.org/10.1016/j.conbuildmat.2021.126162>.
- [5] Sholevar N, Golroo A, Esfahani SR. Machine learning techniques for pavement condition evaluation. *Autom Constr* 2022;136:104190. <https://doi.org/https://doi.org/10.1016/j.autcon.2022.104190>.
- [6] Soni J, Gujar R, Shah D, Parmar P. A Review on Strategic Pavement Maintenance with Machine Learning Techniques BT - Intelligent Infrastructure in Transportation and Management. In: Shah J, Arkatkar SS, Jadhav P, editors., Singapore: Springer Singapore; 2022, p. 141–51.
- [7] Arezoumand S, Mahmoudzadeh A, Golroo A, Mojaradi B. Automatic pavement rutting measurement by fusing a high speed-shot camera and a linear laser. *Constr Build Mater* 2021;283:122668. <https://doi.org/https://doi.org/10.1016/j.conbuildmat.2021.122668>.
- [8] Dong C-Z, Catbas FN. A review of computer vision–based structural health monitoring at local and global levels. *Struct Heal Monit* 2020;20:692–743. <https://doi.org/10.1177/1475921720935585>.
- [9] Mokhtari S, Wu L, Yun H-B. Comparison of Supervised Classification Techniques for Vision-Based Pavement Crack Detection. *Transp Res Rec* 2016;2595:119–27. <https://doi.org/10.3141/2595-13>.
- [10] Cubero-Fernandez A, Rodriguez-Lozano FJ, Villatoro R, Olivares J, Palomares JM. Efficient pavement crack detection and classification. *EURASIP J Image Video Process* 2017;2017:39. <https://doi.org/10.1186/s13640-017-0187-0>.
- [11] Hoang N-D. Classification of Asphalt Pavement Cracks Using Laplacian Pyramid-Based Image Processing and a Hybrid Computational Approach. *Comput Intell Neurosci* 2018;2018:1312787. <https://doi.org/10.1155/2018/1312787>.
- [12] Hoang N-D, Nguyen Q-L. A novel method for asphalt pavement crack classification based on image processing and machine learning. *Eng Comput* 2019;35. <https://doi.org/10.1007/s00366-018-0611-9>.

- [13] Sylvester I, John S, Adrian B, Xufeng N. Pavement Crack Rating Using Machine Learning Frameworks: Partitioning, Bootstrap Forest, Boosted Trees, Naïve Bayes, and K-Nearest Neighbors. *J Transp Eng Part B Pavements* 2019;145:4019031. <https://doi.org/10.1061/JPEODX.0000126>.
- [14] Ibragimov E, Lee H-J, Lee J-J, Kim N. Automated pavement distress detection using region based convolutional neural networks. *Int J Pavement Eng* 2022;23:1981–92. <https://doi.org/10.1080/10298436.2020.1833204>.
- [15] Hsieh Y-A, Yang Z, James Tsai Y-C. Convolutional neural network for automated classification of jointed plain concrete pavement conditions. *Comput Civ Infrastruct Eng* 2021;36:1382–97. <https://doi.org/https://doi.org/10.1111/mice.12640>.
- [16] Li B, Wang KCP, Zhang A, Yang E, Wang G. Automatic classification of pavement crack using deep convolutional neural network. *Int J Pavement Eng* 2020;21:457–63. <https://doi.org/10.1080/10298436.2018.1485917>.
- [17] Nhat-Duc H, Nguyen Q-L, Tran V-D. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Autom Constr* 2018;94:203–13. <https://doi.org/https://doi.org/10.1016/j.autcon.2018.07.008>.
- [18] Zhang C, Nateghinia E, Miranda-Moreno LF, Sun L. Pavement distress detection using convolutional neural network (CNN): A case study in Montreal, Canada. *Int J Transp Sci Technol* 2022;11:298–309. <https://doi.org/https://doi.org/10.1016/j.ijtst.2021.04.008>.
- [19] Liu F, Liu J, Wang L. Asphalt Pavement Crack Detection Based on Convolutional Neural Network and Infrared Thermography. *IEEE Trans Intell Transp Syst* 2022;23:22145–55. <https://doi.org/10.1109/TITS.2022.3142393>.
- [20] Cano-Ortiz S, Pascual-Muñoz P, Castro-Fresno D. Machine learning algorithms for monitoring pavement performance. *Autom Constr* 2022;139:104309. <https://doi.org/https://doi.org/10.1016/j.autcon.2022.104309>.
- [21] Arya D, Maeda H, Ghosh SK, Toshniwal D, Mraz A, Kashiya T, et al. Deep learning-based road damage detection and classification for multiple countries. *Autom Constr* 2021;132:103935. <https://doi.org/https://doi.org/10.1016/j.autcon.2021.103935>.
- [22] Du Z, Yuan J, Xiao F, Hettiarachchi C. Application of image technology on pavement distress detection: A review. *Measurement* 2021;184:109900. <https://doi.org/https://doi.org/10.1016/j.measurement.2021.109900>.
- [23] Ke G., Meng Q., Finley T., Wang T., Chen W., Ma W., Ye Q. LT-Y. LightGBM: a highly efficient gradient boosting decision tree. *31st Int. Conf. Neural Inf. Process. Syst., Curran Associates Inc. United States; 2017, p. 3149–3157*.
- [24] Barua L, Zou B, Noruzoliaee M, Derrible S. A gradient boosting approach to understanding airport runway and taxiway pavement deterioration. *Int J Pavement Eng* 2021;22:1673–87. <https://doi.org/10.1080/10298436.2020.1714616>.
- [25] Chun P, Izumi S, Yamane T. Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. *Comput Civ Infrastruct Eng* 2021;36:61–72. <https://doi.org/10.1111/mice.12564>.
- [26] Asghari V, Leung YF, Hsu S-C. Deep neural network based framework for complex correlations in engineering metrics. *Adv Eng Informatics* 2020;44:101058. <https://doi.org/https://doi.org/10.1016/j.aei.2020.101058>.

- [27] Goodfellow I, Bengio Y. CA. Deep Learning (Adaptive Computation and Machine Learning series). The MIT Press; 2016.
- [28] Le H Van, Hoang DA, Tran CT, Nguyen PQ, Tran VHT, Hoang ND, et al. A new approach of deep neural computing for spatial prediction of wildfire danger at tropical climate areas. *Ecol Inform* 2021;63:101300. <https://doi.org/https://doi.org/10.1016/j.ecoinf.2021.101300>.
- [29] Hadjidemetriou MG, Vela AP, Christodoulou ES. Automated Pavement Patch Detection and Quantification Using Support Vector Machines. *J Comput Civ Eng* 2018;32:4017073. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000724](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000724).
- [30] Shaofan W, Shi Q, Wenjuan W, Danny X, P. WKC. Cracking Classification Using Minimum Rectangular Cover-Based Support Vector Machine. *J Comput Civ Eng* 2017;31:4017027. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000672](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000672).
- [31] Sari Y, Prakoso PB, Baskara AR. Road Crack Detection using Support Vector Machine (SVM) and OTSU Algorithm. 2019 6th Int. Conf. Electr. Veh. Technol., 2019, p. 349–54. <https://doi.org/10.1109/ICEVT48285.2019.8993969>.
- [32] Perona P. Deformable kernels for early vision. *IEEE Trans Pattern Anal Mach Intell* 1995;17:488–99. <https://doi.org/10.1109/34.391394>.
- [33] Freeman WT, Adelson EH. The design and use of steerable filters. *IEEE Trans Pattern Anal Mach Intell* 1991;13:891–906. <https://doi.org/10.1109/34.93808>.
- [34] Hernandez-Matamoros A, Bonarini A, Escamilla-Hernandez E, Nakano-Miyatake M, Perez-Meana H. Facial expression recognition with automatic segmentation of face regions using a fuzzy based classification approach. *Knowledge-Based Syst* 2016;110:1–14. <https://doi.org/https://doi.org/10.1016/j.knosys.2016.07.011>.
- [35] Hoang N-D, Nguyen Q-L. Automatic Recognition of Asphalt Pavement Cracks Based on Image Processing and Machine Learning Approaches: A Comparative Study on Classifier Performance. *Math Probl Eng* 2018;2018:6290498. <https://doi.org/10.1155/2018/6290498>.
- [36] Kayhan N, Fekri-Ershad S. Content based image retrieval based on weighted fusion of texture and color features derived from modified local binary patterns and local neighborhood difference patterns. *Multimed Tools Appl* 2021;80:32763–90. <https://doi.org/10.1007/s11042-021-11217-z>.
- [37] Theodoridis S, Koutroumbas K. *Pattern Recognition*. Academic Press; 2009.
- [38] Haralick RM. Statistical and structural approaches to texture. *Proc IEEE* 1979;67:786–804. <https://doi.org/10.1109/PROC.1979.11328>.
- [39] Sonka M., Hlavac V. BR. *Image processing, Analysis, and Machine Vision*. Cengage Learning, Printed in the United States of America; 2014.
- [40] Bhagat PK, Choudhary P, Singh KM. Chapter 13 - A comparative study for brain tumor detection in MRI images using texture features. In: Dey N, Chaki J, Kumar RBT-S for HM, editors. *Adv. ubiquitous Sens. Appl. Healthc.*, vol. 5, Academic Press; 2019, p. 259–87. <https://doi.org/https://doi.org/10.1016/B978-0-12-819361-7.00013-0>.
- [41] Zubair AR, Alo OA. Grey level co-occurrence matrix (GLCM) based second order statistics for image texture analysis. *Int J Comput Appl* 2019;12–20. <https://doi.org/https://doi.org/10.5120/ijca2020920475>.
- [42] Liang W, Luo S, Zhao G, Wu H. Predicting Hard Rock Pillar Stability Using GBDT, XGBoost, and LightGBM Algorithms. *Mathematics* 2020;8. <https://doi.org/10.3390/math8050765>.

- [43] Alzamzami F, Hoda M, Saddik AE. Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation. *IEEE Access* 2020;8:101840–58. <https://doi.org/10.1109/ACCESS.2020.2997330>.
- [44] Perner P. Decision Tree Induction Methods and Their Application to Big Data BT - Modeling and Processing for Next-Generation Big-Data Technologies: With Applications and Case Studies. In: Xhafa F, Barolli L, Barolli A, Papajorgji P, editors., Cham: Springer International Publishing; 2015, p. 57–88. https://doi.org/10.1007/978-3-319-09177-8_3.
- [45] Hagan MT, Demuth HB, M.H. B, Esús OD. *Neural Network Design* (2nd Edition). Martin Hagan; 2014.
- [46] Haykin SO. *Neural Networks and Learning Machines*. Pearson; 2008.
- [47] Czarnecki S, Shariq M, Nikoo M, Sadowski Ł. An intelligent model for the prediction of the compressive strength of cementitious composites with ground granulated blast furnace slag based on ultrasonic pulse velocity measurements. *Measurement* 2021;172:108951. <https://doi.org/https://doi.org/10.1016/j.measurement.2020.108951>.
- [48] Kingma DP, Ba JA. A Method for Stochastic Optimization. *ArXiv:14126980 [CsLG]* 2015. <https://doi.org/https://doi.org/10.48550/arXiv.1412.6980>.
- [49] LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. *Handb. brain theory neural Netw.*, MIT Press Cambridge, MA, USA; 1998, p. 255–8.
- [50] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521:436–44. <https://doi.org/10.1038/nature14539>.
- [51] Kim P. *MatLab Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence*. Apress; 2017.
- [52] Padsumbiya M, Brahmabhatt V, Thakkar SP. Automatic Crack Detection Using Convolutional Neural Network. *J Soft Comput Civ Eng* 2022;6:1–17. <https://doi.org/10.22115/scce.2022.325596.1397>.
- [53] Dhillon A, Verma GK. Convolutional neural network: a review of models, methodologies and applications to object detection. *Prog Artif Intell* 2020;9:85–112. <https://doi.org/10.1007/s13748-019-00203-0>.
- [54] Calin O. *Deep Learning Architectures: A Mathematical Approach*. Cham: Springer International Publishing; 2020.
- [55] Davies ER. Chapter 1 - The dramatically changing face of computer vision. In: Davies ER, Turk MABT-AM and DL in CV, editors. *Comput. Vis. Pattern Recognit.*, Academic Press; 2022, p. 1–91. <https://doi.org/https://doi.org/10.1016/B978-0-12-822109-9.00010-2>.
- [56] Microsoft. LightGBM's documentation. <<https://LightgbmReadthedocsIo/En/v332/>> (Last Access Date 7/14/2022) 2022.
- [57] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. *J Mach Learn Res* 2012;12.
- [58] Beale MH, Hagan MT, Demuth HB. *Deep Learning Toolbox User's Guide*. Mathworks <https://WwwMathworksCom/Help/Pdf_doc/Deeplearning/Nnet_ugPdf > (Last Access Date 05/11/2019) 2018.
- [59] Sure-Seal-Pavement. *Different Types of Asphalt Pavement Cracks*. Pavement Maint Serv Ontario, Sure-Seal Pavement Maint Inc 2023.

- [60] Nhat-Duc H, Quoc-Lam N, Dieu TB. Image Processing–Based Classification of Asphalt Pavement Cracks Using Support Vector Machine Optimized by Artificial Bee Colony. *J Comput Civ Eng* 2018;32:4018037. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000781](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000781).
- [61] van der Walt S, Schönberger J, Nunez-Iglesias J, Boulogne F, Warner J, Yager N, et al. scikit-image: Image processing in Python. *PeerJ* 2014;2. <https://doi.org/10.7717/peerj.453>.
- [62] Robnik-Šikonja M, Kononenko I. Theoretical and Empirical Analysis of ReliefF and RReliefF. *Mach Learn* 2003;53:23–69. <https://doi.org/10.1023/A:1025667309714>.
- [63] Kononenko I. Estimating attributes: Analysis and extensions of RELIEF BT - Machine Learning: ECML-94. In: Bergadano F, De Raedt L, editors., Berlin, Heidelberg: Springer Berlin Heidelberg; 1994, p. 171–82.
- [64] Urbanowicz RJ, Meeker M, La Cava W, Olson RS, Moore JH. Relief-based feature selection: Introduction and review. *J Biomed Inform* 2018;85:189–203. <https://doi.org/https://doi.org/10.1016/j.jbi.2018.07.014>.
- [65] Fawcett T. An introduction to ROC analysis. *Pattern Recognit Lett* 2006;27:861–74. <https://doi.org/10.1016/j.patrec.2005.10.010>.
- [66] van Erkel AR, Pattynama PMT. Receiver operating characteristic (ROC) analysis: Basic principles and applications in radiology. *Eur J Radiol* 1998;27:88–94. [https://doi.org/https://doi.org/10.1016/S0720-048X\(97\)00157-5](https://doi.org/https://doi.org/10.1016/S0720-048X(97)00157-5).
- [67] Tien Bui D, Hoang N-D, Martínez-Álvarez F, Ngo P-TT, Hoa PV, Pham TD, et al. A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Sci Total Environ* 2020;701. <https://doi.org/10.1016/j.scitotenv.2019.134413>.
- [68] López V, Fernández A, García S, Palade V, Herrera F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf Sci (Ny)* 2013;250:113–41. <https://doi.org/https://doi.org/10.1016/j.ins.2013.07.007>.
- [69] Arif T. *Introduction to Deep Learning for Engineers - Using Python and Google Cloud Platform*. Morgan & Claypool; 2020.
- [70] Chollet F. *Deep Learning with Python*. Manning Publications; 2015.
- [71] McHugh M. Interrater reliability: The kappa statistic. *Biochem Med (Zagreb)* 2012;22:276–82. <https://doi.org/10.11613/BM.2012.031>.
- [72] Hollander M, Wolfe DA. *Nonparametric Statistical Methods*. John Wiley & Sons; 1999.
- [73] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., New York, NY, USA: ACM; 2016, p. 785–94*. <https://doi.org/10.1145/2939672.2939785>.