Primljen / Received: 22.1.2024.
Ispravljen / Corrected: 24.7.2024.
Prihvaćen / Accepted: 2.9.2024.
Dostupno online / Available online: 10.12.2024.

Pavement condition detection using acceleration data collected by smartphones based on 1D convolutional neural network

Authors:

Yudong Han, MSc. CE
University of Science and Technology Liaoning, China
Schoo of Civil Engineering
hanyudong@ustl.edu.cn

Zhaobo Li, MCE
Center for Innovative Services in Science and Technology Hohhot, China
China University of Mining and Technology
Schoo of Mechanical and Civil Engineering
lb23030001@cumt.edu.cn

Jiaqi Li, PhD. CE
Liaoning University of Science and Technology, China
Schoo of Civil Engineering
lijiaqi@ustl.edu.cn
Corresponding author

Research Paper

Yudong Han, Zhaobo Li, Jiaqi Li

Pavement condition detection using acceleration data collected by smartphones based on 1D convolutional neural network

Vibration-based pavement condition detection methods have advanced in recent years, and it has been proven to be feasible to identify pavement conditions by analysing acceleration data. In this study, a public participation solution is proposed, and a one-dimensional convolutional neural network (1D-CNN) is introduced to directly process acceleration signals, addressing the limitations of traditional machine-learning classification methods. In this study, a smartphone and bicycle were used as the experimental tools, and 422 samples of acceleration data across the X-, Y-, and Z-axes were collected, including four types of pavement conditions: bumpy pavement, speed bumps, smooth pavement, and potholes. Five types of 1D-CNN with different activation functions and network structures were designed to classify the data and were compared with machine learning algorithms, including support vector machine (SVM) and radial basis function (RBF) neural networks. The results show that a 1D-CNN, with three convolution layers and three pooling layers using the ReLU activation function, achieved the best classification performance, with a classification accuracy of 0.9976. Compared with SVM and RBF neural networks, CNN not only saves considerable time by eliminating manual feature extraction operations but also provides higher classification accuracy.

Key words:
pavement detection, convolutional neural network, deep learning, pavement, smartphone

Prethodno priopćenje

Yudong Han, Zhaobo Li, Jiaqi Li

Utvrđivanje stanja kolnika pomoću podataka o ubrzanju prikupljenih pametnim telefonima na temelju 1D konvolucijske neuronske mreže

Posljednjih je godina postignut napredak metoda utvrđivanja stanja kolnika na temelju vibracija i dokazano je da je moguće utvrditi stanje kolnika analizom podataka o ubrzanju. U ovom se radu predlaže sudjelovanje javnosti u prikupljanju podataka i uvedena je jednodimenzijska konvolucijska neuronska mreža (1D-CNN) za izravnu obradu signala ubrzanja kako bi se otklonila ograničenja tradicionalnih metoda kategoriziranja strojnim učenjem. U ovom istraživanju kao alati za ispitivanje upotrijebljeni su pametni telefon i bicikl, a prikupljena su 422 uzorka podataka o ubrzanju duž X, Y i Z osi, uključujući četiri tipa stanja kolnika: neravan kolnik, kolnik s uspornicima (tzv. ležećim policajcima), ravni kolnik i kolnik s udarnim rupama. Projektirano je pet tipova mreže 1D-CNN-a s različitim funkcijama za aktivaciju i mrežnim strukturama za klasifikaciju podataka i uspoređeno je s algoritmima za strojno učenje, uključujući neuronske mreže stroja potpornih vektora (SVM) i radijalne bazne funkcije (RBF). Rezultati pokazuju da je jednodimenzijska konvolucijska neuronska mreža, uz tri konvolucijska sloja i tri sloja udruživanja primjenom rektificirane linearne aktivacije najučinkovitija u pogledu klasifikacije, s klasifikacijskom točnošću od 0,9976. U usporedbi s neuronskim mrežama stroja potpornih vektora (SVM) i radijalne bazne funkcije (RBF), konvolucijska neuronska mreža ne samo da štedi mnogo vremena jer ne zahtijeva ručno izdvajanje značajki, već pruža i veću klasifikacijsku točnost.

Ključne riječi:
utvrđivanje stanja kolnika, konvolucijska neuronska mreža, duboko učenje, kolnik, pametni telefon

1. Introduction

Pavement condition detection is crucial for the maintenance and management of road infrastructure. The accurate and timely evaluation of pavement conditions helps identify areas that require maintenance, thereby preventing minor issues from developing into significant structural problems. This proactive approach extends the lifespan of roadways and ensures the safety and comfort of users by reducing the risk of accidents caused by poor road conditions. Furthermore, effective pavement condition assessment can lead to significant cost savings for local authorities and transportation agencies by optimising maintenance schedules and resource allocation. As urbanisation and traffic volumes increase, the demand for robust and scalable pavement-monitoring solutions becomes even more critical in supporting sustainable infrastructure development [1-4].

As attention to the management and maintenance of road infrastructure increases and the diversity of data collection devices grows, health monitoring has been widely applied in the field of pavements. Currently, three methods are widely used: 3D reconstruction, vision-based detection, and vibration-based detection.

Three-dimensional (3D) reconstruction methods can be further classified into 3D laser scanning methods, stereo methods, and visualisation using a true depth camera. Chang et al. [5] and Li et al. [6] used 3D laser scanning technology to detect potholes in real-time. Wang [7] and Hou et al. [8] used stereovision methods to detect road potholes, achieving good results. Joubert et al. [9] and Mozzam et al. [10] used Microsoft's Kinect camera to obtain three-dimensional information about potholes. However, 3D laser scanning technology and stereo vision methods have not been widely applied due to the relatively high cost of equipment and the complexity of calculations and processing in the later stages.

Vision-based detection methods are intuitive and have been widely used in civil engineering inspections in recent years. Koch and Brilakis [11] presented a method for automated pothole detection in asphalt pavement images based on a histogram threshold successfully applied using MATLAB software. Hoang [12] used SVM and ANN to build a pothole detection model; 89 % classification accuracy was achieved using SVM. Eisenbach et al. [13] used a mobile mapping system, S.T.I.E.R., to detect road damage, such as cracks, potholes, and patches, with an accuracy exceeding 90 %. Li et al. [14] classified pavement cracks into four categories (longitudinal, transverse, block, and alligator cracks); four CNNs with different structures were applied, and the classification precision exceeded 0.9. In recent years, with the development of object detection technology, several approaches have been proposed for pavement condition monitoring and damage detection, achieving satisfactory accuracy [15, 18]. Vision-based methods offer advantages in rapid identification and damage localisation. However, they are not well integrated with the mechanical response of the pavement to provide the initial indication of the health of a particular roadway section.

A considerable amount of information can be obtained from acceleration signals, such as in human motion recognition [19, 22], structural health monitoring [23, 25], and seismic analysis [26, 28]. Yu and Yu [29] proposed that road conditions can be assessed using acceleration signals obtained during driving. Erikson et al. [30] described a pothole patrol system that uses a machine learning algorithm to detect pavement potholes. Mednis et al. [31] developed a method for monitoring potholes using acceleration sensors inside smartphones, with a real positive rate of 90 %. Fox et al. [32] collected acceleration data during driving and proposed a machine-learning-based pothole detection method that achieved an experimental accuracy of 0.889. Bhatt et al. [33] compared the classification performance of gradient boosting and SVM for acceleration signals, and the results showed that the accuracy of these two classifiers for identifying pavement potholes was 92.02 % and 92.9 %, respectively. Du et al. [34] distinguished abnormal pavement types using acceleration data via the k-nearest neighbour (KNN) algorithm; the accuracy exceeded 0.9. Yang and Zhou [35] analysed the acceleration data in the time and frequency domains and detected transverse cracks. Egaji et al. [36] proposed a machine-learning model for pothole detection using acceleration data, with random forest and KNN showing the best performance. Zhang et al. [37] indicated that road surface roughness could be evaluated using an acceleration sensor built into a smartphone, and the experimental results showed that the error rate was less than 10 %.

Among the studies mentioned above, most were threshold- or machine-learning-based approaches. The former is designed to set a threshold for acceleration; as soon as the acceleration value exceeds this threshold, the pavement is considered to have a pothole. The accuracy of this method cannot be guaranteed, and it cannot distinguish other situations that lead to sudden changes in acceleration (e.g. driving over speed bumps). Machine learning-based methods for processing data are generally divided into two steps: feature extraction and classification. Although there is faster speed and higher accuracy in the second stage, the feature extraction operation adds complexity and reduces the efficiency of the entire process. In terms of detection equipment, intelligent, economical, and portable technologies are being developed. Currently, smartphones can compete in some aspects of structural health monitoring and enable public participation [38, 39].

To summarise the previous discussion, we noted that acceleration signals respond better to the mechanical properties of pavements. Smartphones are widely available portable devices that enable public participation and large-scale data collection and analysis. Therefore, this study proposes a public-participation-based pavement acceleration data classification

method for pavement condition detection that better serves the management and maintenance of transportation infrastructure. In terms of devices, the widespread availability of bicycles and smartphones provides favourable conditions for public participation. In terms of data recognition, we propose the use of convolutional neural networks (CNNs) for automatic feature extraction and classification without the need for manually designing and extracting features, as required in traditional machine learning methods. This approach has not been proposed in the past and is the focus of this study.

2. Methodology

To complete this study, we first used bicycles, which are still widely used, as carriers and smartphones fixed at the front as acceleration acquisition devices. Next, cycling through smooth pavement, bumpy pavement, potholes, and speed bumps at a uniform speed and recording the X-, Y-, and Z-axes acceleration data in three directions using an application called Orion CC. Finally, we used the CNN, SVM, and RBF neural networks to classify the four different acceleration signals and compared the classification performances of the three algorithms.

2.1. Dataset

As shown in Figure 1, the device used in this study was an iPhone 8 Plus, which was fixed at the front of the bicycle to collect acceleration data. To reduce the workload of CNN training, we analysed the acceleration of the Z-axis alone so that the Z-axis was perpendicular to the ground direction when the mobile phone was fixed. However, due to the influences of turning, tilting, and vibration, the Z-axis was not always perpendicular to the ground during actual data acquisition. Therefore, the acceleration data in the X-, Y-, and Z-axes were still included in the dataset.

The purpose of this study is to identify pavement conditions. Not only will the potholes cause acceleration changes, but the speed bump and bumpy pavement will cause significant changes in the acceleration. Therefore, the final acceleration dataset included smooth pavement, potholes, uneven pavement, and speed bumps (Figure 2). When collecting the acceleration data, the speed was set at 10 to 15 m/s, the sampling frequency was 100 Hz, and the sampling time was 5 s. Each time-series signal contained 500 elements. Each contained three directions: X, Y and Z. 422 groups of data were collected: 108 groups for bumpy pavements, 99 groups for speed bumps, 113 groups for smooth pavements, and 102 groups for potholes. Figure 3 shows the original acceleration time-domain data collected by the smartphone and the acceleration frequency-domain data obtained using the fast Fourier transform.

Figure 1. Experimental devices and Orion CC

2.2. Convolutional Neural Network

The most significant difference between a CNN (*Convolutional Neural Network*) and a traditional artificial neural network is that the addition of convolution and pooling layers. As shown in Figure 4, the ANN (*Artificial Neural Network*) includes an input layer, hidden layer and output layer. Neurons are the most basic structure of ANNs. Each neuron is considered a node. Neighbouring layers of neurons connect, whereas there is no connection between neurons within the same layer.

Figure 2. Examples of Bumpy pavement, smooth pavement, potholes, and speed bump

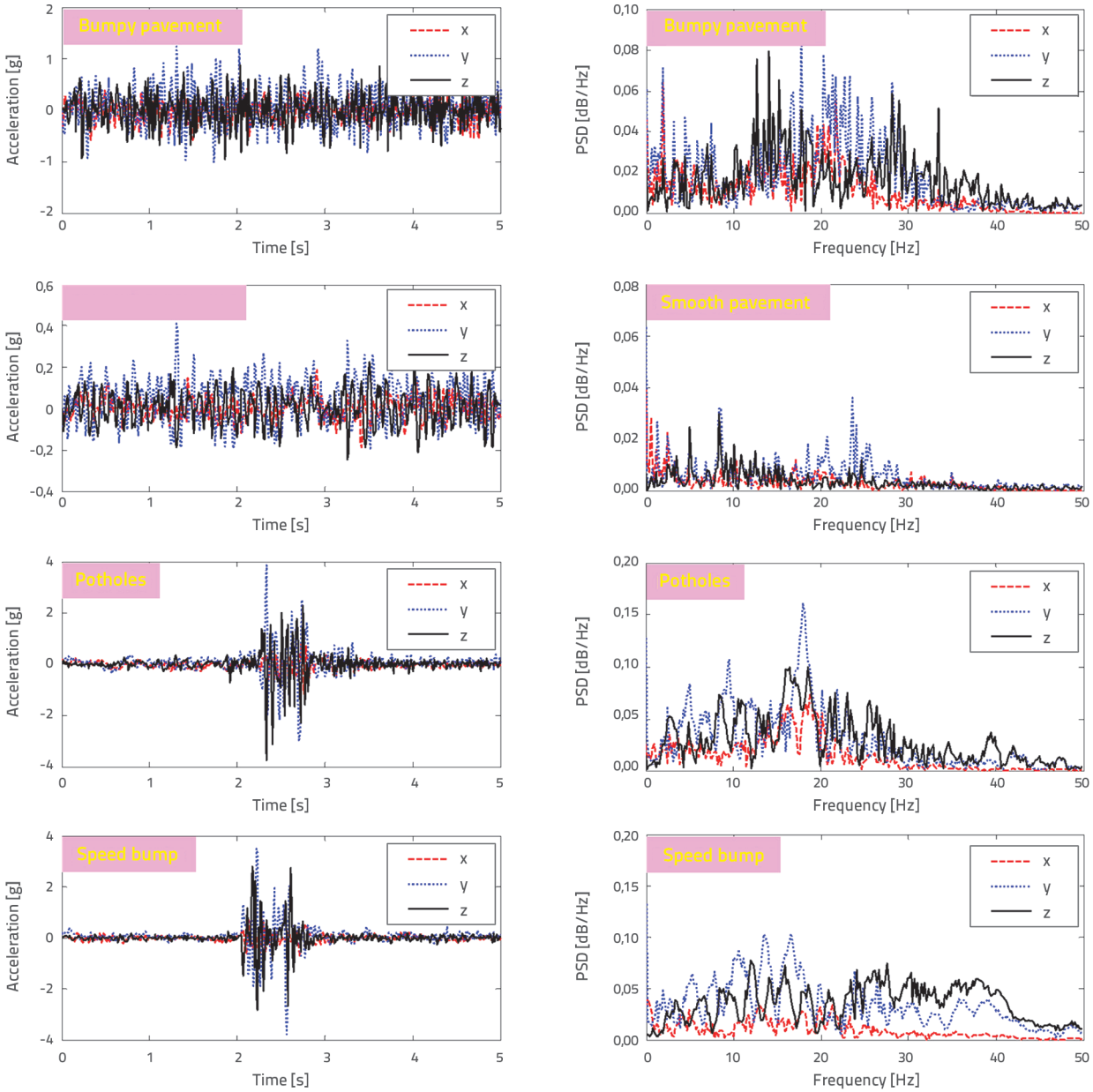

Figure 3. Examples of Acceleration data in Time domain and Frequency domain (x, y, z axes)

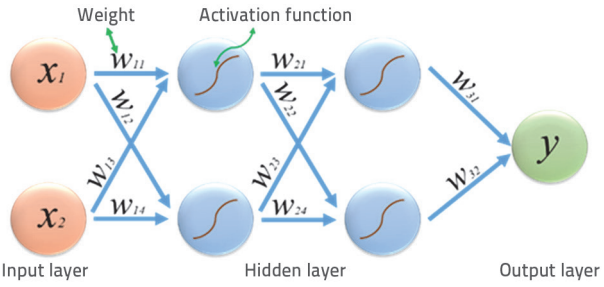

Figure 4. Examples of Acceleration data in Time domain and Frequency domain

ANNs process complicated problems slowly. For example, when processing a 28 × 28 black-and-white image in the MNIST dataset, if the number of nodes in the first hidden layer is 500, then 392,500 parameters are required in the entire connection layer. A black-and-white image can be regarded as a two-dimensional matrix composed of data. When the number of nodes is excessively large, the training speed of the ANN decreases. If overfitting occurs, the accuracy will also be affected.

CNN was first proposed by Lecun et al. [40], achieving 99.7 % accuracy on the MNIST dataset. CNN increases matrix depth through convolution and pooling layers, reduces dimensionality

Table 1. Details of CNNs

	Layers	Input shape	Kernel size	Kernel number	Strides	Padding	Activation	Output shape
CNN0 /CNN1 /CNN2	Conv1	(500,3)	10	16	1	Same	ReLu/tanh/Sigm	(500,16)
	Pooling1	(500,16)	4	None	4	Valid	None	(125,16)
	Conv2	(125,16)	10	32	1	Same	ReLu/tanh/Sigm	(125,32)
	Pooling2	(125,32)	4	None	4	Same	None	(32,32)
	Conv3	(32,32)	10	64	1	Same	ReLu/tanh/Sigm	(32,64)
	Pooling3	(32,64)	4	None	4	Valid	None	(8,64)
	Flatten	(8,64)	None	None	None	None	None	(512)
	Dense1	(512)	None	None	None	None	ReLu/tanh/Sigm	(128)
	Dense2	(128)	None	None	None	None	Softmax	(4)
CNN3	Conv1	(500,3)	10	16	1	Same	ReLu	(500,16)
	Pooling1	(500,16)	4	None	4	Valid	None	(125,16)
	Conv2	(125,16)	10	32	1	Same	ReLu	(125,32)
	Pooling2	(125,32)	4	None	4	Same	None	(32,32)
	Flatten	(32,32)	None	None	None	None	None	(1024)
	Dense1	(1024)	None	None	None	None	ReLu	(128)
	Dense2	(128)	None	None	None	None	Softmax	(4)
CNN4	Conv1	(500,3)	10	16	1	Same	ReLu	(500,16)
	Pooling1	(500,16)	4	None	4	Valid	None	(125,16)
	Conv2	(125,16)	10	32	1	Same	ReLu	(125,32)
	Pooling2	(125,32)	4	None	4	Same	None	(32,32)
	Conv3	(32,32)	10	64	1	Same	ReLu	(32,64)
	Pooling3	(32,64)	4	None	4	Valid	None	(8,64)
	Conv4	(8,64)	4	128	1	Valid	ReLu	(5,128)
	Pooling4	(5.128)	2	None	2	Valid	None	(2.128)
	Flatten	(2.128)	None	None	None	None	None	(256)
	Dense1	(256)	None	None	None	None	ReLu	(128)
	Dense2	(128)	None	None	None	None	Softmax	(4)

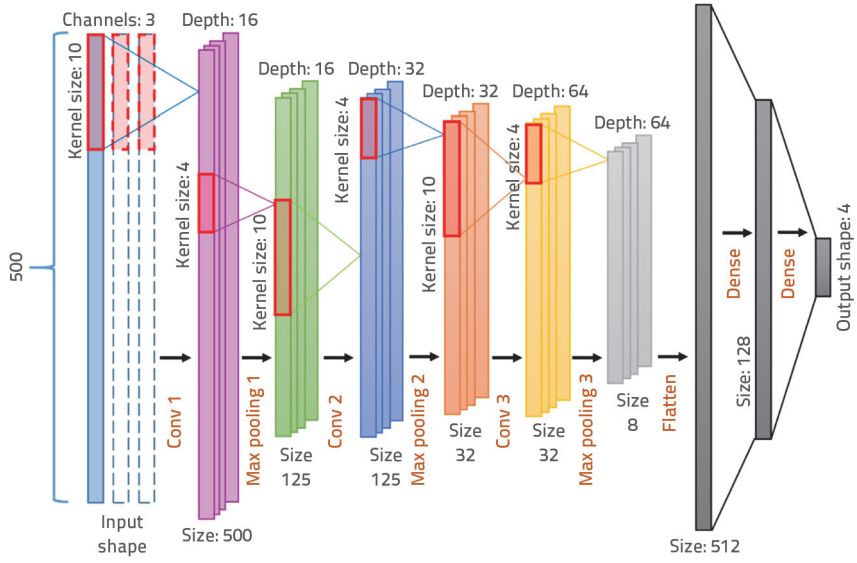

Figure 5. Examples of Acceleration data in Time doma (TD) in and Frequency domain (FD)

and the number of parameters used for operation, and is more sensitive to complex data and matrices, which can improve accuracy. The time-acceleration data used in this study can be regarded as a one-dimensional matrix. As shown in Table 1, five CNNs were designed to address this issue. CNN0, CNN1, and CNN2 had the same network structure but different activation functions. CNN3 and CNN4 reduced and increased the number of convolution and pooling layers, respectively. Figure 5 illustrates the general working process using CNN0 as an example; there are 422 groups of input data, each of which contains 500 elements, including three channels of the X-, Y-, and Z-axes. Through three convolution layers

and three pooling layers, a matrix of size 8 × 1 × 64 is output, with 64 representing the depth. It is then transformed into a vector of 512 elements through the flattening layer. Finally, classification is performed through the final two dense layers.

2.2.1. Convolution layer

The convolution layer was used to expand the dimensions of the matrix. The convolution layer consisted of a convolution kernel. The kernel depth must be equal to that of the input matrix. Taking the Conv2 layer as an example, the input size was 125, the depth was 16, the convolution kernel size and depth were 10 and 16, respectively, and the number of kernels was 32. Once convoluted, the kernel moves one step from top to bottom and convolves with the input matrix to generate a new matrix with a depth of 32. If the padding is set to "same", the output vector length remains the same as the input. If the padding is "valid", the output length is 125-10+1 = 116. Figure 6(a) illustrates the principle of convolution layers with a simple column vector.

2.2.2. Pooling layer

The primary function of the pooling layer is to reduce the dimensions of the input matrix and extract features. As mentioned previously, after the Conv2 operation is completed, the matrix is input into the pooling layer. The size is 125; the kernel size and strides are all 4, and from top to bottom, a maximum value is selected for every four elements. If the padding is the "same", 125 is padded to 128, making it divisible by 4, and the output size is 32. If the padding is "valid", the integer part after division is taken, resulting in 31. Figure 6.b shows the principle of the pooling layer with the output shown in Figure 6.a.

2.2.3. Dense layer

The dense layer functions as a "classifier" in the entire CNN. If the operations of the convolution layer, pooling layer and activation function map the original data to the hidden layer feature space, the dense layer maps the learned "distributed feature representation" to the sample label space. In Figure 6, after Pooling3 and flattening, the input data are expanded into a vector of 1 × 512, 512 nodes are processed by Dense1 to generate a vector of 1 × 128, and the final 1 × 4 vector is obtained by the softmax function in Dense2, corresponding to the four pavement condition categories.

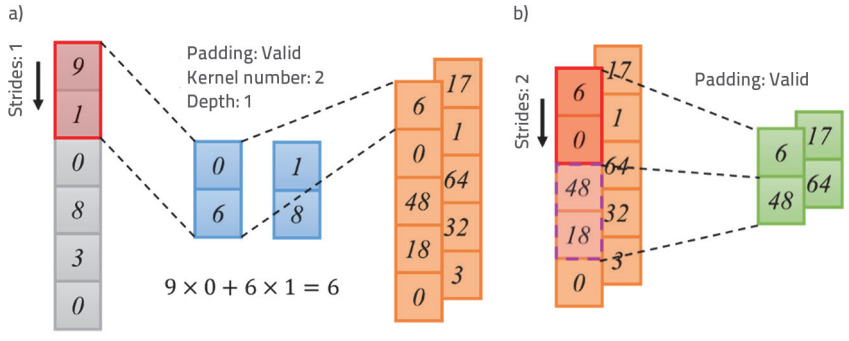

Figure 6. Examples of Convolution layer and pooling layer: a) Convolution layer; b) Pooling layer

2.2.4. Activation functions

An activation function is used in the convolutional and dense layers. For the layer described in Figure 6, the activation function is not represented when convolution simplicity and intuitiveness are prioritised. The activation function activates a portion of the neurons in the network and transmits the activation information to the next layer of the neural network. Using the neural network shown in Figure 5 as an example, if the activation function is not added, the output of the first node of the first hidden layer will be $y = x_1w_{11}+x_2w_{13}$; however, this classification method cannot solve the problem of linear inseparability, and the activation function introduces nonlinear factors to address the inadequate expression of inadequate expression of the linear model. Using Figure 5 as an example and assuming the activation function is $f(x)$, the output is $y = f(x_1w_{11} + x_2w_{13})$. As shown in Figure 7, the standard nonlinear activation functions are sigmoid, tanh, and ReLU. ReLU is the most widely used.

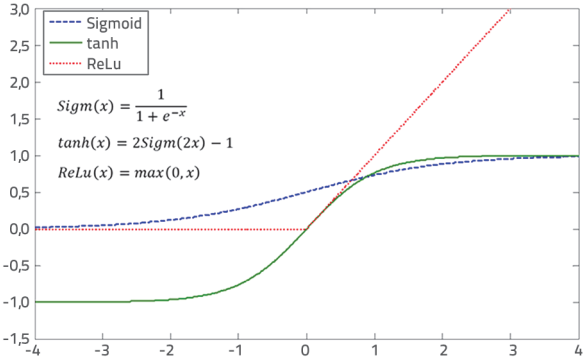

Figure 7. Examples of Acceleration data in Time domain and Frequency domain

2.3. Support vector machine (SVM)

The SVM was first proposed by Cortes and Vapnik in 1995 [41]. When addressing classification problems, the primary objective is to find a hyperplane that separates the training sample points while minimising classification error. In cases of linear separability, one or more hyperplanes can fully separate the training samples. The goal of SVM is to find the optimal solution.

Figure 8 shows a classification problem for two-dimensional data in which the hyperplane is a straight line. When dealing with three-dimensional spatial data, a hyperplane becomes

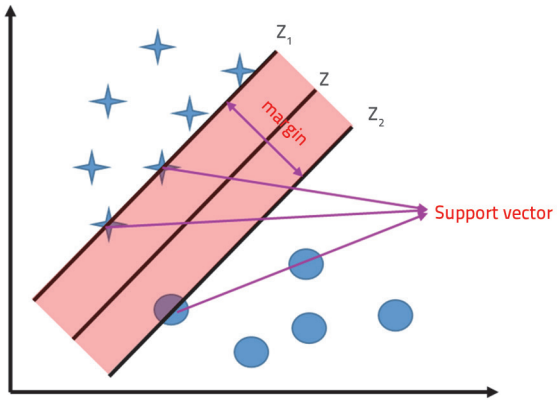

Figure 8. Principle of SVM

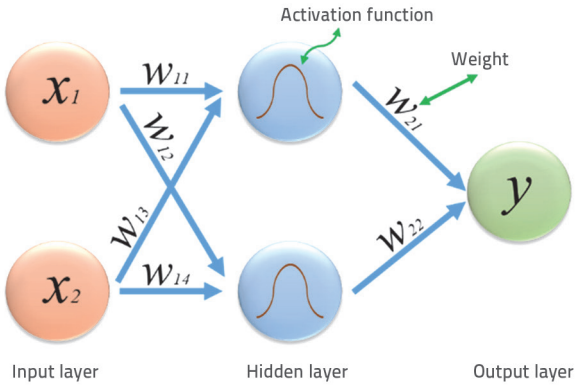

Figure 9. Principle of RBF neural network

a 2D plane. The hyperplane can be described with the following formula.

$\omega^t x + b = 0$

gdje *x* predstavlja dvodimenzijske koordinate točaka, a Z where *x* represents the two-dimensional coordinates of the points, and Z represents the hyperplane. Z1 and Z2 are the nearest parallel lines to Z for all samples. The distance between Z1 and Z2 is the classification margin between the two sample types, and the training sample points on Z1 and Z2 are called support vectors. The goal is to find the vector ω^t and *b* through training so as to maximise the distance between Z1 and Z2 and achieve optimal classification.

A kernel function is used when the problem is linearly inseparable [42]. The essential idea behind using a kernel function to solve the problem of linear inseparability is to map the original sample into a high-dimensional space, allowing for linear separation in the high-dimensional feature space. Then, the linear classifier SVM is applied for classification. LIBSVM, an SVM software package developed by Taiwan University [43], was used to construct the classification model.

2.4. Radical basis function neural network (RBF Neural Network)

The concept of the RBF Network was presented by Broomhead and Lowe in 1988 [44]. As shown in Figure 9, the RBF neural network is a three-layer forward network with a single hidden layer. The first layer is the input layer, which is composed of signal source nodes. The second layer is the hidden layer, and the number of nodes in the hidden layer depends on the specific problem being addressed. The activation function of neurons in the hidden layer, namely the radial basis function, is a radially symmetric and attenuated nonlinear function of the central point. The commonly used activation function is generally a Gaussian function. The third layer is the output layer, which corresponds to the input and uses a linear optimisation strategy. The main idea is to use RBF as the "basis" to form the hidden layer space. The hidden layer transforms low-dimensional data into a high-dimensional space, allowing a linearly non-separable problem to be separated linearly in this space. The RBF network model and activation function can be expressed using the following formulas (1) and (2):

$$y_j = \sum_{i=1}^{n} w_{ij} \varnothing \left(\| x - u_i \|^2 \right), (j = 1, ..., p) \quad (1)$$

$$\varnothing (x - u_i \|^2) = e^{-\frac{\|x-u_j\|^2}{\sigma^2}} \quad (2)$$

where u_i is the central point obtained using the k-means algorithm. φ is the Gaussian function used as an activation function, σ can be calculated by the KNN algorithm, and w_{ij} is the weight.

3. Pavement condition detection

3.1. Detection results using one-dimensional CNN

In this study, CNN was trained using Python. The original time-domain acceleration data were used directly during the training. The number of channels was set to three, corresponding to the X-, Y-, and Z-axes. Of the dataset, 80 % were used for training, 20 % were used as the validation set, and 422 groups of original acceleration data were used to test the classification performance of the model. The training set was used to fit and construct the model, which was preliminarily evaluated using the validation set.

The batch size represents the number of samples trained in each step. The learning rate determines the convergence speed of the model. In this study, the batch size and learning rate were set to 10 and 0.0001, respectively. The epoch was set to 100, and an epoch was trained once with all 337 samples in the training set; therefore, the iteration number was 3370. Figure 10(a) and (b) show the curves of loss and accuracy versus iterations for CNN0. Accuracy is the ratio of the number of correctly classified samples to the total number of samples. The higher the accuracy, the better the classification performance of the model. Loss measures the degree of inconsistency between the predicted and actual values; it is a non-negative real-valued function. The smaller

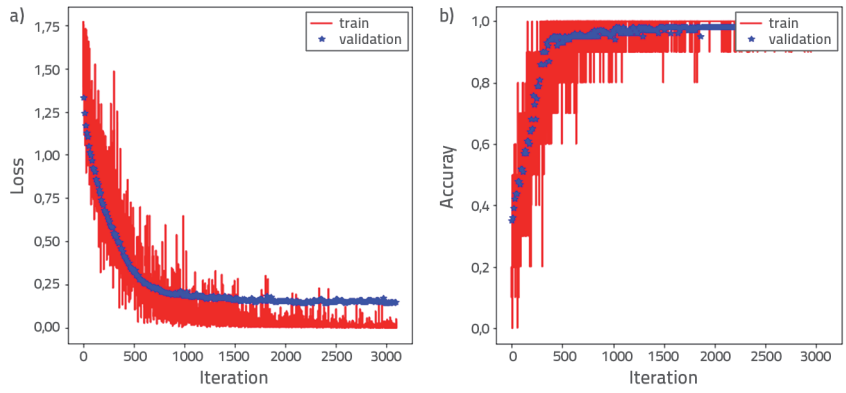

Figure 10. Training and validation results of CNN0: a) Loss-Iteration curve; b) Accuracy-Iteration curve

the loss function, the more robust the model. As the number of iterations increased, the loss value decreased continuously and stabilised at the end. It shows that the model has converged. Simultaneously, the accuracy continued to grow; the training accuracy and validation accuracy of CNN0 reached 1.00 and 0.99, respectively, and the test accuracy reached 0.9976, with 421 samples correctly classified. Details of the training results of CNN0-CNN4 are listed in Table 2, including the Precision, Recall and Accuracy. The calculation method is as follows.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

where true positive (TP) and false positive (FP) are the numbers of correctly classified and incorrectly classified samples as positive. The false negative (FN) is the number of incorrectly classified negative samples. Taking potholes in CNN0 as an example, 101 groups were correctly identified out of 102, with one misclassified as a speed bump, and no other pavement types were misclassified as potholes. Therefore,

TP = 101, FP = 0, FN = 1,

Precision = 101/101 = 1.0000,

Recall = 101/102 = 0.9907.

Table 2 shows that a change in the activation function has a significant effect on CNN performance. The accuracy of CNN1 using tanh and CNN2, using the sigmoid function, were 0.9900 and 0.8571, respectively. This is because, during backpropagation, the weight *w* is calculated using a differential. In tanh and sigmoid, when the input value is large or small, the output is almost smooth, and the gradient is minimal, making weight updates difficult. This problem is also known as gradient saturation. The output interval of tanh is (-1,1), and the entire function is centred at zero. Gradient saturation is less severe than the sigmoid function; thus, CNN1 has higher accuracy. For the popular ReLU function, when the input is positive, there is no gradient saturation problem, and because ReLU has a simple linear relationship, the calculation speed is much faster. After removing the convolution and pooling layers, the accuracy of CNN3 was reduced to 0.9952; however, it still exhibited satisfactory classification performance. CNN0 and CNN4 exhibited the best classification performance; the correct classification instances were all 421, and the accuracy was 0.9976. Considering that CNN0 had a simpler network structure, we believe that CNN0 exhibited the best classification performance.

Table 2. Details of the results of CNNs*

	CNN0	CNN1	CNN2	CNN3	CNN4
TRA	1.0000	1.0000	0.9100	1.0000	1.0000
VDA	0.9900	0.9700	0.8500	0.9800	0.9900
TEA	0.9976	0.9900	0.8571	0.9952	0.9976
CCI	421	417	361	420	421
BPP	1.0000	0.9727	0.8774	0.9907	1.0000
BPR	1.0000	0.9907	0.8611	0.9907	0.9907
SBP	1.0000	1.0000	0.7578	1.0000	1.0000
SBR	1.0000	0.9898	0.9798	1.0000	1.0000
SPP	0.9900	1.0000	0.9072	1.0000	1.0000
SPR	1.0000	0.9912	0.7788	0.9912	1.0000
PTP	1.0000	0.9804	0.9326	0.9902	0.9902
PTR	0.9902	0.9804	0.8137	1.0000	1.0000

TRA: Training Accuracy, VDA: Validation Accuracy; TEA: Test Accuracy; CCI: correct classification instances; BPP: bumpy pavement precision; BPR: bumpy pavement recall; SBP: speed-bump precision; SBR: speed-bump recall; SPP: smooth pavement precision; SPR: smooth pavement recall; PTP: pole precision; PTR: pothole recall

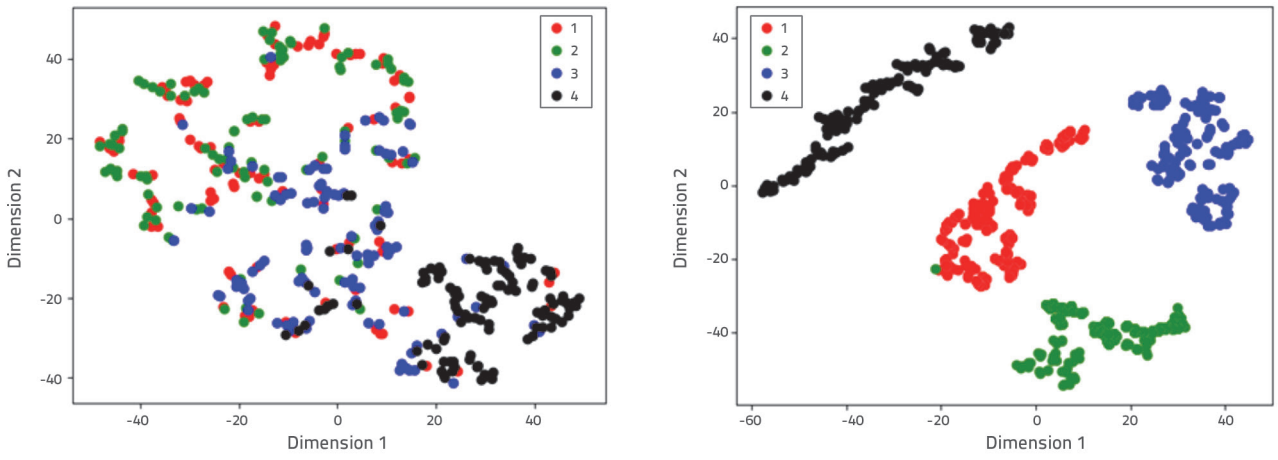

Figure 11. Two-dimensional display of raw data and output data (1-speed bump, 2-potholes, 3-bumpy pavement, 4-smooth pavement)

Finally, to display the classification performance of the model more intuitively, we adopt a visualisation tool named "t-SNE (t-distributed stochastic neighbour embedding)", which can nonlinearly reduce the dimension of data, especially for the visualisation of high-dimensional data. It was used to visualise the raw and output data from the CNN. As shown in Figure 11, after dimensionality reduction, there is no apparent distinction between different classes on the two-dimensional plane. However, after the CNN0 classification, there were noticeable differences among the other classes, showing that CNN0 had an excellent classification performance.

3.2. Detection results using machine learning algorithm

3.2.1. Dataset

In this study, two traditional machine-learning algorithms, an SVM and a Radial Basis Function (RBF) neural network, were selected for comparison with a CNN. As mentioned before, a CNN uses convolution and pooling layers to extract features and then classifies them through dense layers. Therefore, it was not necessary to extract features prior to inputting the data. While SVM and RBF networks are classifiers, they must extract features before inputting the acceleration data.

In contrast to CNN, the datasets used in the SVM and RBF networks consist of feature values. 26 features of the time and frequency domains were calculated using MATLAB, and each feature contained three directions, X, Y, and Z; thus, so a total of 78 attribute values were obtained. Then, 28 of them were chosen by "CfsSubsetEval" in WEKA. The attribute values listed in Table 3, 1-57 are from the time domain (TD), and the others are from the frequency domain (FD).

Finally, we obtained a dataset consisting of a 422 × 29 matrix, the last column of which was labelled.

Table 3. Attribute types

Number	Attribute	Number	Attribute
1-3	TD Mean (x/y/z)	40-42	TD Pulse factor (x/y/z)
4-6	TD Variance (x/y/z)	43-45	TD Margin (x/y/z)
7-9	TD Correlation coefficient (xy/xz/yz)	46-48	TD Kurtosis (x/y/z)
10-12	TD Maximum (x/y/z)	49-51	TD Waveform factor
13-15	TD Minimum (x/y/z)	52-54	TD Skewness (x/y/z)
16-18	TD Standard deviation (x/y/z)	55-57	FD Mean (x/y/z)
19-21	TD Peak to a peak value (x/y/z)	58-60	FD Standard deviation (x/y/z)
22-24	TD Median (x/y/z)	61-63	FD Maximum (x/y/z)
25-27	TD Absolute peak value (x/y/z)	64-66	FD Peak frequency (x/y/z)
28-30	TD Absolute mean (x/y/z)	67-69	FD Peak factor (x/y/z)
31-33	TD Root mean square value (x/y/z)	70-72	FD Kurtosis (x/y/z)
34-36	TD Interquartile range (x/y/z)	73-75	FD Skewness (x/y/z)
37-39	TD Peak factor (x/y/z)	76-78	FD Entropy (x/y/z)
Selected attributes	2, 3, 5, 8, 9, 19, 24, 26, 30, 34, 35, 36, 38, 39, 45, 46, 47, 50, 51, 52, 57, 58, 59, 64, 65, 66, 68, 69		

Table 4. Test results using SVM (Kernel function: Radial basis function)

Classified as:	Bumpy pavement	Speed bump	Smooth pavement	Pothole
Bumpy pavement	91	7	8	2
Speed bump	0	98	0	1
Smooth pavement	6	3	104	0
Pothole	1	7	0	94

Table 5. Test results using SVM (Kernel function: Polynomial)

Classified as:	Bumpy pavement	Speed bump	Smooth pavement	Pothole
Bumpy pavement	99	0	2	7
Speed bump	0	97	0	2
Smooth pavement	2	0	110	1
Pothole	2	1	0	99

Table 6. Test results using RBF Network

Classified as:	Bumpy pavement	Speed bump	Smooth pavement	Pothole
Bumpy pavement	105	0	0	3
Speed bump	0	98	0	1
Smooth pavement	1	0	112	0
Pothole	1	0	0	101

3.2.2. Detection results

Two different activation functions of the SVM (Radial Basis Function and Polynomial) and the RBF neural network were used, and a ten-fold cross-validation method was adopted for the classification test. The results are shown in Table 4–6. The numbers of correctly classified instances were 387, 405, and 417, and the accuracies were 0.9171, 0.9597, and 0.9858, respectively.

3.3. Comparison

CNN0, which had the best classification performance among the CNNs, was selected for comparison with SVM and RBF neural networks. The results are listed in Table 7. It can be clearly seen that CNN0 has the best effect. CNN0 achieved an accuracy of 0.9976, which was 8.1 %, 3.8 %, and 1.2 % higher than those of the other three algorithms. This shows that CNN

Table 7. Details of the comparison results

	CNN0	SVMR	SVMP	RBF
TA	0.9976	0.9171	0.9597	0.9858
CCI	421	387	405	416
BPP	1.0000	0.9286	0.9612	0.9813
BPR	1.0000	0.8426	0.9167	0.9722
SBP	1.0000	0.8522	0.9898	1.0000
SBR	1.0000	0.9899	0.9798	0.9899
SPP	1.0000	0.9286	0.9821	1.0000
SPR	0.9907	0.9204	0.9735	0.6612
PTP	0.9902	0.9691	0.9083	0.9619
PTR	1.0000	0.9216	0.9706	0.9902

TA: Test Accuracy, CCI: correctly classified instances, BPP: bumpy pavement precision, BPR: bumpy pavement recall, SBP: speed bump precision, SBR: speed bump recall, SPP: smooth pavement precision, SPR: smooth pavement recall, PTP: pole precision, PTR: pole recall.

can not only classify objects in images but can also be applied to the detection of acceleration data. Moreover, compared with the traditional machine learning algorithm, CNN does not need to extract features manually, making the entire recognition and classification process more efficient. CNN shows strong potential for application to one-dimensional data problems.

4. Discussion

This paper proposes a novel pavement condition detection method, and its contributions are as follows. This study proposes a pavement condition detection method based on public participation. By utilising widely available bicycles and smartphones, the general public can participate in data collection and pavement condition assessments, allowing professionals to conduct further evaluations based on summarised data without the need for on-site inspections, which can significantly improve operational efficiency.

In terms of the technical means of detection, we proposed the use of a one-dimensional CNN. Unlike other studies, the application of a 1D-CNN does not require setting specific thresholds or manually designing and extracting features. This study also has limitations that need to be addressed in future research:

- The generalisability of the proposed method has not yet been tested. The data collection in this study was mainly carried out on a university campus, and more training data, as well as more road condition categories, such as rutting, uphill, and downhill, are needed to achieve better practical results.
- Acceleration data at multiple riding speeds were not collected. In real-world public participation, the riding speeds of different individuals may vary significantly; therefore, acceleration data at different riding speeds must be collected. Additionally, smartphone models may impact the final data, necessitating analysis to analyse the data collected from different phone models.

5. Conclusion

A pavement condition detection method using acceleration data classification based on CNNs was proposed. In the experiment, 422 groups of acceleration data, including the X-, Y-, and Z-axes, were collected using a smartphone and bicycle, covering bumpy pavement, speed bumps, smooth pavement, and potholes. Five 1D-CNNs were designed and trained as detectors, leading to the following conclusions.

- For the acceleration data classification and recognition task in this study, applying the ReLU activation function has better performance.
- Adding convolutional layers and pooling layers will improve the feature extraction and classification ability.
- The 1D-CNN with two convolutional layers, two pooling layers, and a ReLU activation function achieved a classification accuracy of 0.9976, which was the best performance among the five CNNs.
- Using CNN, there is no need to design and extract feature values manually, so it is more efficient than traditional machine learning algorithms
- Among the machine learning algorithms applied in this study, RBF neural networks demonstrated higher classification accuracy.

The novelty of this study is the idea of assessment based on public participation; instead of focusing CNN solely on image data, we used it to handle time-series acceleration data and achieved satisfactory detection results. This study had some limitations in terms of generalisability and experimental design. In future research, additional data will be collected to improve the generalisation ability of the network model and conduct data collection experiments with different riding speeds and smartphones. Additionally, we will refine the application of pavement acceleration analysis so that it can be implemented on a small scale as soon as possible.

Acknowledgments

This work was supported by the Basic Research Program for Universities of the Educational Department of Liaoning Province under Grant JYTQN2023241, the Outstanding Young Scientist Program of the University of Science and Technology Liaoning under Grant 2023YQ03, and the University of Science and Technology Liaoning Talent Project Grants.

REFERENCES

[1] Hrapović, K.: Sustainability in road construction – Two case studies, GRAĐEVINAR, 76 (2024) 5, pp. 413-423, https://doi.org/10.14256/JCE.3979.2024.

[2] Zhang, Y., Li, H., Zhang, Y., Zang, W., Yao, K.: Optimal prioritisation of asphalt pavement maintenance using grey relation analysis and cost-benefit analysis, GRAĐEVINAR, 75 (2023) 12, pp. 1155-1164, https://doi.org/10.14256/JCE.3822.2023.

[3] Pradena, M., Houben, L., César, A.: Experimental evaluation of load transfer efficiency of non-dowelled concrete pavements, GRAĐEVINAR, 74 (2022) 3, pp. 179-187, https://doi.org/10.14256/JCE.2844.2019.

[4] Gáspár, L., Bencze, Z.: Increasing the life expectancy of road pavements, GRAĐEVINAR, 72 (2020) 6, pp. 515-522, https://doi.org/10.14256/JCE.2644.2019.

[5] Chang, K.T., Chang, J.R., Liu, J.K.: Detection of pavement distresses using 3D laser scanning technology, In Computing in civil engineering (2005), pp. 1-11, https://doi.org/10.1061/40794(179)103.

[6] Li, Q., Yao, M., Yao, X., Xu, B.: A real-time 3D scanning system for pavement distortion inspection, Measurement Science and Technology, 21 (2009) 1, 015702, https://doi.org/10.1088/0957-0233/21/1/015702.

[7] Wang, K.C.: Challenges and feasibility for comprehensive automated survey of pavement conditions, In Applications of advanced technologies in transportation engineering (2004), pp. 531-536, https://doi.org/10.1061/40730(144)99.

[8] Hou, Z., Wang, K.C., Gong, W.: Experimentation of 3D pavement imaging through stereovision, In International Conference on Transportation Engineering 2007, pp. 376-381, https://doi.org/10.1061/40932(246)62.

[9] Joubert, D., Tyatyantsi, A., Mphahlehle, J., Manchidi, V.: Pothole tagging system, In Proceedings of the 4th Robotics and Mechatronics Conference of South Africa, (2011), pp. 1-4.

[10] Moazzam, I., Kamal, K., Mathavan, S., Usman, S., Rahman, M.: Metrology and visualization of potholes using the microsoft kinect sensor, In 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 1284-1291, https://doi.org/10.1109/ITSC.2013.6728408.

[11] Koch, C., Brilakis, I.: Pothole detection in asphalt pavement images, Advanced Engineering Informatics, 25 (2011) 3, pp. 507-515, https://doi.org/10.1016/j.aei.2011.01.002.

[12] Hoang, N.D.: An artificial intelligence method for asphalt pavement pothole detection using least squares support vector machine and neural network with steerable filter-based feature extraction, Advances in Civil Engineering, 2018, https://doi.org/10.1155/2018/7419058.

[13] Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Gross, H.M.: How to get pavement distress detection ready for deep learning? A systematic approach, In 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2039-2047, https://doi.org/10.1109/IJCNN.2017.7966101.

[14] Li, B., Wang, K.C., Zhang, A., Yang, E., Wang, G.: Automatic classification of pavement crack using deep convolutional neural network, International Journal of Pavement Engineering, 21 (2020) 4, pp. 457-463, https://doi.org/10.1080/10298436.2018.1485917.

[15] Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Mraz, A., Kashiyama, T., Sekimoto, Y.: Deep learning-based road damage detection and classification for multiple countries, Automation in Construction, 132 (2021), 103935, https://doi.org/10.1016/j.autcon.2021.103935.

[16] Du, Y., Pan, N., Xu, Z., Deng, F., Shen, Y., Kang, H.: Pavement distress detection and classification based on YOLO network, International Journal of Pavement Engineering, 22 (2021) 13, pp. 1659-1672, https://doi.org/10.1080/10298436.2020.1714047.

[17] Li, L., Fang, B., Zhu, J.: Performance Analysis of the YOLOv4 Algorithm for Pavement Damage Image Detection with Different Embedding Positions of CBAM Modules, Applied Sciences, 12 (2022) 19, 10180, https://doi.org/10.3390/app121910180.

[18] Guo, G., Zhang, Z.: Road damage detection algorithm for improved YOLOv5, Scientific Reports, 12 (2022) 1, 1-12, https://doi.org/10.1038/s41598-022-19674-8.

[19] Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J.: Complex human activity recognition using smartphone and wrist-worn motion sensors, Sensors, 16 (2016) 4, 426, https://doi.org/10.3390/s16040426.

[20] Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., Roning, J.: Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines, In 2009 17th Mediterranean Conference on Control and Automation, pp. 401-405, https://doi.org/10.1109/MED.2009.5164574.

[21] Ryu, J., Seo, J., Jebelli, H., Lee, S.: Automated action recognition using an accelerometer-embedded wristband-type activity tracker, Journal of Construction Engineering and Management, 145 (2019) 1, 04018114, https://doi.org/10.1061/(ASCE)CO.1943-7862.0001579.

[22] Sanhudo, L., Calvetti, D., Martins, J.P., Ramos, N.M., Mêda, P., Gonçalves, M.C., Sousa, H.: Activity classification using accelerometers and machine learning for complex construction worker activities, Journal of Building Engineering, 35 (2021), 102001, https://doi.org/10.1016/j.jobe.2020.102001.

[23] Zhang, Y., Yuen, K.V.: Review of artificial intelligence-based bridge damage detection, Advances in Mechanical Engineering, 14 (2022) 9, 16878132221122770, https://doi.org/10.1177/16878132221122770.

[24] Bao, Y., Li, H.: Machine learning paradigm for structural health monitoring, Structural Health Monitoring, 20 (2021) 4, pp. 1353-1372, https://doi.org/10.1177/1475921720972416.

[25] Feng, S., Jia, J.: Acceleration sensor placement technique for vibration test in structural health monitoring using microhabitat frog-leaping algorithm, Structural Health Monitoring, 17 (2018) 2, pp. 169-184, https://doi.org/10.1177/1475921716688372.

[26] Pentaris, F.P., Stonham, J., Makris, J.P.: A cost effective wireless structural health monitoring network for buildings in earthquake zones, Smart Materials and Structures, 23 (2014) 10, 105010, https://doi.org/10.1088/0964-1726/23/10/105010.

[27] Fan, G., Li, J., Hao, H.: Dynamic response reconstruction for structural health monitoring using densely connected convolutional networks, Structural Health Monitoring, 20 (2021) 4, pp. 1373-1391, https://doi.org/10.1177/1475921720916881.

[28] Kazantzi, A.K., Vamvatsikos, D., Miranda, E.: Evaluation of seismic acceleration demands on building nonstructural elements, Journal of Structural Engineering, 146 (2020) 7, 04020118, https://doi.org/10.1061/(ASCE)ST.1943-541X.0002676.

[29] Yu, B.X., Yu, X.: Vibration-based system for pavement condition evaluation, In Applications of advanced technology in transportation, pp. 183-189, https://doi.org/10.1061/40799(213)31.

[30] Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., Balakrishnan, H.: The pothole patrol: using a mobile sensor network for road surface monitoring, In Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, pp. 29-39, https://doi.org/10.1145/1378600.1378605.

[31] Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., Selavo, L.: Real time pothole detection using android smartphones with accelerometers, In 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), pp. 1-6, https://doi.org/10.1109/DCOSS.2011.5982206.

[32] Fox, A., Kumar, B.V., Chen, J., Bai, F.: Crowdsourcing undersampled vehicular sensor data for pothole detection, In 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 515-523, https://doi.org/10.1109/SAHCN.2015.7338353.

[33] Bhatt, U., Mani, S., Xi, E., Kolter, J.Z.: Intelligent pothole detection and road condition assessment, arXiv preprint arXiv:1710.02595 (2017).

[34] Du, R., Qiu, G., Gao, K., Hu, L., Liu, L.: Abnormal road surface recognition based on smartphone acceleration sensor, Sensors, 20 (2020) 2, 451, https://doi.org/10.3390/s20020451.

[35] Yang, Q., Zhou, S.: Identification of asphalt pavement transverse cracking based on vehicle vibration signal analysis, Road Materials and Pavement Design, 22 (2021) 8, pp. 1780-1798, https://doi.org/10.1080/14680629.2020.1714699.

[36] Egaji, O.A., Evans, G., Griffiths, M.G., Islas, G.: Real-time machine learning-based approach for pothole detection, Expert Systems with Applications, 184 (2021), 115562, https://doi.org/10.1016/j.eswa.2021.115562.

[37] Zhang, Z., Zhang, H., Xu, S., Lv, W.: Pavement roughness evaluation method based on the theoretical relationship between acceleration measured by smartphone and IRI, International Journal of Pavement Engineering, (2021), https://doi.org/10.1080/10298436.2021.1881783.

[38] Kong, Q., Allen, R.M., Kohler, M.D., Heaton, T.H., Bunn, J.: Structural health monitoring of buildings using smartphone sensors, Seismological Research Letters, 89 (2018) 2A, pp. 594-602, https://doi.org/10.1785/0220170111.

[39] Figueiredo, E., Moldovan, I., Alves, P., Rebelo, H., Souza, L.: Smartphone Application for Structural Health Monitoring of Bridges, Sensors, 22 (2022) 21, 8483, https://doi.org/10.3390/s22218483.

[40] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86 (1998) 11, pp. 2278-2324, https://doi.org/10.1109/5.726791.

[41] Cortes, C., Vapnik, V.: Support-vector networks, Machine Learning, 20 (1995) 3, pp. 273-297, https://doi.org/10.1007/BF00994018.

[42] Amari, S.I., Wu, S.: Improving support vector machine classifiers by modifying kernel functions, Neural Networks, 12 (1999) 6, pp. 783-789, https://doi.org/10.1016/S0893-6080(99)00032-5.

[43] Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology (TIST), 2 (2011) 3, pp. 1-27, https://doi.org/10.1145/1961189.1961199.

[44] Broomhead, D.S., Lowe, D.: Radial basis functions, multi-variable functional interpolation and adaptive networks, Royal Signals and Radar Establishment Malvern (United Kingdom), (1988).