COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **0** (2018) 1–17

Convolutional Neural Network for Asphalt Pavement Surface Texture Analysis

Zheng Tong, Jie Gao, Aimin Sha* & Liqun Hu

School of Highway, Chang'an University, Xi'an, China

&

Shuai Li

School of Materials Science and Engineering, Chang'an University, Xi'an, China

Abstract: *Several data processing techniques (DPTs) have been implemented for evaluating pavement surface texture to partially replace onsite inspections by humans. However, the extensively varying real-world situations have resulted in challenges in the widespread adoption of DPTs. To overcome these challenges, we propose the use of a convolutional neural network (CNN) to calculate the mean texture depth (MTD) without computing the surface texture feature statistics. Because a CNN is capable of automatically learning data features, the proposed method does not require the conjugation of DPTs for extracting features. The proposed CNN was trained and tested using 8,000 and 1,000 3D scan data samples, respectively, and achieved an average error of 0.0024 cm. The stability of the CNN was analyzed based on various test results. Comparative studies were conducted to verify the superiority of the CNN over conventional MTD algorithms. The results demonstrated that the CNN-based method is significantly more stable in various real-world situations. Additionally, the CNN-based method achieved higher accuracy of automatic feature extraction than traditional MTD methods. Finally, the CNN-based method was applied to evaluate the surface texture statistics of four highways in Shanxi, China, which were different from the training and testing samples; the results establish the transferability of this method to different highways.*

1 INTRODUCTION

With the rapid development of pavement construction in China, it has become increasingly important to inspect pavement lifecycle-performance and quality, during its construction. Pavement surface texture is a key parameter for evaluating and predicting asphalt pavement lifecycle-performance and quality, during its construction (Alhasan et al., 2016; Mosa et al., 2013). The pavement surface texture reflects both its drainage performance and rolling resistance (Bendtsen, 2004; Sohaney and Rasmussen, 2013). It also influences vehicle operation, durability, noise emission, and fuel consumption (Zaniewski et al., 1982; Oijer and Edlund, 2004). Therefore, it is important to evaluate pavement surface texture rapidly and accurately.

The evaluation parameters for pavement surface texture are highly numerous and diverse. These parameters include the surface macrotexture, mega-texture for vehicle suspension responses, and international roughness index. As a key index, the surface macrotexture is related to the tire-sidewall deformation of moving vehicles (Wang et al., 2010), energy dissipation (Zhou et al., 2014), and rolling resistance (Praticò and Vaiana, 2015; Yuan and Lai, 2010). Therefore, it is essential to measure the macrotexture while evaluating and predicting the pavement surface texture.

*To whom correspondence should be addressed. E-mail: *ams@chd.edu.cn*.

DOI: 10.1111/mice.12406

The surface macrotexture (wavelengths between 0.5 and 50 mm) is assessed using both intrinsic and extrinsic indicators. The extrinsic indicators such as the pendulum friction coefficient reflect the influence of the surface macrotexture on vehicles. The intrinsic indicators such as mean texture depth (MTD) and mean profile depth (MPD) reflect both the geometry and surface characteristics of surfaces and the indirect influences of the surfaces on vehicles. The main methods for assessing intrinsic indicators include volumetric, laser-based, and 3D scan methods. The typical indicator used for volumetric methods is the MTD (Yuan and Lai, 2010), which can be calculated from the ratio between the volume and surface area. The MPD is a typical indicator for laser-based methods (Wang et al., 2010; Zhou et al., 2014). It is calculated from the ratio of the cross-sectional area to the length. 3D scanning methods rely on accurately scanning the pavement and evaluation of the macrotexture characteristics from the scanned data (Garbowski and Gajewski, 2017; Čelko et al., 2016).

The sand patch method is a popular volumetric method that is widely used as a standard for pavement condition assessment (ASTM E965-96, 1996; EN 13036-1, 2010). Particularly, the electric sand patch method, which is more precise than the manual sand patch test, is widely used in China (JTG H20-2007, 2007). In practice, the sand patch method has demonstrated remarkable performance of evaluation of asphalt surface texture exhibiting a close relationship between the MTD and macrotexture (Praticò and Vaiana, 2015); however, it also exhibits certain disadvantages. The sand patch method is not observed to be reliable if it is used on porous surfaces because of the negative effects of the pavement pores (ISO 13473-1, 1997). Additionally, the sand patch method is not appropriate for certain surface conditions such as freshly paved surfaces and wet pavements (JTG H20-2007, 2007). With regard to laser-based methods, laser devices scan surface profiles and calculate the MPD based on the profile height coordinates. This method is less affected by pavement conditions compared to the sand patch method. However, an MPD calculated only from a profile cannot fully characterize the pavement surface texture. Various studies (Miao et al., 2011; Torbruegge and Wies, 2015) have verified that the relationship between MPD and skid resistance was not as close as expected when using laser-based methods. For research on 3D scan methods, numerous indexes such as power spectrum energy (Gendy et al., 2011), root mean squared error (Kanafi et al., 2014), and estimated texture depth (Vilaca et al., 2010) have been developed to represent the surface macrotexture based on 3D scanned data. However, these indexes are not likely to capture the expected correlations between macrotexture and pavement performance. Therefore, it is necessary to develop a method combining MTD and 3D scan data to overcome the disadvantages mentioned above. However, it is challenging to create a formula, unsupervised algorithm, or supervised algorithm (Adeli, 2001; Amezquita-Sanchez, 2006) such as artificial neural network (ANN) to identify the relationship between 3D scan data and MTD for understanding the effects of the various pavement materials. This is because different pavement materials have different mean depth planes (MDPs) and maximum height of the disturbing sand, which are mainly affected by the pavement materials, geometry, and surface characteristics. The MDP and maximum height of the disturbing sand are key features for calculating the MTD (Hu et al., 2016; Slabej and Kotek, 2014). A straightforward formula or unsupervised algorithm cannot extract the different MDPs and maximum heights from the 3D scan data of various pavement materials (Wen, 2009).

With the continuous development of deep learning techniques, convolutional neural networks (CNNs) have demonstrated significant advantages in the field of object recognition, such as string recognition (Barat and Ducottet, 2016), script identification (Shi et al., 2016), 3D object retrieval (Leng et al., 2015), and human-action recognition (Xu et al., 2016b). First proposed by Lecun et al. (LeCun and Bengio, 1995; LeCun, 1989), CNNs can be considered as a type of supervised highly nonlinear mapping that output target-features in a specified format based on the input data. CNNs have been successfully applied in the fields of 2D image recognition (Ortega-Zamorano et al., 2017), 3D object recognition (Fan et al., 2010), natural language processing (Zeinalia and Story, 2017), audio classification (Zhu et al., 2016), brain–computer interaction (Cecotti and Graser, 2011), human tracking (Fan et al., 2010), image restoration (Jain et al., 2007), denoising (Koziarski and Cyganek, 2017; Jain and Seung, 2008), and segmentation (Turaga et al., 2010). Detailed information regarding CNNs is available in Leng and Xiong (2011) and Leng et al. (2010). For example, Rafiei (Rafiei and Adeli, 2017) used a novel machine learning model for estimating sale prices of real estate. The robustness and automation of feature extraction are generally considered desirable properties in various engineering fields. Robustness indicates a strong tolerance for translation and distortion while learning deep features from input data. Automation implies learning deep features with no assistance from humans. These two properties are important for regression problems in MTD analysis when handling the complex foreground, background, and feature information in 3D scan data and extracting different MDPs from various pavement materials. Therefore, CNNs can be utilized to construct a relationship between scanned data and MTD.

A CNN is a type of ANN whose structure of shared weights reduces the complexity of a network model. Its structure is similar to that of biological neural networks (Xu et al., 2016a; Ijjina and Chalavadi, 2016). Data are directly used as inputs in CNN algorithms, which eliminates the complex data preprocessing steps required for conventional unsupervised algorithms. Complex preprocessing is likely to result in the deletion of important data. Additionally, the CNN network structure exhibits high tolerance for transformation (Arevalo et al., 2016), translation (Li et al., 2016), scaling (Li et al., 2016), tilting (Liu et al., 2016), etc. This property is useful for analyzing complex 3D scan data. CNNs have been used to detect structure performance in recent years. For example, Tong and colleagues (Tong et al., 2018; Tong et al., 2017) employed a CNN to calculate the lengths of pavement cracks and concealed cracks based on image data and ground penetrating radar data, respectively. The results of these studies demonstrated that CNNs exhibit remarkable stability notwithstanding the influence of pavement materials and other interference factors. Cha and colleagues (Cha, et al., 2017a; Cha et al., 2017b) detected pavement damage using CNNs, and the results demonstrated that CNNs are capable of identifying concrete-cracks in real-world situations. Lin et al. (2017) detected structural damage using automatic feature-extraction via deep learning. Liao presented a deep-learning method for carbon emission reduction (Liao, 2017). Additionally, there have been a few studies combining CNNs with images for damage detection, in the past few years (Rafiei et al., 2017; Koziarski and Cyganek, 2017). Therefore, it is reasonable to employ CNNs to construct a relationship between the 3D scanned data and MTD based on the major factors introduced above.

In this study, we used a CNN to construct a relationship between 3D scan data and MTD. The first task in this research was collecting pavement surface 3D scan data for training and testing data sets and identifying a deep representation of these data sets for our CNN. The second task was to construct an accurate CNN that was robust to various pavement surfaces to guarantee stability and wide-range adaptability. The main advantages of the proposed CNN-based evaluation of MTD are that it requires no manual feature extraction and that its prediction results are accurate and robust to various real-word conditions (e.g., different pavement materials). Our goal is to replace the sand patch method with 3D scanning technology and CNNs in certain conditions where the sand patch method is not convenient for evaluating MTD, such as gap-graded surfaces, surfaces with defects, continuous pavement construction, and freeway-condition inspection without affecting traffic capacity. The remainder of this article can be summarized as follows: our methodology, focusing on a method to scan pavement surfaces, an architectural description of our CNN, and a few important functions and parameters for network learning, is introduced in Section 2. This is followed by a discussion of the performance of our CNN-based method in Section 3. The performance discussion of the CNN-based method includes the performance of training and testing, stability analysis, comparative studies, and an application of a trained CNN to four highways in Shanxi, China. Notably, the four highways used for the application of the trained CNN were not the ones used to test the CNN after training. Our conclusions are summarized in Section 4.

2 METHODOLOGY

This section presents a method to scan pavement surfaces, an architectural description of our CNN, and a few other important methods such as the backpropagation algorithm. The methods used to collect the surface texture data and generate the training and testing data sets are described in Section 2.1; this is followed by the description of our CNN architecture in Section 2.2. Finally, other important functions and parameters used in this study are described in Section 2.3.

2.1 Building data sets for CNN

2.1.1 Scanning method. The first step for developing our CNN was to collect information regarding surface texture features using a specific method. Generally, the methods for 3D object representation are divided into model-based and view-based approaches. Model-based methods, generally referred to as 3D scanning, detect digital objects directly from the original 3D objects using range measuring systems (Osada et al., 2002; Patane et al., 2009). View-based methods (Passalis et al., 2006; Ansary et al., 2007), generally referred to as structure from motion, 3D reconstruction, and photogrammetry 3D reconstruction, first capture 2D images of the original objects from fixed viewpoints using 2D cameras and then use these images as object information. Surface texture features cannot be accurately reflected by view-based methods. However, 3D scan technology has achieved remarkable performance of generation of 3D model representations of pavement, as demonstrated in the research by Hu et al. (2016). Therefore, a model-based method was employed in this research.

A multiple-point 3D scanner was used to generate point clouds of pavement surfaces from 52 core samples. The distributions of the training and testing core

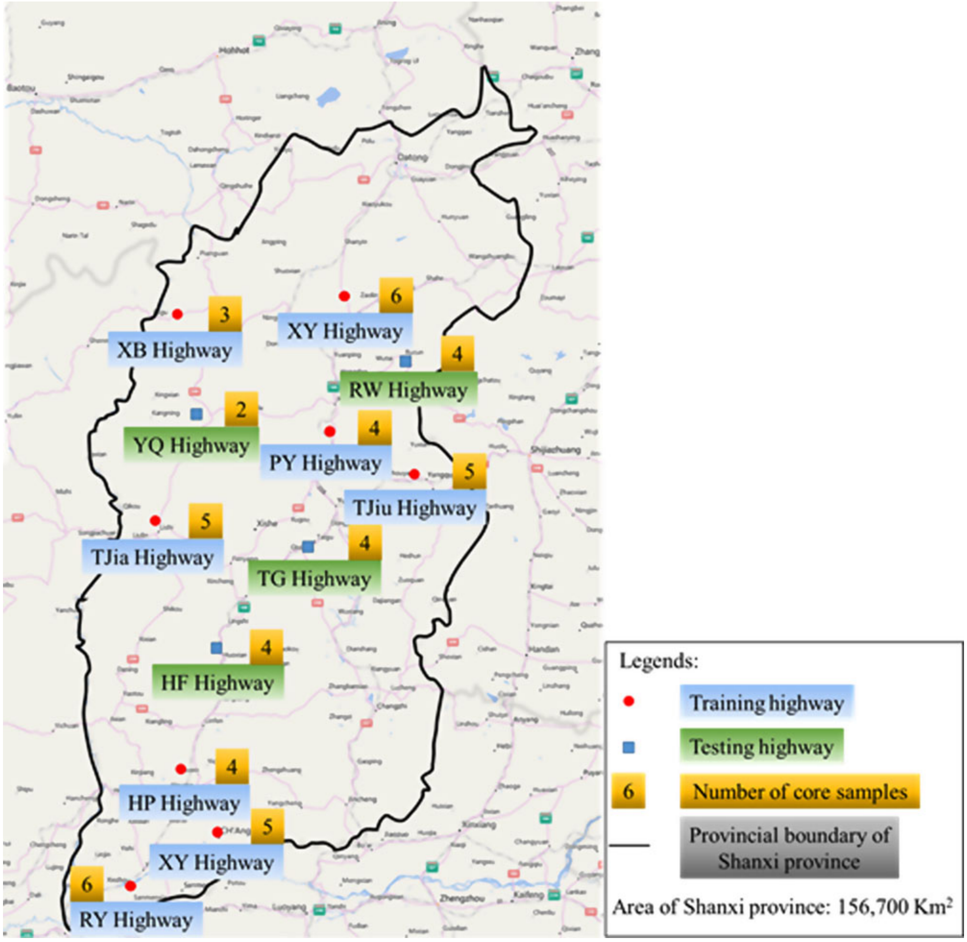


Fig. 1. The distributions of training and testing cores.

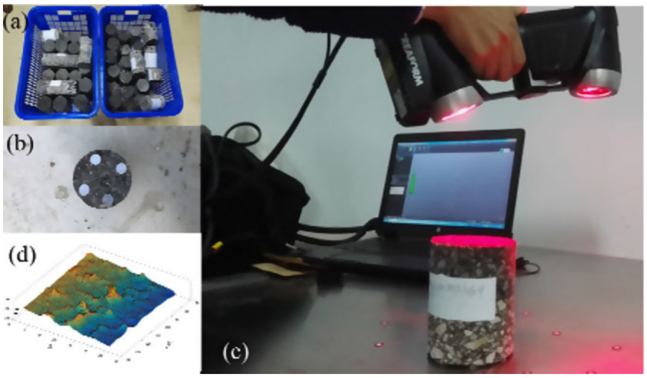


Fig. 2. Processes and results of 3D surface images.

samples are illustrated in Figure 1. The number of dots per inch (DPI) of the point clouds was 100. The resolution in both the *x* and *y* directions is 0.25 mm. The scanning process is illustrated in Figures 2a–c. As shown in Figure 2c, four positions were defined on a core sample, and only the results inside the four positions were retained for generating point clouds. Then, 3D surface reconstruction of this point cloud of the core sample was performed. An example result of 3D surface reconstruction is presented in Figure 2d. By shifting the four positions, different 3D surface images could be collected.

2.1.2 Training and testing data generation. The selection of training data is a key issue because the performance of a CNN is largely dependent on the number and quality of the training samples. In general, it is advantageous to acquire as many training data as possible or to collect training data that is representative of a larger pavement structure. However, collecting a large amount of training data implied the necessity of scanning a large area of the pavement. Using the method introduced in Section 2.1.1, 9,000 3D scanning results were produced from 52 core samples. Eight thousand of the 9,000 3D scanning results from 38 core samples of eight training highways were used to train the CNN. The remaining 1,000 3D scanning results from the remaining core samples of four testing highways were used as the testing data set. The distributions of the training and testing highways are shown in Figure 1.

The target data is another important issue when developing a CNN. As mentioned in the Introduction, the main goals behind using a CNN to replace the sand patch test are as follows: to acquire MTD results close to the MTD measured by the sand patch test, reduce the measurement time for pavement detection, prevent operator error, and provide an alternative method for use in certain conditions that are unsuitable for the sand patch test. To measure the MTD of the scan range, a modified electric sand patch test was utilized. In the modified sand patch test, we first calibrated the electric sand patch device by executing a series of tests on a glass plate. Then, we patched sand in a scan range on a core sample. Dry sand was gradually deposited in the scan range, whose boundary was defined by the four aforementioned positions, after scanning. We patched 5 ml of sand in a 75-mm^2 area. The constant width of the sand patching area was 75 mm. After the sand was completely patched, we measured the length of the sand patching area. Then, the MTD was calculated using Equation (1). The test on the core sample had the same device and operation as the field test, which could guarantee results with a similar accuracy, although the evaluation areas were different. The obtained MTD of the core sample was close to the MTD obtained in the field test conducted at the location from where the core sample was collected.

$$MTD = \frac{Volume\ of\ dry\ sands}{Area\ of\ scan\ range} \tag{1}$$

2.2 Architecture description

Our CNN was created using multiple layers including input, convolutional, pooling, and output layers (LeCun and Bengio, 1995; LeCun, 1989). A deep representation for the pavement is derived in the input layer. The convolution and pooling operations are conducted in the convolutional and pooling layers, respectively. Figure 3 presents the CNN architecture used in our research. Table 1 lists the detailed dimensions of each layer and operation.

2.2.1 Input layer. The first layer of the CNN is the input layer. Considering the fixed format requirement for the input data fed into the input layer (Leng and Xiong, 2011), a transformation should be performed, as shown in Figure 4. Figure 4a presents a portion of an original data sample, where each column indicates the x, y, and z coordinates of the pavement surface. Figure 4b presents the format after the transformation, wherein the z coordinates are represented in an image by a sequence of x and y coordinates.

2.2.2 Convolutional layer. In contrast to a conventional ANN, a CNN has a special structural layer that is used to extract local features from the training data, which is called a convolutional layer. In our research, given a training data set $\{(x_i, y_i), i = 1, \ldots, n\}$, the input is $x_i \in R^3$ and the target output is $y_i \in R$. The convolutional layer calculates the feature maps in the following form:

$$I_j = \phi\left(\sum_i x_i * \Theta_j + b_j\right) \tag{2}$$

$$\Theta_j = \left[\theta_{ij}\right]_{3\times 3} \tag{3}$$

where Θ_j and b_j are the weight vector and bias vector, respectively, of the convolutional kernel j. Θ_j is calculated as illustrated in Equation (3). θ_{ij} are the connecting weight parameters. θ_{ij} vary from one convolutional kernel to another. Therefore, different convolutional kernels extract different features from the training data set $\{(x_i, y_i), i = 1, \ldots, n\}$. θ_{ij} are acquired from the training stage of the CNN. $\phi()$ represents an activation function. The sigmoid function was utilized as the activation function in our research. I_j represents the feature maps acquired by convolutional kernel j. The feature maps acquired from the convolutional layer encode the local features, and their dimensions depend on those of the input data and convolutional kernels. An example convolution is illustrated in Figure 5. As shown in Figure 3, our convolutional layer consists of 16 convolutional kernels each of size 3 × 3. Sixteen feature maps are acquired after x_i is fed into the convolutional layer.

2.2.3 Max-pooling with dropout. With the goal of capturing the most relevant global features from the training data set, acquiring a form of spatial transformation invariance, and reducing the computation complexity for completely connected layers, a pooling operation is utilized to down-sample the feature maps I acquired from the convolution layer as shown in Equation (4).

$$I_j^{l+1} = pooling\left(I_1^l, \ldots I_i^l, \ldots, I_n^l\right) \tag{4}$$

where I_j^{l+1} is the result of the pooling operation on the feature maps acquired by convolutional kernel j. I_i^l is the feature map acquired by convolutional kernel j. *Pooling()* is the pooling function. i represents the i^{th} input from the 3D scan data described in Section 2.2.2.

An ideal pooling function is expected to preserve task-related information while discarding irrelevant details. Two popular options are average-pooling and

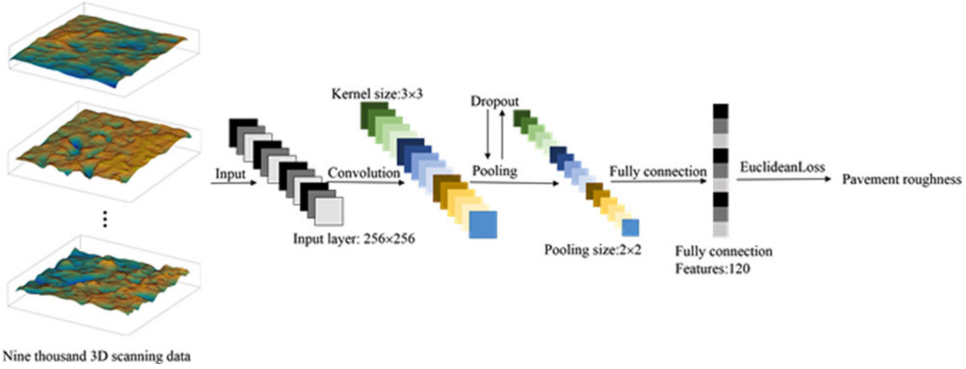
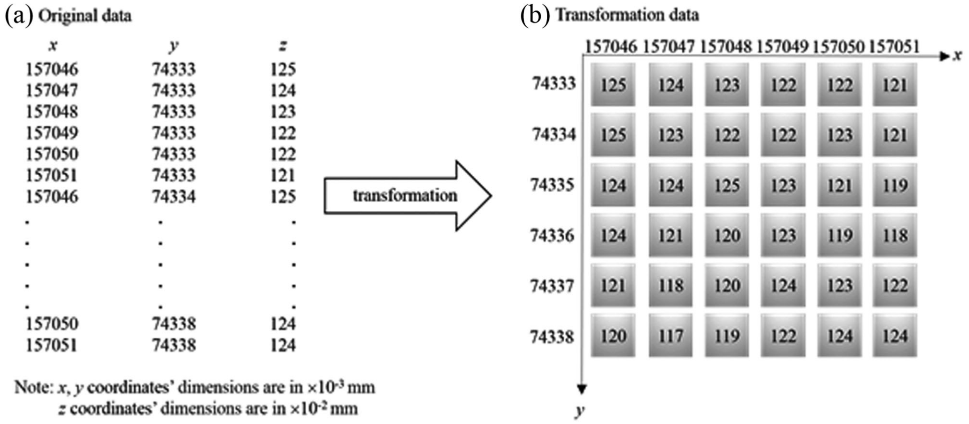


Fig. 3. Neural network for roughness analysis.

Table 1
Dimensions of layers and operations

Layer	*Height*	*Width*	*Depth*	*Operator*	*Height*	*Width*	*Depth*	*Number*	*Stride*
Input	256	256	1	C	3	3	1	16	2
L1	128	128	16	P	2	2	–	–	2
L2	64	64	16	Fc	–	–	–	120	–
L3	–	–	120	A	–	–	–	120	–
L4	–	–	120	E	–	–	–	–	–
Output	1	1	n	–	–	–	–	–	

Note: C = convolution; P = pooling; Fc = full connection; A = activation (sigmoid); E = Euclidean Loss.



(a) Original data

x	*y*	*z*
157046	74333	125
157047	74333	124
157048	74333	123
157049	74333	122
157050	74333	122
157051	74333	121
157046	74334	125
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
157050	74338	124
157051	74338	124

Note: *x*, *y* coordinates' dimensions are in $\times 10^{-3}$ mm
z coordinates' dimensions are in $\times 10^{-2}$ mm

(b) Transformation data

	157046	157047	157048	157049	157050	157051
74333	125	124	123	122	122	121
74334	125	123	122	122	123	121
74335	124	124	125	123	121	119
74336	124	121	120	123	119	118
74337	121	118	120	124	123	122
74338	120	117	119	122	124	124

Fig. 4. Transformation of 3D scan data.

max-pooling. Average-pooling considers all the activations in a pooling region to contribute equally. This is likely to downplay high activations because numerous low activations are included with an identical weight. Max-pooling captures only the strongest feature parameters and disregards all other units in the pooling region. This is likely to downplay medium activations because only high activations are utilized. These two methods evidently have their own advantages and disadvantages. Therefore, we employ a modified pooling method, or more accurately, a "max-pooling with dropout" method to prevent the disadvantages mentioned above.

With the inclusion of dropout, Equation (4) is transformed to Equations (5) and (6):

$$\hat{I}_i^l = m^l * I_i^l \tag{5}$$

$$\hat{I}_j^{l+1} = pooling\left(\hat{I}_1^l, \ldots \hat{I}_i^l, \ldots, \hat{I}_n^l\right) \tag{6}$$

where * and m^l denote an element-wise product and binary mask, respectively, with each element drawn

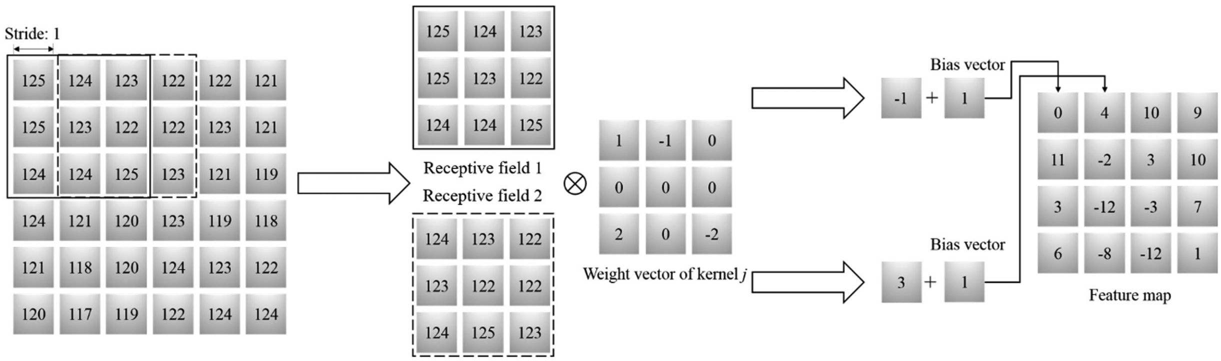


Fig. 5. An example convolution.

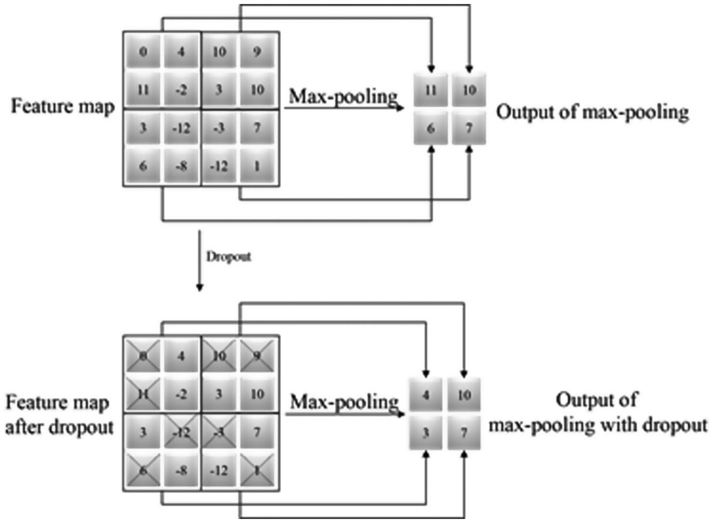


Fig. 6. An illustrative example showing the procedure for max-pooling with dropout. It is supposed that the units (a_1, a_2, a_3, a_4) are reordered as $0 \leq a_1 \leq a_2 \leq a_3 \leq a_4$.

independently from a Bernoulli distribution. Figure 6 presents an example to illustrate the process of max-pooling with dropout. Compared to max-pooling without dropout, where all the strongest activations form the output, max-pooling with dropout is a stochastic procedure because it is not guaranteed that the strongest activations become the outputs.

The process used to formulate this stochastic procedure is summarized as follows:

Step 1 Reorder the units in each pooling region in nondescending order. As shown in Figure 2, suppose the units (a_1, a_2, a_3, a_4) are reordered as $0 \leq a_1 \leq a_2 \leq a_3 \leq a_4$.

Step 2 Set dropout rate. With dropout included, each unit in the pooling region may be set to zero with a probability p_i. The feasibility of the occurrence of this event is illustrated in Figure 6 with the probability set according to probability theory in Equation (7),

$$P(remain\ a_i) = p_i(1 - p_i)^{n-1} \tag{7}$$

where n denotes the number of units in the pooling region. p_i is called the dropout rate and is typically set to 0.5 because the maximum number of events can be extracted using a dropout rate of 0.5 (Srivastava et al., 2014).

Step 3 Handling special events. A special event that occurs with a probability of p_i^n is one wherein all the units in a pooling region are dropped and the results of the pooling operation are zero. A multinomial distribution is adapted to handle this special event. Detailed information regarding the multinomial distribution used in max-pooling with dropout is available in Wu and Gu (2015).

Step 4 Calculate a final output for the pooling region. A final output is received from a max-pooling operation on the condition that a portion of the units in a pooling region is removed and the others are retained.

2.2.4 Output layer. After the inputs x_i pass through the convolutional layers and max-pooling layers described above, a feature representation $\hat{I}_j^{l+1}$ of fixed-size is acquired. The layers following the max-pooling layer are fully connected layer and Euclidean Loss layer. The function of the fully connected layer with connecting weights W_z in our CNN is identical to that in a conventional CNN. For the feature representation set $\hat{I}_j^{l+1}$ with a fixed-sized acquired from the pooling layer, a linear transformation is performed using Equation (8). The output of the fully connected layer is a vector with a fixed number of dimensions.

$$\phi(x_i, W_z) = W_z \hat{I}_j \tag{8}$$

where $\hat{I}_j$ denotes the representation set of $\hat{I}_j^{l+1}$.

In conventional CNNs (Zhu et al., 2016; Fu et al., 2015), the layer following the fully connected layer is a softmax layer, which considers the outputs of the fully connected layer as likely scores for the corresponding classes and then transforms the score vector into a probability distribution. However, considering that the goal of our CNN is to perform quantitative surface texture evaluation rather than simple classification, the softmax layer is replaced with a Euclidean Loss layer in our network. The function of the Euclidean Loss layer in our CNN is to perform regression utilizing the output of the fully connected layer. The regression task in the Euclidean Loss layer is performed by calculating the Euclidean distances between the outputs of the CNN and the targets using Equation (9) and then minimizing the distances via network learning by the Euclidean error gradient as shown in Equation (10).

$$E = \frac{1}{2N} \sum_{i=1}^{N} \|\hat{y}_i - y_i\|_2^2 \tag{9}$$

$$\frac{\partial E}{\partial y} = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{10}$$

where $\hat{y}_i$ and y_i denote the output of the Euclidean Loss layer and the target outputs, respectively. Equations (9) and (10) represent the most significant difference between the Euclidean Loss layer and a softmax layer. The output of the Euclidean Loss layer is a likely result for pavement MTD in our framework.

2.3 Network learning

The goal of CNN learning is to minimize the Euclidean distance between the predicted result of pavement MTD $\hat{y}_i$ and the actual result of pavement MTD y_i for all the training data samples by adjusting the connection weights W_z of the fully connected layer, weight vectors Θ_j, and bias vectors b_j of the convolutional kernels using the backpropagation algorithm. The conventional backpropagation algorithm exhibits the disadvantage of low convergence speed. Therefore, a modified backpropagation algorithm with a damping term is utilized to overcome this disadvantage.

In our backpropagation algorithm, W_z, Θ_j, and b_j are optimized using Equations (11)–(13), respectively:

$$W_i(p+1) = W_i(p) - \alpha_z \frac{\partial E}{\partial W_i(p+1)} + \beta[W_i(p) - W_i(p-1)] \tag{11}$$

$$\Theta_j(p+1) = \Theta_j(p) - \alpha_j \frac{\partial E}{\partial \Theta_j(p+1)} + \beta[\Theta_j(p) - \Theta_j(p-1)] \tag{12}$$

$$b_j(p+1) = b_j(p) - \alpha_j \frac{\partial E}{\partial b_j(p+1)} + \beta[b_j(p) - b_j(p-1)] \tag{13}$$

where $p-1$, p, and $p+1$ denote the $p-1$th, pth, and $p+1$th training iterations, respectively. W_i denotes the connection weights of the neuron i in the fully connected layer, whereas Θ_j and b_j denote the weight and bias vectors, respectively, in convolutional kernel j. α_z and α_j are the learning rates for the fully connected layer and convolutional layer, respectively, and their range is $[0, +\infty]$. β is the damping coefficient.

Based on our backpropagation algorithm, the procedure for developing a CNN in our research is as follows:

Step 1 Assign stochastic parameters W_z, Θ_j, and b_j to the initial CNN.

Step 2 Select 1,000 stochastic training samples $\{(x_i, y_i), i = 1, \ldots, 1,000\}$ for each iteration, and then, input them into the CNN. The input is $x_i \in R^3$, and the target output is $y_i \in R$.

Step 3 Calculate the MTD of the 1,000 stochastic training samples, and then, evaluate the outputs using Equation (9).

Step 4 Adjust the parameters W_z, Θ_j, and b_j using Equations (10)–(12).

Step 5 Select another 1,000 stochastic training samples and repeat Steps 2–4 until the obtained MTD satisfies

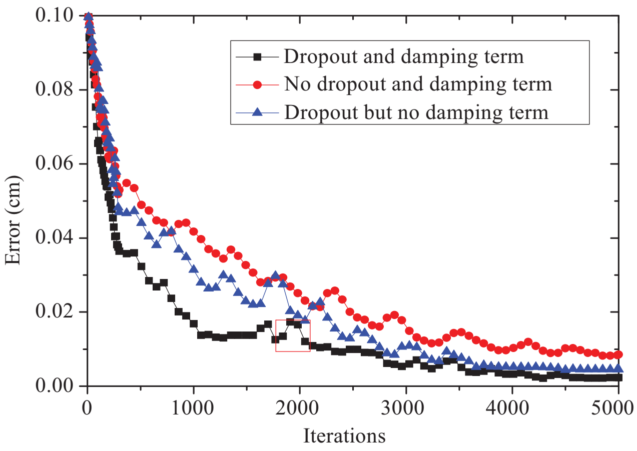


Fig. 7. Training errors for different models.

the criterion that the error calculated using Equation (9) is lower than 10^{-2} cm.

Step 6 Input the testing data set into the well-trained CNN, and calculate the error for the testing data set using Equation (9).

The above procedure was performed using Caffe on a computer equipped with an Intel® Core™ i7-6700 CPU, 8.00 GB RAM, and an NVIDIA GeForce GTX 1060 6 GB GPU.

3 CNN PERFORMANCE AND DISCUSSION

To validate the developed CNN described in the previous section, a discussion on the performance of the CNN-based method is presented in this section. The discussion includes the performance of training and testing in Section 3.1, stability analysis in Section 3.2, comparative studies in Section 3.3, and an application of the trained CNN to four highways in Shanxi, China in Section 3.4.

3.1 Training and testing

To investigate the performance of the CNN and the effects of dropout and the damping term, models with dropout and the damping term, dropout but no damping term, and no dropout or damping term were developed separately using the training data set. The results of the different models are presented in Figure 7.

Figure 7 illustrates the root mean squared error for the model using Equation (10) over 5,000 iterations during the training. With dropout and the damping term, the error fell rapidly to 0.0094 cm in the 2,330th iteration and then to 0.0024 cm in the 5,000th iteration. In the CNN with dropout but no damping term, the error fell to 0.0100 cm in the 2,820th iteration and 0.0035 cm in the 5,000th iteration. This is a slower improvement than that of the model with a damping term. The speeds of improvement of the two models reveal that the rate of convergence can be improved by utilizing a damping term. However, the damping term exerted negligible influence on the final accuracy of the MTD evaluation based on a comparison of the final errors of these two models. The reason for the positive effect on the rate of convergence is that a damping term decreases the amplitude variation of the parameters W_z, Θ_j, and b_j in each iteration. Therefore, certain inappropriate parameter adjustments, which appear as error increases during training, are reduced (see hollow red box in Figure 7).

Without either dropout or a damping term, the error fell to 0.0097 cm in the 3,871st iteration and 0.0085 cm in the 5,000th iteration, revealing that the max-pooling layer with dropout exerted a significant effect on both the rate of convergence and the accuracy of the MTD evaluation by the CNN. Therefore, the modified CNN in our research achieved higher performance for both training and testing compared to conventional CNNs. The reason for the positive effect of dropout on both the rate of convergence and accuracy of the MTD evaluation is that certain unnecessary samples from the 3D scanning data are removed by the dropout operation. These unnecessary data are likely to influence the training effectiveness or even result in overfitting. The max-pooling layer with dropout overcomes the disadvantage of conventional max-pooling, which captures only the strongest feature parameters and disregards all the other units within the pooling region.

The speed of training the CNN with dropout and damping was approximately 60 min per 100 iterations using the GPU; however, the runtime for training using the CPU was approximately 1–2 d per 100 iterations. Evidently, using a GPU significantly improves the efficiency of training.

Figure 8 presents features of receptive fields of the convolutional layer in the CNN with dropout and damping in the 5,000th iteration. The figure contains a clear area indicating that the network is properly trained. This is a positive indication because a well-trained network typically exhibits smooth and clear patterns (Cha et al., 2017a). In Figure 8, the features of the 3rd, 10th, 13th, and 14th regions can be considered as MDPs, whereas the other regions can be considered as planes that are higher or lower than the MDPs. The reason is that the average coordinates of the mapped regions of the 3rd, 10th, 13th, and 14th were very close

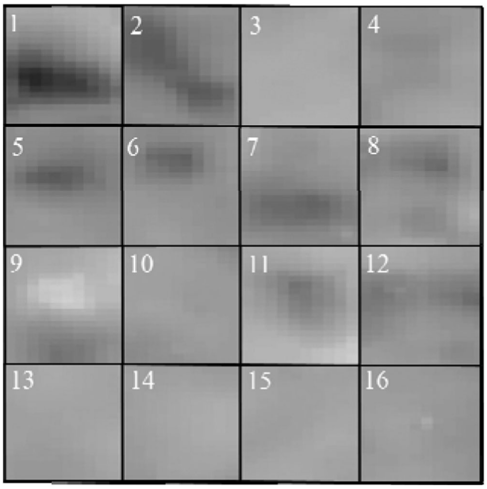


Fig. 8. Features of receptive fields of a well-trained CNN.

to the coordinates of the MDPs, whereas the average coordinates of the other mapped regions were apparently different from the coordinates of the MDPs, which we calculated based on the results of the electric sand patch tests. Therefore, the convolutional kernels in the well-trained CNN have the capability to automatically extract the features of the MDPs.

From the perspective of pavement surface texture detection, all the CNNs achieved adequate performance of MTD evaluation because the average errors between the results of the sand patch test and the outputs of the CNNs were 0.0024, 0.0035, and 0.0085 cm. For the sand patch test, the required precision was 0.01 cm based on guideline JTG H20-2007 of the Highway Performance Assessment Standards of China (JTG H20-2007, 2007). However, the average error could only reflect the average performance of the CNNs and has no significance for evaluating the stability of the CNNs for different pavement materials or different 3D scanning data with different DPIs. For example, a well-trained CNN with a reasonable average error rate is likely to achieve adequate performance for SBS modified asphalt concrete (AC-16C) albeit low performance for stone mastic asphalt (SMA-13). Therefore, an analysis based on the test results must be performed to verify the stability of our CNN.

3.2 Stability analysis

Stability analysis based on the test results was performed to further verify the performance of our CNN. The analysis consisted of two main parts: the stability of different pavement materials and stability of different DPIs.

3.2.1 Pavement materials. To verify the robustness of the CNN to different pavement materials, the test results, specifically the errors between the results of the sand patch tests and the outputs of the modified CNN in the 5,000th iteration, were divided according to the pavement materials. The texture characteristics of the four pavements were different from those of the pavements used for training, owing to the different service times. Figure 9 illustrates the absolute error distribution of testing divided according to the different pavement materials. As shown in the frequency distribution histograms, all the errors are between 0.013 and 0.034 cm regardless of the pavement type, which are highly similar and close to the electric sand patch test results. Therefore, we can conclude that these four surface materials exert negligible influence on the accuracy of our CNN. Considering that SBS modified asphalt concrete (AC-16), asphalt concrete (AC-13 and AC-16), and stone mastic asphalt (SMA-13) are the most common surface materials in China, it is reasonable to assume that our CNN could be generally utilized to simulate sand patch testing in China.

The errors between the outputs of the CNN and the results of the sand patch tests are likely to have been caused by three factors: (1) the variations in the 3D scans at the same location, (2) the variations in the sand patch tests at the same location (ASTM E965-96, 1996; JTG H20-2007, 2007), and (3) the errors of the predicted MDPs of the CNN. The variations in the 3D scans at the same location were smaller than the variations in the sand patch tests at the same location (Prowell and Hanson, 2005), particularly for open-grade pavement. The variations in the sand patch tests at the same location contributed more to the errors than the variations in the 3D scans at the same location. However, it is challenging to compare the effect of the variations in the sand patch tests and the errors of the predicted MDPs based only on the stability analysis of the pavement materials; this is further discussed in Section 3.3 based on the results of comparative studies.

3.2.2 DPIs. The accuracy of our CNN is also influenced by another important factor, namely, DPI; this is because DPI is likely to influence the quality of the input data. Therefore, a stability analysis on DPI was performed. As mentioned above, the DPI of the scans for the training and testing data generation was 100. As demonstrated in Section 3.1, the CNN achieved adequate performance using 100 DPI scan files. Next, we decreased the scanning DPI of the 3D scanner. Using this method, testing data sets of 85, 50, 30, and 25 DPI were generated. These data sets exhibit high comparability with the testing data set of 100 DPI. The testing data sets with 85, 50, 30, and 25 DPI were then imported

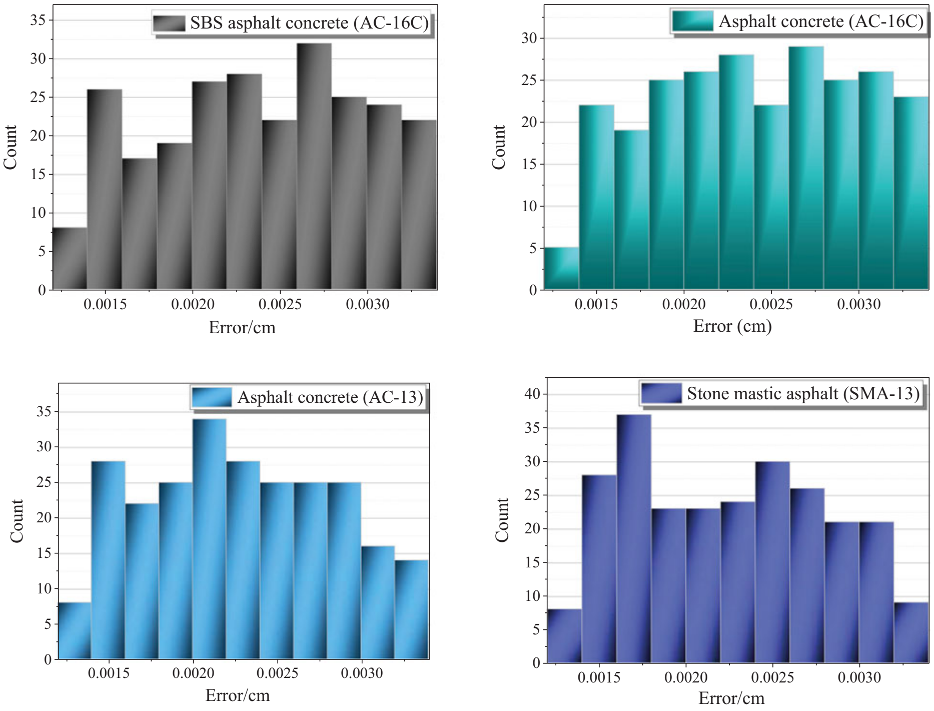


Fig. 9. Errors from different pavement materials.

Table 2
Errors of data sets with different DPIs

DPI	100	85	50	30	25
Error/cm	0.0024	0.0130	0.0156	0.0238	0.0257

into the well-trained CNN. The average errors of the data sets with the different DPIs are listed in Table 2. The average errors between the outputs of the data sets with the different DPIs and the results of the sand patch tests were calculated using Equation (9).

As presented in Table 2, the accuracy of the CNN dropped rapidly with the decrease in DPI, particularly from 100 to 85 DPI. Based on the results obtained from the data sets with the different DPIs, we can conclude that a decrease in DPI negatively affects the accuracy of our CNN. Therefore, the DPI should be guaranteed to be higher than 85 mm when using our CNN-based method. The negative effect of the decrease in DPI on the accuracy of the CNN is likely to be because as the DPI reduces, an increasing amount of information regarding the pavement surface texture is lost, which results in 3D scan files that do not accurately reflect the pavement roughness features, implying that the CNN cannot accurately compute the surface texture MTDs based on the 3D scan files.

3.3 Comparative studies

To compare the effectiveness of our CNN-based method to those of available methods, 40 3D scan data from the testing data set with various pavement materials were tested. The output errors of the CNN for the 40 3D scans were selected from the largest errors among the 1,000 total scan data to establish that the CNN-based method is superior even in certain conditions wherein the performance of the CNN is not optimal. Two methods that have been traditionally utilized in China for evaluating MTD were selected for comparison. The details of the two traditional methods are available in Hu et al. (2016) and Zhou et al. (2014). It is noteworthy that the profiles used in the traditional method proposed in Zhou et al. (2014) can be acquired from 3D scan data. In the method by Zhou et al., profiles from laser scanning technology were acquired to calculate MPD. Then, a regression formula presented in Zhou et al. (2014) was utilized to calculate MTD based on MPD. Figure 10 illustrates the process of acquiring profiles using 3D scan data. 3D scan technology is more feasible than laser technology because it can gather details regarding pavement surfaces, rather than gathering only profiles. In the method proposed in Hu et al. (2016), MTD was calculated using the sum of the height differences between each scan point and the plane average height.

Table 3
Results of three methods

	CNN-based method/cm		*Method proposed in Hu et al., 2016/cm*		*Method proposed in Zhou et al., 2014/cm*		*Sand patch test/cm*
	Results/cm	Error/%	Results/cm	Error/%	Results/cm	Error/%	Results/cm
SBS asphalt concrete (AC-16C)	0.092908	4.43	0.094908	2.37	0.084405	13.18	0.09721
	0.106740	5.15	0.106941	5.35	0.093304	8.08	0.10151
	0.104235	4.20	0.095837	4.20	0.110146	10.11	0.10004
	0.104348	5.03	0.104348	5.03	0.108660	9.37	0.09935
	0.093279	5.07	0.093279	5.07	0.092749	5.61	0.09826
	0.095218	5.41	0.106110	5.41	0.111137	10.40	0.10066
	0.093738	4.90	0.093738	4.90	0.083978	14.80	0.09857
	0.092384	6.10	0.092384	6.10	0.082553	16.09	0.09839
	0.095392	4.46	0.095392	4.46	0.093782	6.07	0.09984
	0.103867	5.42	0.093178	5.42	0.108258	9.88	0.09852
Asphalt concrete (AC-16C)	0.095438	4.41	0.095438	4.41	0.117002	17.19	0.09984
	0.102638	5.30	0.092305	5.30	0.091727	5.89	0.09747
	0.092534	5.99	0.104328	5.99	0.104891	6.56	0.09843
	0.108652	5.21	0.108652	5.21	0.113739	10.13	0.10328
	0.094151	6.08	0.106340	6.08	0.104287	4.03	0.10025
	0.098790	4.21	0.098790	4.21	0.095583	7.32	0.10313
	0.106936	4.59	0.106936	4.59	0.109256	6.85	0.10225
	0.095305	4.90	0.095305	4.90	0.091915	8.28	0.10021
	0.101250	4.41	0.101250	4.41	0.102181	5.37	0.09698
	0.091444	6.11	0.103348	6.11	0.110030	12.97	0.09740
Asphalt concrete (AC-13)	0.079008	6.38	0.076820	8.97	0.079430	5.88	0.08439
	0.091222	6.05	0.077468	9.94	0.093614	8.83	0.08602
	0.086934	5.47	0.089380	8.43	0.079046	4.10	0.08243
	0.083976	5.30	0.098547	11.13	0.097193	9.60	0.08868
	0.076471	7.13	0.074537	9.48	0.089653	8.88	0.08234
	0.094356	6.06	0.097800	9.93	0.095131	6.93	0.08897
	0.090573	6.60	0.091927	8.20	0.090062	6.00	0.08496
	0.084118	4.79	0.092643	4.86	0.097378	10.22	0.08835
	0.088269	5.48	0.090125	7.69	0.090299	7.90	0.08369
	0.077219	5.34	0.076104	6.70	0.080743	1.02	0.08157
Stone mastic asphalt (SMA-13)	0.081372	5.67	0.083643	8.62	0.082115	6.64	0.07701
	0.080145	5.78	0.083752	10.54	0.080405	6.13	0.07576
	0.068811	8.13	0.079202	5.74	0.065459	12.61	0.07490
	0.079965	5.89	0.083112	10.06	0.080104	6.08	0.07551
	0.079915	8.19	0.079217	7.24	0.065565	11.24	0.07387
	0.074623	5.50	0.071748	9.14	0.069510	11.97	0.07896
	0.078686	7.96	0.080801	10.86	0.081992	12.49	0.07289
	0.069550	5.89	0.082023	10.99	0.068925	6.73	0.07390
	0.069477	7.69	0.081355	8.09	0.082331	9.39	0.07527
	0.072178	6.48	0.084597	9.61	0.067740	12.23	0.07718

The MTD results based on these three methods are presented in Table 3. Compared to the method proposed in Hu et al. (2016), the CNN-based method achieved remarkable performance on all the four types of pavement materials, whereas the other two methods achieved suboptimal performance for a few materials. For example, the errors of the CNN-based method for SMA pavements, which are gap-graded surfaces, are in the range of 5.50–8.19%, whereas the errors of the other two methods are in the ranges of 5.74–8.19% and 6.13–12.49%. These results demonstrate that the CNN-based method exhibited higher robustness to pavement materials. Additionally, the CNN-based method can handle certain problematic surfaces, such as gap-graded surfaces. This is because the method proposed in Hu et al. (2016) assumes that the MDPs represent the planes of

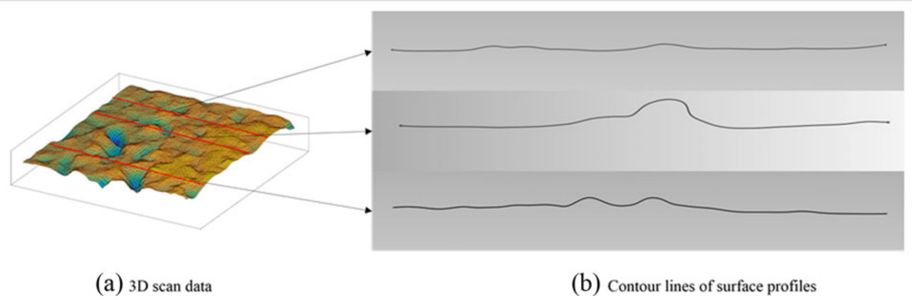


Fig. 10. Profiles from 3D scan data.

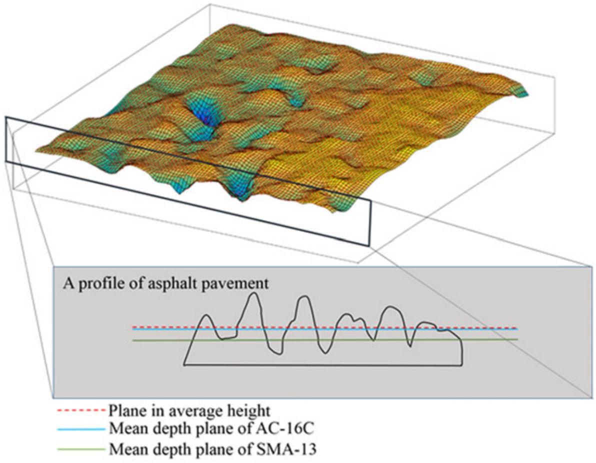

A profile of asphalt pavement
Plane in average height
Mean depth plane of AC-16C
Mean depth plane of SMA-13

Fig. 11. Profiles from 3D scan data.

average height; this is likely suitable for AC-16C pavement, although not for SMA-13 pavement because the MDP of SMA-13 is not equivalent to the plane average height, as shown in Figure 11. Therefore, the CNN-based method exhibited higher robustness to the pavement materials. The CNN-based method exhibits higher stability because the 16 kernels extract different features of the MDPs to adjust for the changes in the pavement materials during the MTD evaluation, rather than assuming the MDPs to be equivalent to the planes of average height. The property eliminated the effect of the different MDPs to a certain degree. From this comparative analysis, we can conclude that the major advantage of using a CNN is that feature extraction techniques are not required because our CNN learns features automatically when it is trained using the modified backpropagation algorithm. This advantage can save significant time compared to traditional methods that require human-assisted feature extraction. For example, to evaluate the MTD of a group of 3D scan data utilizing traditional methods, it is necessary to perform various tests to determine the MDP or even assume that the MDP is equivalent to the plane average height. Additionally, as presented in Table 3, all the errors of the CNN-based method are either + 0.005 or –0.005 cm with nothing in between. It is close to the variation of the sand patch tests (ASTM E965-96, 1996; JTG H20-2007, 2007). This indicated that the errors of the predicted MDPs of the CNN are smaller than the variation of the sand patch tests.

A comparison of the errors between the CNN-based method and the method proposed in Zhou et al. (2014) reveals that the CNN-based method achieved higher accuracy. The method proposed in Zhou et al. (2014) utilized surface profiles to calculate MPD and then evaluated MTD based on the results of MPD. The surface profiles included less information about the surface texture compared to the 3D scan data. Additionally, a few researchers have observed an unexpected relationship between MTD and MPD (Praticò and Vaiana, 2015; Miao et al., 2011). Therefore, the results provided by the method proposed in Zhou et al. (2014) are not reliable.

3.4 Application of the proposed method

To further verify the performance of the proposed CNN, the scanning data from an actual pavement, rather than only from cores, should be utilized as a test data set. Therefore, the CNN-based method was used to measure the MTD of an actual pavement following the processes described in Section 2 to establish the transferability of this method to different highways. Another four highways in Shanxi, China, which were not the same ones used to test the CNN after training, were observed. The scanning method must not damage pavement. Therefore, we scanned the pavement directly rather than extract cores. Four position markers were placed on the pavement and only the results within the boundary of the four positions were retained from scanning. The modified sand patch test introduced in Section 2.1.1 was also performed within the scan range to compare the results of the CNN outputs and sand patch tests.

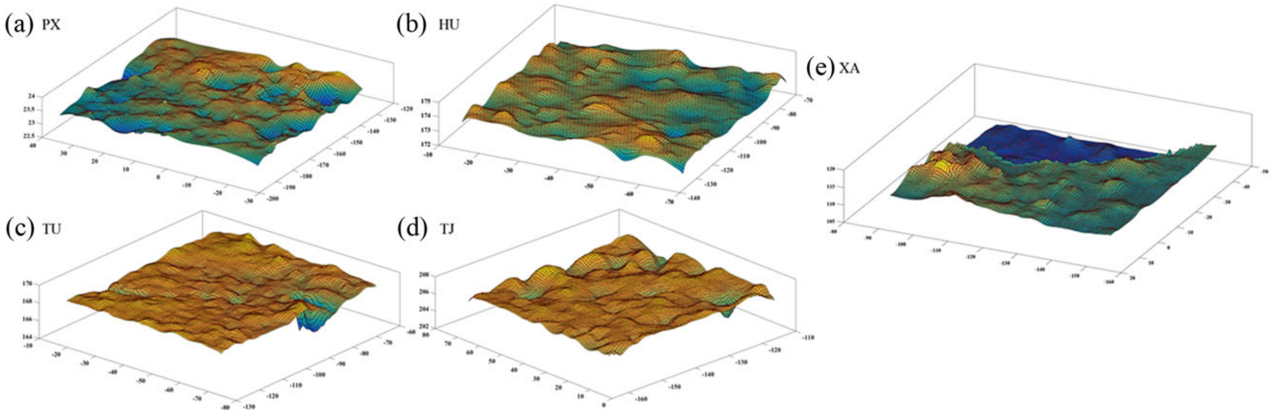


Fig. 12. Part results of 3D scan.

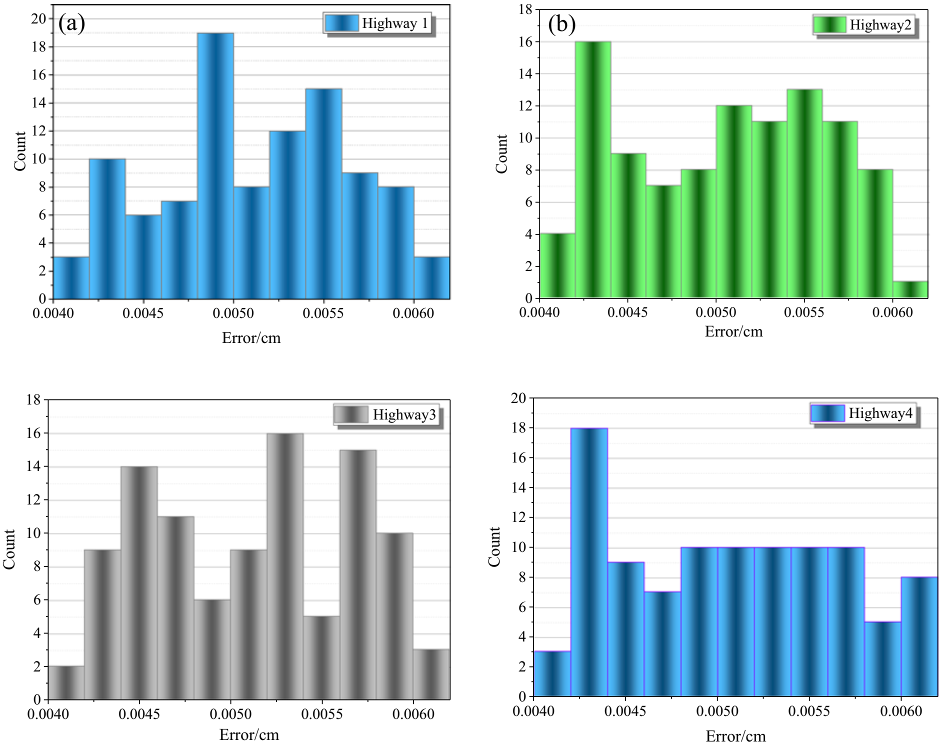


Fig. 13. Errors for different highways.

One thousand five hundred data samples were collected and used for testing. A portion of the 3D scanning results from the four highways is presented in Figure 12. The results of the 3D scanning are not influenced by the scan target, regardless of whether it is a core sample or pavement surface. Additionally, the surfaces labeled TU and XA contain a crack and hole, respectively, indicating that the 3D scanning data can be used for various problematic surfaces containing pavement defects. Therefore, the MTD outputs of our CNN are capable of considering surface damage. After scanning, the 3D scan data were imported into the well-trained CNN. The absolute errors between the outputs and measurement results were calculated. Figure 13 shows the distribution of the errors. From the results obtained, the maximum errors for the application were 0.006119, 0.006094, 0.0061151, and 0.006118 cm for the four highways. Although the maximum errors of the application are higher than the maximum errors from testing, they still satisfy the requirements for highway engineering detection (JTG H20-2007, 2007). The errors of the surfaces labeled TU and XA, containing a crack and a hole pit, respectively, were 0.0060217 and 0.005891 cm, respectively. This indicated that the CNN output errors of the problematic pavements were less than 0.01 cm, which was acceptable (JTG H20-2007, 2007). This demonstrates that the configured CNN is capable of addressing more complex problems, including certain conditions not included in the training data set. However, comparing with the results shown in Figure 13, the errors of the problematic pavements were larger than the errors of the pavements with no defect. The maximum errors also reveal that the pavement materials exert negligible impact on the accuracy of the CNN. All the results demonstrate that the modified CNN can satisfy the requirements for MTD evaluation on different highways.

Based on the results obtained from the modified CNNs for MTD evaluation, several recommendations can be made for highway surface detection. For example, detection efficiency can be improved by saving time by replacing the sand patch test with the CNN-based method. Additionally, the CNN and 3D scan technology can be utilized in certain conditions that are not suitable for the sand patch test, such as ongoing pavement construction. During pavement construction, the pavement MTD should be evaluated in real time to guarantee proper macrosurface texture, drainage performance, and skid resistance. However, the sand patch test, which requires significant time, cannot be performed immediately following the pavement construction because of the pavement temperature. The CNN and 3D scan technology can be utilized during paving, immediately after fresh pavement, or with semirigid bases without significant interference, and achieve results close to those of the sand patch test.

4 CONCLUSIONS

In this study, a modified CNN for pavement MTD analysis (in batches) using 3D scan data was applied. The performance of the CNN-based method was discussed in detail. The following conclusions can be drawn from the presented results and discussion:

1. The modified CNN was designed to evaluate pavement MTD based on 3D scan data. During training, the modified CNN learned features without human assistance. During testing, the CNN simulated the sand patch test with an average error of 0.0024 cm with remarkably high stability for different pavement materials. It is reasonable to assume that the CNN can be generally utilized to simulate sand patch tests in China.
2. A damping term can improve the rates of convergence albeit exerts negligible influence on the final accuracies of the MTD evaluation for decreasing the amplitude variations in the parameters W_z, Θ_j, and b_j in each iteration. Utilization of dropout significantly affected both the rate of convergence and accuracy of MTD evaluation because it removes unnecessary samples from the 3D scanning data. Additionally, a GPU can significantly improve the efficiency of training.
3. The accuracy of the CNN dropped rapidly with a decrease in DPI, particularly from 100 to 85 DPI. Based on the results obtained from data sets with different DPIs, we can conclude that a decrease in DPI negatively affects the accuracy of the CNN. It is necessary to guarantee that the DPI of a 3D scan is greater than 50 DPI.
4. A comparison of the CNN-based method with available methods reveals that the CNN-based method achieved higher performance in terms of robustness to pavement materials and the accuracy of the final results.
5. The application of the proposed method demonstrated that the CNN-based method could satisfy the requirements for MTD evaluation in highway engineering detection, although the maximum error of the application was larger than the maximum errors from testing. A few recommendations regarding highway surface detection were made based on the results obtained from the CNN.

5 FUTURE RESEARCH

Future research should focus on perfecting the method to gather target data because the method proposed in Section 2.1.2 for gathering target data was straightforwardly affected by the researchers' operation. An improved method that is not straightforwardly affected by researchers' operation should be developed.

REFERENCES

Adeli, H. (2001), Neural networks in civil engineering: 1989–2000, *Computer-Aided Civil and Infrastructure Engineering*, **16**(2), 126–42.

Alhasan, A., White, D. J. & De Brabanter, K. (2016), Wavelet filter design for pavement roughness analysis, *Computer-Aided Civil and Infrastructure Engineering*, **31**(12), 907–20.

Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Aldwaik, M. & Adeli, H. (2006), Neurocomputing in civil infrastructure, *Scientia Iranica-A*, **23**(6), 2417–28.

Ansary, T.-F., Daoudi, M. & Vandeborre, J.-P. (2007), A Bayesian 3D search engine using adaptive views clustering, *IEEE Transactions on Multimedia*, **9**(1), 78–88.

Arevalo, J., González, F. A., Ramos-Pollán, R., Oliveira, J. L. & Guevara Lopez, M. A. (2016), Representation learning for mammography mass lesion classification with convolutional neural networks, *Computer Methods and Programs in Biomedicine*, **127**, 248–57.

ASTM E965-96 (1996), *Standard Test Method for Measuring Pavement Macrotexture Depth Using a Volumetric Technique*, American Society for Testing and Materials, Washington DC.

Barat, C. & Ducottet, C. (2016), String representations and distances in deep convolutional neural networks for image classification, *Pattern Recognition*, **54**, 104–15.

Bendtsen, H. (2004), *Rolling Resistance, Fuel Consumption: A Literature Review*, Road Directorate, Danish Road Institute, ISSN: 1395–5530.

Cecotti, H. & Graser, A. (2011), Convolutional neural networks for P300 detection with application to brain–computer interfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(3), 433–45.

Čelko, J., Kováč, M. & Kotek, P. (2016), Analysis of the pavement surface texture by 3D scanner, *Transportation Research Procedia*, **14**, 2994–3003.

Cha, Y. J., Choi, W. & Büyüköztürk, O. (2017a), Deep learning-based crack damage detection using convolutional neural networks, *Computer-Aided Civil and Infrastructure Engineering*, **32**(5), 361–78.

Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S. & Büyüköztürk, O. (2017b), Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, *Computer-Aided Civil and Infrastructure Engineering*, https://doi.org/10.1111/mice.12334.

EN 13036-1 (2010), *Road and Airfield Surface Characteristics—Test Methods—Part 1: Measurement of Pavement Surface Macrotexture Depth Using a Volumetric Patch Technique*, European Norm, England, English.

Fan, J., Xu, W., Wu, Y. & Gong, Y. (2010), Human tracking using convolutional neural networks, *IEEE Transactions on Neural Networks*, **21**(10), 1610–23.

Fu, M., Xu, P., Li, X., Liu, Q., Ye, M. & Zhu, C. (2015), Fast crowd density estimation with convolutional neural networks, *Engineering Applications of Artificial Intelligence*, **43**, 81–88.

Garbowski, T. & Gajewski, T. (2017), Semi-automatic inspection tool of pavement condition from three-dimensional profile scans, *Procedia Engineering*, **172**, 310–18.

Gendy, A. E., Shalaby, A., Saleh, M. & Flintsch, G. W. (2011), Stereo-vision applications to reconstruct the 3D texture of pavement surface, *International Journal of Pavement Engineering*, **12**(3), 263–73.

Hu, L., Yun, D., Liu, Z., Du, S., Zhang, Z. & Bao, Y. (2016), Effect of three-dimensional macrotexture characteristics on dynamic frictional coefficient of asphalt pavement surface, *Construction and Building Materials*, **126**, 720–29.

Ijjina, E. P. & Chalavadi, K. M. (2016), Human action recognition using genetic algorithms and convolutional neural networks, *Pattern Recognition*, **59**, 199–212.

ISO 13473-1 (1997), *Characterization of Pavement Texture by Use of Surface Profiles, Part 1: Determination of Mean Profile Depth*, International Organization for Standardization, England, English.

Jain, V., Murray, J. F., Roth, F., Turaga, S. C., Zhigulin, V., Briggman, K. L., Helmstaedter, M. N., Denk, W. & Seung, H. S. (2007), Supervised learning of image restoration with convolutional networks, in *IEEE 11th International Conference on Computer Vision*, 1–8.

Jain, V. & Seung, H. S. (2008), Natural image denoising with convolutional networks, in *Neural Information Processing Systems*, 769–76.

JTG H20-2007 (2007), *Highway Performance Assessment Standards*, Ministry of Transport of the People's Republic of China, Beijing, China.

Kanafi, M. M., Kuosmanen, A., Pellinen, T. & Tuononen, A. J. (2014), Macro- and micro-texture evolution of road pavements and correlation with friction, *International Journal of Pavement Engineering*, **16**(2), 168–79.

Koziarski, M. & Cyganek, B. (2017), Image recognition with deep neural networks in presence of noise—dealing with and taking advantage of distortions, *Integrated Computer-Aided Engineering*, **24**(4), 337–50.

LeCun, Y. (1989), Generalization and network design strategies, in R. Pfeifer, Z. Schreter, F. Fogelman, and L. Steels (eds.), *Connectionism in Perspective*, Elsevier, Zurich, Switzerland, pp. 143–55.

LeCun, Y. & Bengio, Y. (1995), Convolutional networks for images, speech, and time series, in M. A. Arbib (ed.), *The Hand book of Brain Theory and Neural Networks*, MIT Press, p. 3361.

Leng, B., Guo, S., Zhang, X. & Xiong, Z. (2015), 3D object retrieval with stacked local convolutional autoencoder, *Signal Processing*, **112**, 119–28.

Leng, B. & Xiong, Z. (2011), Modelseek: an effective 3D model retrieval system, *Multimedia Tools and Applications*, **51**(3), 935–62.

Leng, B., Xiong, Z. & Fu, X. (2010), A 3D shape retrieval framework for 3D smart cities, *Frontiers of Computer Science in China*, **4**(3), 394–404.

Li, Q., Jin, Z., Wang, C. & Zeng, D. D. (2016), Mining opinion summarizations using convolutional neural networks in Chinese microblogging systems, *Knowledge-Based Systems*, **107**, 289–300.

Liao, T. Y. (2017), On-line vehicle routing problems for carbon emissions reduction, *Computer-Aided Civil and Infrastructure Engineering*, **32**(12), 1047–63.

Lin, Y. Z., Nie, Z. H. & Ma, H. W. (2017), Structural damage detection with automatic feature-extraction through deep learning, *Computer-Aided Civil and Infrastructure Engineering*, **32**(12), 1025–46.

Liu, Z., Zhang, J. & Liu, L. (2016), Upright orientation of 3D shapes with convolutional networks, *Graphical Models*, **85**(5), 22–29.

Miao, Y., Cao, D. W. & Lin, Q. Q. (2011), Relationship between surface macrotexture and skid resistance of asphalt pavement, *Journal of Beijing University of Technology*, **4**(37), 547–53.

Mosa, A. M., Rahmat, R. A. O. K., Ismail, A. & Taha, M. R. (2013), Expert system to control construction problems in flexible pavement, *Computer-Aided Civil and Infrastructure Engineering*, **28**(4), 307–23.

Oijer, F. & Edlund, S. (2004), Identification of transient road obstacle distributions and their impact on vehicle durability and driver comfort, *Vehicle System Dynamic*, **41**, 744–53.

Ortega-Zamorano, F., Jerez, J. M., Gómez, I. & Franco, L. (2017), Layer multiplexing FPGA implementation for deep back-propagation learning, *Integrated Computer-Aided Engineering*, **24**(2), 171–85.

Osada, R., Funkhouser, T., Chazelle, B. & Dobkin, D. P. (2002), Shape distributions, *ACM Transactions on Graphics*, **21**(4), 807–32.

Passalis, G., Theoharis, T & Kakadiaris, I. A. (2006), PTK: a novel depth buffer-based shape descriptor for three-dimensional object retrieval, *The Visual Computer*, **23**(1), 5–14.

Patane, G., Spagnuolo, M. & Falcidieno, B. (2009), A minimal contouring approach to the computation of the Reeb graph, *IEEE Transactions on Visualization and Computer Graphics*, **15**(4), 583–95.

Praticò, F. G. & Vaiana, R. (2015), A study on the relationship between mean texture depth and mean profile depth of asphalt pavements, *Construction and Building Materials*, **101**, 72–79.

Prowell, B. & Hanson, D. (2005), Evaluation of circular texture meter for measuring surface texture of pavements, *Transportation Research Record*, **1929**, https://doi.org/10.3141/1929-11.

Rafiei, M. H. & Adeli, H. (2017), A novel machine learning based algorithm to detect damage in high-rise building structures, *The Structural Design of Tall and Special Buildings*, **26**, 18, https://doi.org/10.1002/tal.1400.

Rafiei, M. H., Khushefati, W. H., Demirboga, R. & Adeli, H. (2017), Supervised deep restricted Boltzmann machine for estimation of concrete compressive strength, *ACI Materials Journal*, **114**(2), 237–44.

Shi, B., Bai, X. & Yao, C. (2016), Script identification in the wild via discriminative convolutional neural network, *Pattern Recognition*, **52**, 448–58.

Slabej, M. & Kotek, P. (2014), Using of 3D road surface model for monitoring of transverse unevenness and skid resistance, *Procedia Engineering*, **91**, 475–80.

Sohaney, R. C. & Rasmussen, R. O. (2013), Pavement texture evaluation and relationships to rolling resistance at MnROAD, Minnesota Department of Transportation, Research Services, Final Report: MN/RC 2013–16. Online version is available at: http://www.dot.state.mn.us/research/TS/2013/201316.pdf, accessed June 2013.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014), Dropout: a simple way to prevent neural networks from overfitting, *Journal of Machine Learning Research*, **15**(1), 1929–58.

Tong, Z., Gao, J., Han, Z. & Wang, Z. (2018), Recognition of asphalt pavement crack length using deep convolutional neural networks, *Road Materials and Pavement Design*, **19**(6), 1334–49.

Tong, Z., Gao, J. & Zhang, H. (2017), Recognition, location, measurement and 3D reconstruction of concealed cracks using convolutional neural networks, *Construction and Building Materials*, **146**, 775–87.

Torbruegge, S. & Wies, B. (2015), Characterization of pavement texture by means of height difference correlation and relation to wet skid resistance, *Journal of Traffic Transportation (English Edited)*, **2**(2), 59–67.

Turaga, S. C., Murray, J. F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W. & Seung, H. S. (2010), Convolutional networks can learn to generate affinity graphs for image segmentation, *Neural Computation*, **22**(2), 511–38.

Vilaca, J. L., Fonseca, J. C., Pinho, A. C. & Freitas, E. F. (2010), 3D surface profile equipment for the characterization of the pavement texture-TexScan, *Mechatronics*, **20**(6), 674–85.

Wang, W. F., Yan, X. P., Chu, X. M., Xie, L. & Ma, Y. L. (2010), Feature description and its solution of asphalt road pavement micro-topography based on fractal theory, *Journal of Jilin University (Engineering and Technology Edition)*, **40**(6), 1538–42.

Wen, J. (2009), *Study on Evaluating Texture Depth of Asphalt Pavement with Digital Technology*. Chang'an University, Xi'an, China.

Wu, H. & Gu, X. (2015), Towards dropout training for convolutional neural networks, *Neural Networks*, **71**, 1–10.

Xu, J., Luo, X., Wang, G., Gilmore, H. & Madabhushi, A. (2016a), A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images, *Neurocomputing*, **191**, 214–23.

Xu, T., Zhu, F., Wong, E. K. & Fang, Y. (2016b), Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition, *Image and Vision Computing*, **55**(2), 127–37.

Yuan, Y. & Lai, A.-X. (2010), Pavement surface texture measurement with fractal theory and digital image, *Journal of Wuhan University of Technology*, **32**(18), 45–48, 164.

Zaniewski, J. P., Butler, B., Cunningham, G., Elkins, G. & Paggi, M. (1982), *Vehicle Operating Costs, Fuel Consumption, and Pavement Type and Condition Factors*, Final Report, Sep 79–Oct 81, Texas Research and Development Foundation, Austin.

Zeinalia, Y. & Story, B. (2017), Competitive probabilistic neural network, *Integrated Computer-Aided Engineering*, **24**(2), 105–18.

Zhou, X. L., Jiang, N. D., Xiao, W. X., Ran, M. P. & Xie, X. F. (2014), Measurement method for mean texture depth of asphalt pavement based on laser vision, *China Journal of Highway and Transport*, **27**(3), 11–16.

Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X. & Liu, W. (2016), Traffic sign detection and recognition using fully convolutional network guided proposals, *Neurocomputing*, **214**, 758–66.