Research Article

A Depth Camera-Based Intelligent Method for Identifying and Quantifying Pavement Diseases

Hao Bai,1,2 Xiangyu Hu,3 Fei Chen,4,5 Zhiyong Liao,1,5 Kai Li,6 Guangjiong Ran,7 and Fengni Wei8

1Sichuan Expressway Construction and Development Group Co., Ltd., Chengdu, China
2National Engineering Research Center for Geological Disaster Prevention Technology in Land Transportation, Ltd., Chengdu, China
3Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
4Sichuan Intelligent High-Speed Technology Co., Ltd., Chengdu, China
5Shudao Investment Group Co., Ltd., Chengdu, China
6Sichuan Intelligent Highway Technology Co., Ltd., China
7Chang’an University, Xi’an, China
8The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Correspondence should be addressed to Xiangyu Hu; huxiangyu233@outlook.com

Received 29 December 2021; Revised 15 March 2022; Accepted 16 March 2022; Published 11 April 2022

Academic Editor: Thanh-Canh Huynh

Copyright © 2022 Hao Bai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this study, a depth camera-based intelligence method is proposed. First, road damage images are collected and transformed into a training set. Then training, defect detection, defect extraction, and classification are performed. In addition, a YOLOv5 is used to create, train, validate, and test the label database. The method does not require a predetermined distance between the measurement target and the sensor; can be applied to moving scenes; and is important for the detection, classification, and quantification of pavement diseases. The results show that the sensor can achieve plane fitting at investigated working distances by means of a deep learning network. In addition, two pavement examples show that the detection method can save a lot of manpower and improve the detection efficiency with certain accuracy.

1. Introduction

As an important mode of transportation, roads play a significant role in the political, economic, and military development of our country, and also in the improvement in people’s living standards. By the end of 2018, the road network density reached 50.13 km for each 100 square km, with an annual increase of about 5000 km. However, road surfaces are easily damaged due to external factors such as overloading and overuse [1–4]. If the treatment of road surface diseases is not timely enough, it will not only reduce the life span of the road and lead to more serious diseases, but also cause huge post-maintenance costs and even cause traffic accidents, which can jeopardize the lives of citizens [5–7]. Therefore, the road surface condition assessments constitute a necessary mission. Traditional manual methods for detecting road pavement diseases are inefficient and error-prone. Thus, many methods based on image processing and machine learning have been proposed by researchers for pavement disease detection.

Laser scanning can be used to obtain road pavement disease maps and perform disease detection and analysis. Aleskey and Funkhouser [8] analyzed the 3D discrete data of superimposed scans at different time points and successfully extracted the characteristics of road surface diseases. Guan et al. [9] scanned the images of road pavement diseases and segmented the images through morphological algorithms to obtain disease features. Furthermore, in addition to the scanned data, traditional image processing algorithms are used. Traditional image processing algorithms can be
roughly divided into three categories, namely pure image classification, block-level classification, and pixel-level segmentation. The collection of pavement data through special equipment and recent advances in high-performance computing have paved the way for crack detection algorithms. For example, Bo et al. [10] used a morphological filtering algorithm to detect road surface diseases, and [11] proposed a local binary algorithm. However, recently, machine learning algorithms with higher accuracy and better feature extraction capabilities have been applied to pavement disease detection. Shi et al. [12] used an algorithm based on structured random forest to extract the pavement disease features, Li [13] used the LeNet-5 network structure, and Zhang [14] used YOLOv5 for pavement disease detection, and the classification effect was good. Guo et al. [15] used the support vector machine (SVM) method to detect the surface defects of steel plates. Cha et al. [16] proposed a convolutional neural network (CNN) model to detect cracks. Snively [17] fused convolutional features in encoder and decoder networks and built DeepCrack based on SegNet architecture for crack detection. Song et al. [18] proposed a new trainable convolutional network to automatically detect cracks in complex environments using multi-scale feature attention networks. Jahanshahi et al. [19] proposed a state-of-the-art pixel-level crack detection architecture called CrackU-net, which is characterized by leveraging advanced deep convolutional neural network technology (DCNN).

Among various types of road pavement diseases (such as cracks, potholes, ruts, and fracturing), potholes and cracks are two typical diseases of the road surface, which seriously affect the smoothness of the road surface and the driving comfort. Among them, due to the visual uniqueness of potholes, machine learning methods can quickly detect road potholes based on two-dimensional images. However, two-dimensional images can only detect and classify lesions and cannot quantify the volume damage of potholes. Therefore, some scholars adopt the method of structure from motion (SfM) to study the condition of asphalt pavement with 3D depth images [17]. Torok et al. [18] proposed a SfM-based data acquisition method to collect post-disaster scene data from a machine platform for 3D reconstruction, damage identification, and geometric data recording. A sensor system consisting of RGB-D sensors and depth sensors is also used to extract 3D point clouds to detect and quantify potholes in road surfaces [19].

However, these methods are manual or semiautomatic for quantifying known diseases. Although the data collection process can be carried out using systems capable of detecting data, disease assessment is still a manual process done by trained assessors. Due to the subjectivity of the detection and classification process, different assessors may interpret defects differently. None of these methods can independently identify potholes or other types of volume damage.

Based on these factors, an intelligent method for detecting, classifying, and quantifying pavement diseases is proposed. Based on a Microsoft Kinect V2 camera, the 2D images are first trained for defect detection, defect extraction, and disease classification for pavement diseases. Due to the lightweight property and suitable for large-scale training, through depth sensors, the YOLOv5 algorithm is employed to identify the road potholes and then the identified potholes are quantified. The proposed method does not require presetting the distance between the measurement target and the sensor and can be applied in mobile scenarios.

2. Research Method

Based on the depth camera Kinect V2, the method proposed in this study can automatically detect and locate pavement diseases. As shown in Figure 1, the specific process is divided into: (1) image collection for color images; (2) depth information collection based on the depth sensor which include an IR camera and an IR projector; (3) defect detection and extraction based on color images, and disease classification based on ResNet networks; and (4) disease segmentation and quantification using YOLOv5 based on deep point cloud data. Note that the defect detection and extraction part is inspired by the research of [20]; by anomaly detection based on the cropped blocks of small images, it is flexible to control the error tolerance of the detection to save the computational costs.

Four types of diseases including crack, pothole, rut, and fracturing are explored in this study, as shown in Table 1. To avoid redundancy, the results of common diseases of crack and pothole are shown in the main content, while the results of rut and fracturing are listed in appendices.

2.1. Two-Dimensional Image Disease Classification

2.1.1. Two-Dimensional Image Dataset. In this article, 200 images were taken with Microsoft Kinect V2 for pavement diseases and stored at 1920 × 1080 pixel resolutions. In appropriate cases, a tripod and a stabilizer are required to ensure image quality. In order to eliminate the effects of environmental changes in different road pavements, various shooting angles, light, and distances are needed to obtain road pavement disease maps, thus augmenting the size of the training dataset and improving the ability of the model generalization. Due to the limitation of GPU memory, the complete high-resolution image cannot be used directly as input. Therefore, the images need to be cropped into small blocks, and the road pavement diseases are detected and classified block by block using a deep learning model. In this study, the small blocks are cropped into 300 × 300 pixels, and Figure 2 shows the examples of blocks (cropped images) for different categories of road pavement diseases.

2.1.2. Detection Procedure. Figure 3 illustrates the architecture of the road damage classification used in this study. The process has three core component steps, namely defect detection, defect extraction, and pavement disease classification. First, defect detection is implemented to produce defect maps. According to the defect map, suspicious pavement diseases can be extracted and undamaged areas can be filtered. Afterwards, the extracted plots are classified as corresponding disease categories.
Figure 1: The flow diagram of the algorithm framework.

Table 1: Types of road diseases.

| Type    | Description                                                                 | Example |
|---------|-----------------------------------------------------------------------------|---------|
| Crack   | The cracks on the road surface which are caused by vehicle load, environmental or construction factors, etc. | ![Example](image1.png) |
| Pothole | The road surface is damaged into potholes with a depth of more than 20 mm and an area of more than 0.04 m². | ![Example](image2.png) |
| Rut     | The longitudinal strip grooves along the wheel tracks on the road surface, with a depth of more than 15 mm. | ![Example](image3.png) |
| Fracturing | The massive irregular cracks which have width more than 3 mm, and most of the distance is less than 100 mm, and the total area is more than 1 m². | ![Example](image4.png) |

Figure 2: Cropped images of road pavement diseases: (a) normal road; (b) cracks; (c) potholes.
2.1.3. Defect Detection. In the inspection process, defect detection allows for fast and reliable detection of pavement diseases from a large set of image data. In this study, pavement diseases are classified as anomalous, while normal (undamaged) pavement and other objects found in pavement are classified as normal. In this study, the defect detector is trained by constructing a convolutional autoencoder, which is a reconstruction deep learning model. In existing studies, similar defect detection methods are already applied in many fields, for example, textile surfaces, video surveillance, and detection of anomalous temperature regions from thermal images [21–23]. In addition, some deep learning models such as variational autoencoders (VAEs) and generative adversarial networks (GANs) can also be used for defect detection. The extensions have been widely used in medical imaging and aviation safety inspections. However, these methods have limitations. For deep generative models, pattern collapse often occurs during training process. In addition, the generators are only able to generate a limited variety of samples, so classifiers may not be able to distinguish between normal and abnormal samples. Hendrycks et al. [24] claimed the probability distributions learned by deep learning models may not be able to describe the ground truth of the training data. Therefore, based on what Chow et al [20] proposed, this study decided to apply convolutional autoencoders for defect detection of pavement diseases. The main features are described as follows.

A convolutional autoencoder is essentially a structure formed by encoder module and decoder module [25]. The encoder takes charge for learning and extracting specific features of the input data, and then the decoder maps the features backwards to reestablish the input data. The feature dimension obtained after encoding (also called the bottleneck) is usually much smaller than the input dimension to prevent direct replication by the convolutional autoencoder. Using a large number of normal images, the convolutional autoencoder is trained to reestablish the normal instances. Since no defective features are learned in advance, the reconstruction has poor performance of pavement diseases and huge errors.

In the encoder, the input data are fed into a continuous convolutional block and the number of channels is twice over at each down-sampling module to enhance the coded features. Before a bottleneck occurs, the feature map is spread and down-sampled for consistency among features into a fully connected layer. On the contrary, the decoder is the reverse operation, performing progressive up-sampling to reestablish the original data. The hyperbolic tangent activation function is introduced in the last layer to compress the value of each neuron in the range of $-1$ and $1$.

The squared difference between the ground truth and the prediction is calculated to evaluate the reconstruction quality of a pixel. This error is called the reconstruction error $e$:

$$e = \frac{1}{c} \sum_{i=1}^{c} \left[ p(x, y) - \tilde{p}(x, y) \right]^2,$$

where $p$ and $\tilde{p}$ are the input pixels and the reestablished input pixels positioned at $x$-$y$ row and column, respectively; $c$ is the channel of the image in color. Since the input and output values of pixels are in the interval of $[-1, 1]$, the $e$ has the maximum value of 12.

2.1.4. Defect Extraction. After obtaining the defect map, the next is to segment and extract the pavement diseases in the blocks according to the defect score. For high value of the threshold, the segmentation may be insufficient, and the extracted diseases may be incomplete and inconsistent with the ground truth. Nevertheless, if the threshold is too low, it may lead to over-segmentation, which wastes computational resources and time.
The method of randomly selecting defect scores as thresholds is ineffective since the properties of pavement diseases vary greatly at different test locations. In this study, multiple thresholds are calculated based on defect scores to facilitate defect extraction of suspected pavement diseases. A combination of local threshold (according to an individual image, \( T_L \)) and global threshold (according to the whole dataset, \( T_G \)) is used to extract diseased blocks and filter normal pavement images. In the case of normal pavement images, only a few blocks can be extracted regardless of the defect score taken by local threshold. Therefore, global threshold is introduced to avoid unnecessary information affecting the subsequent data analysis.

According to Chow et al. [20], the value of the global threshold \( T_G \) is set to a default value of 0.5 as a general case. The defect scores of all pixels of the image are first sorted, and then the defect scores corresponding to the required percentile are selected as follows:

\[
A_{P_G} = \frac{P_G}{100} \times \sum_{i=1}^{N} n_i,
\]

where \( A_{P_G} \) represents the defect score of percentile \( P_G \), \( n_i \) is the total count of all the pixels in image \( i \), and \( N \) is the total count of all the images. If \( A_{P_G} \) surpasses the threshold \( T_G \), the value of \( T_G \) is updated. Next, \( T_L \) is determined from the defect scores as follows:

\[
T_L = \frac{P_L}{100} \times n,
\]

where \( P_L \) represents the selected percentage used to calculate the \( T_L \) and \( n \) represents the total number of pixels in the image. Then, to decrease the threshold and ensure the best trade-off between the number of obtained blocks and blocks that actually have disease, the final threshold \( T \) is determined by comparing \( T_L \) with \( T_G \) by a reduction factor \( \alpha \) (0 < \( \alpha \) < 1). The process of determining the threshold \( T \) is outlined below:

1. Set the value of \( T_G \) to 0.5.
2. Calculate \( A_{P_G} \) based on the selected percentile \( P_G \).
3. If \( A_{P_G} \) surpass the default threshold, update the \( T_G \) value.
4. Calculate \( T_L \) based on the selected percentile \( P_L \).
5. Determine if \( T_L > \alpha T_G \). If the equation holds, select \( T_L \) as the threshold \( T \), otherwise, select \( T_G \) as the threshold \( T \).

When the value of the threshold \( T \) is determined, the image is divided into square blocks of equal size. If any defect score of a pixel in the block is larger than \( T \), the corresponding block is extracted for pavement disease classification.

### 2.2. In-Depth Image Disease Classification

#### 2.2.1. RGB-D Sensors

Microsoft Kinect V2 is a depth camera which has RGB image data with the resolution of 1920×1080 pixels, while the point cloud data have the resolution of 512×424 pixels. The minimum and maximum depth distance ranges from 0.5 m to 4.5 m. The time of flight (ToF) concept can be implemented using depth camera with an infrared emitter and an infrared sensor. During the working process, the IR emitter emits IR light to the object, then the light rebounded from the surface and the IR sensor receives the reflected IR light. Since the IR sensor and the IR emitter are manually placed and the distance is known, the 3D coordinates of each sensor pixel can be obtained from the time the IR light propagates from the emitter to the sensor.

#### 2.2.2. Data Point Cloud Acquisition

Microsoft Kinect V2 has interface with MATLAB and can obtain the depth data. Then, the depth camera produced a local 3D grid of coordinate points with the center pixel of the sensor as the origin. The internal algorithm of the sensor converts non-numerical values to depth pixels \((x, y, z)\). From this, the coordinates of different points can be distinguished. Note that due to lens distortion, the error rate of the values at the depth frame boundaries is larger. Yuan et al. [26] proposed that this problem can be avoided by using the image at the central area of the depth sensor with a certain size of 300×300 pixels.

#### 2.2.3. YOLOv5-Based Pavement Pothole Detection

Generally, the depth image has some noises, and the median filter can be used to reduce noises. The pixels of the map can be replaced by the median filter with the neighboring pixels, ensuring that the de-noised depth map only contains value from the existing original dataset.

To detect and locate various pavement potholes, a YOLOv5 detection [27] method is used. The method requires only images from the RGB-D sensor. YOLOv5 is adopted to detect pavement potholes. Compared to the last version of YOLOv4 [28], YOLOv5 saves nearly 90% weight file. Combined some state-of-the-art techniques of computer vision field, the parameter size of YOLOv5 is only 27 MB, much smaller than 244 MB of YOLOv4 using Darknet architecture. Therefore, YOLOv5 is suitable for embedding into devices which aims to apply real-time analysis. The structure of YOLOv5 comprised of three blocks, namely Backbone; Neck; and Head, as shown in Figure 4. The Backbone works as a feature exaction part which contains several calculation modules to extract features from the input images. Then the features are given to the object detector which has different arrangement of the modules (including Detection Neck and Detection Head). The Neck aggregates the features from the Backbone and send to the Detection Head. Finally, the bounding boxes having coordinates of the corners of the objects and the predicted precision rate are generated.

#### 2.3. Roadway Reference Plane

To quantify the pothole volume, the roadway reference plane should be determined first. The Random Sample Consensus (RANSAC) algorithm is employed for roadway plane fitting [19]. This algorithm obtains random points based on predefined parameters, for example, vector of reference, threshold of distance, and angular distance between the normal vector of the point and
the reference vector, and thus fits the plane according to the threshold until the similar data points reach their maximum value. In addition, the surface of the bounding box generated by YOLOv5 can also attribute to fit the plane in the RANSAC algorithm.

In general case, the vector of reference with a maximum angular distance of 5° parallel to the normal direction of the sensor is used. This eliminates the limit to place the sensor exactly parallel to the plane. The threshold of distance is set as 5 mm for the RANSAC algorithm. Data points beyond the threshold are considered as outliers.

The existing researches [10, 11, 16, 18] usually identify road distress according to fixed initial parameters obtained by external measurements. However, the approach used is not feasible for automatic applications and the method ignores the possibility of multiple surfaces. The proposed method can fit a plane that is not restricted by the distance between the sensor and the target, on condition that the sensor is placed within the covering distance of the device. The method relies on the sensor output and does not require any other manual measurements.

2.4. Roadway Pothole Segmentation. The 3D depth point clouds and anomalies are analyzed, and values that lie within the range of anomalies in same plane are identified as potholes. Meanwhile, the process filters out the potential NaN values in the depth data.

To individually quantify multiple potholes within the same plane, the hierarchical clustering algorithm is used to partition the defects into individual potholes. The Euclidean distance $d$ for each pair of data points $P_1, P_2$ in the input data is first calculated.

$$d(P_1, P_2) = \sqrt{\sum_i (P_{1i} - P_{2i})^2},$$

where $i$ corresponds to the reference coordinates of each point.

Then the two pairs of points closely next to each other are merged into two clusters. In addition, the clusters are continuously merged with each other to generate larger clusters. Note that the number of cluster groups is preset, and each number is considered as a feature for the corresponding depth points.

In this study, the number of clusters is preset as 10, which makes it possible to segment up to 10 individual potholes in the same plane. Note that the clustering algorithm is essential for quantization because it segments each pothole region according to the depth data related to the plane to ensure that the pixels generated by the YOLOv5 that lie outside the bounding box can also be used for pothole quantization.

2.5. Volume Quantification. The normal vectors of the output plane of the fitted plane model and the constant coefficients are represented by $a_0, a_1, a_2, d$:

$$a_0 \cdot x + a_1 \cdot y + a_2 \cdot z + d = 0. \quad (5)$$

The $Z_{pn} = f(x, y)$ function can be created by extracting these values to calculate the actual distance of the data point to the fitted surface, as illustrated in Figure 5.

$$Z_{pn}(x, y) = -\frac{(a_0 \cdot x + a_1 \cdot y + d)}{a_2}, \quad (6)$$

where $Z_{pn}$ is the coordinate of the $n$th pixel in the depth direction.

The area of each pixel differs according to the corresponding depth plane. When the distance is small, the area of a pixel is about 1 mm$^2$, while as the distance increases, the coverage area can reach 35 mm$^2$. The formula to determine the coverage area is as follows:

$$A_{pn} = \alpha \cdot \beta \cdot Z_{pn}^2, \quad (7)$$

where $A_{pn}$ is the coverage area of the $n$th pixel; $\alpha$ and $\beta$ are the Kinect constants in the horizontal and vertical directions, respectively.

To quantify the volume of the pixel, it is essential to acquire the distance value from the pixel point to the surface plane obtained by equation (2). In addition, the depth value of the pixel in the surface plane is calculated by equation (3). Thus, the relative depth value of the pixel point from the surface plane is obtained by subtracting the coordinate value of each pixel from the corresponding fitted plane in $z$-axis direction, and expressed as follows:

$$D_{pn} = Z_{pn} - Z_{fpn}, \quad (8)$$

where $D_{pn}$ is the difference in depth of the $n$th pixel, $Z_{pn}$ is the coordinate for depth of the $n$th pixel, and $Z_{fpn}$ is the coordinate for depth of the $n$th pixel in the fitted surface plane.
After calculating the volume of the pixels in the pothole region, it can quantify pavement potholes of any volume shape.

For each segmented pavement pothole, the area covered and the relative depth are multiplied to calculate the volume of the pixel, and then these pixel volumes are summed to obtain the volume of each pavement pothole. Finally, the sum of the volumes of each pavement pothole is calculated as the final total volume.

\[ V_{di} = \sum_{p} A_{pn} \cdot D_{pn} \]
\[ V_t = \sum_{i} V_{di} \]

where \( V_{di} \) is the calculated volume of the \( i \)th pavement pothole and \( V_t \) is the total calculated volume.

3. Model Training and Validation

In this section, the algorithms in Section 2 are implemented and the results are analyzed. All the processes are written in Python with deep learning framework of TensorFlow and executed on a desktop equipped with an Nvidia GTX2080Ti graphics processing unit of 11 GB.

3.1. Two-Dimensional Image Disease Classification

To create image blocks of different sizes for data extension to enhance the features learned by the model, a nonoverlapping sliding window is employed to decompose the images into small square blocks in different scales in the original training dataset. Then according to the method of Ronneberger et al. [29], all the blocks are rescaled to 300×300 pixels, ensuring seamless defect map generation in the subsequent testing phase. The values of each pixel were normalized in the interval of [−1.0, 1.0] and then the blocks were prepared for training. The mean square error is selected as the loss function, and Adam [30] is selected as the optimizer for minimizing the loss value. Finally, the initial learning rate was set to 0.01 with exponential learning rate attenuation of 0.9. The momentum parameter was set to 0.9, and 300 epochs were performed for the training with batch size of 32.

A model validation is performed after training. The normalized original blocks are fed into the classifier one by one to calculate the classification cross-entropy and accuracy. The model having the minimum loss value on validation set is selected as the final model. Each training and validation process takes about 160 s for the equipment in this study. Based on the above setup, the training process takes about 13 hours. Since the labels are the images themselves, the training is an unsupervised training process. Therefore, for this laborious and visually intensive labeling task, a lot of time and manpower can be saved.

In the test phase, a sliding window is employed to crop the test images into different sizes and then the blocks are rescaled to 300×300 pixels. The cropped blocks are scaled and normalized to generate a defect map, where a pixel represents the reestablishment error. Stitching is then performed to produce a defect map of the whole image. To obtain the probability that an image belongs to normal, cracks, or potholes, the normalized blocks are sent sequentially to the five classifiers obtained from the 5-fold cross-validation. Then, the class with the highest probability value is the class to which the blocks are most likely to belong. The classification results are shown in Figure 6. Note that the entire test process takes less than 1 minute to output a defect map of the high-resolution image.

3.1.1. Classifier Performance Evaluation

The variation of loss values and average accuracy for one cross-validation is shown in Figure 7. Although the loss values continue to decrease in the training phase, the weight parameter at the minimum validation loss is chosen to prevent overfitting. In addition, the validation accuracy of each class for cross-validation is calculated. Among them, the precision of
Figure 6: Detection results of cracking image.

Figure 7: The curve of loss and accuracy for cross-validation: (a) the loss curve; (b) the Accuracy curve.
classification of cracks, potholes, and fracturing have good performances in the range of 93.8%–95.9%, 92.4%–94.3%, and 94.1%–96.3%, respectively. However, the precision of the normal class and rut is relatively lower, at about 91.2%–92.5% and 90.4%–92.1%, respectively. The reason for rut is because this type of road disease has less data due to the low frequency for occurrence, and the reason for normal class is because of a more complex subset of images including normal images obtained at different photographic distances, illumination and shooting angles, and images of objects in different pavements.

3.1.2. Application Analysis. The ultimate purpose of pavement disease identification is to summarize pavement disease information and provide sufficient data support for the potential maintenance of pavement. Therefore, to analyze the effectiveness of the intelligent pavement disease detection system based on the deep learning method proposed in this study, the proposed method is conducted on several urban road trunk roads in Changsha (the section from Lujiazui Interchange to Changtan Expressway on Yunqi Road). To compare the detection results, manual visual disease detection is also performed on the road section. Sample results of the identified pavement diseases are shown in Figure 8.

Tables 2 and 3 show the detection results of pavement cracks and pothole by both manual detection and the proposed method. To avoid redundancy, the detection results of other diseases are shown in Tables 4 and 5. Note that the cracks and pothole are identified simultaneously in manual detection, so the consuming time is calculated according to the ratio of cracks and pothole, respectively; the consuming time of the proposed method includes the processing, model training, and testing of the collected data. Comparing the results of manual and intelligent detection, it can be observed that:

1. The results for manual detection and the proposed method are basically the same, and the accuracy of the disease is less than 10%, meeting the demand of China’s engineering inspection. Compared to manual testing, the proposed method takes less time and does not affect normal road traffic.
2. Compared with the manual detection results, the results of the proposed method are relatively inaccurate, but the accuracy rate is still higher than 90%. The reasons for the errors mainly include: (a) the dataset obtained by taking photos is not comprehensive enough; (b) the extraction of some disease features is incomplete.

3.2. In-Depth Image Disease Classification. The YOLOv5 model is trained, validated, and tested for pavement pothole recognition in depth image training. The detected pavement pothole data are used and quantified by depth cloud information using the aforementioned algorithms. The dataset contains 749 RGB images by Kinect V2, and the shooting distance was adjusted from 0.5 m to 2.5 m under different lighting conditions for data extension. Then, the obtained images were decomposed into 1107 images in the size of 853 × 1440 pixels, and labels were added to them to form a database with labels, as shown in Figure 9.

The datasets for training, validation, and test are separated according to the ratio of 8 : 1 : 1. Note that the same potholes image does not appear in different datasets at the same time. In addition, the training and validation sets were expanded by performing both horizontal and vertical flipping and exposure adjustment with values of +20 and -20 on the training and validation sets. After the data expansion, the dataset contains a total of 3455 images and 5522 bounding boxes marked as potholes.

Deep learning models require accurate training on large datasets and the process is very time-consuming. Due to the limitation of small datasets, transfer learning can be used and the YOLOv5 is pretrained. Then the original learning rate was set to 0.005 with exponential learning rate attenuation of 0.9. The momentum parameter was set to 0.9, and 800 epochs were performed for the final training with batch size of 32. Note that the anchor sizes of YOLOv5 model are set to 10; Hu, n.d.; [16, 19, 24, 31] for feature maps in the size of 75 × 75.

3.2.1. YOLOv5 Simulation Analysis. Due to the resize setting of the images in training set, the training process under GPU computing costs about 15 hours. Decreasing the size of images can reduce the training time.

YOLOv5 takes 0.08 s to process an image of 853 × 1440 pixels under GPU computation and 2.80 s under GPU computation. The highest AP value is 90.79% in all training sessions. The final model was tested by new images in test dataset, and the results are as shown in Figure 10.

3.2.2. Quantification of Road Potholes. In each case, the pixel point volume is calculated with three sets of values, which are averaged. Regardless of the target size, depth, and sensor distance, the algorithm has the ability to detect the surface and to segment and quantify the road potholes. The generated bounding boxes of YOLOv5 signify the existence of potholes on the road surface. Once the potholes are segmented and the coordinates of the surface are known, the pixels corresponding to the potholes can be extracted.

The results of the volume calculation are shown in Figure 11. The relative error of the total volume ranges from 1.49% to 13.83%, while the mean precision error (MPE) for each individual pothole considering all the distances is 14.9%.

The error varies when the distance between the sensor and the detection target is changed. And the accuracy is the best at distances of 100 cm and 200 cm.

In addition, the maximum depth values for each group of potholes were also calculated and are shown in Table 6. The error rate for each group of tests is less than 10%. For the lowest distance of 100 cm, the relative error of the test was 3.96%. In addition, the MPE value of the method in this study is lower than 8% within 200 cm compared with the MPE of 8% by Yuan et al [31].
Figure 8: The example results of diseases on Yunqi Road: (a) example results of cracking; (b) example results of pothole.

| Table 2: Statistical results of cracking. | Table 3: Statistical results of pothole. |
|-----------------------------------------|----------------------------------------|
| Manual testing | The proposed method | Manual testing | The proposed method |
| Accuracy (%) | 98 | 92.4 | 98 | 90.7 |
| Time consuming (h) | 49 | 16 | 32 | 14 |
3.2.3. Application Analysis. The road pothole quantification method proposed in this study has practical application meanings. To verify the effectiveness of the road potholes’ quantification method based on deep learning proposed in this study, the model at a distance of 100 cm is used to carry out deep learning automated detection on several township roads in Changsha County, Changsha City. In order to compare the detection results, manual disease detection was also performed on the road section and the potholes were quantified. The detection results of the two methods are shown in Table 7, and the comparison results can be obtained: the results of manual detection and deep learning detection are basically the same, and the error of pothole volume quantification is less than 10%. Compared with manual detection, the quantification method of potholes based on deep learning proposed in this study is shorter in time and more efficient in calculation within a certain accuracy range, which greatly saves human labor.

### Table 4: Statistical results of rut.

|                     | Manual testing | The proposed method |
|---------------------|----------------|--------------------|
| Accuracy (%)        | 93             | 91.7               |
| Time consuming (h)  | 37             | 14                 |

### Table 5: Statistical results of fracturing.

|                     | Manual testing | The proposed method |
|---------------------|----------------|--------------------|
| Accuracy (%)        | 98             | 93.2               |
| Time consuming (h)  | 46             | 17                 |

![The labeled dataset.](image)

![The recognition results of potholes.](image)
4. Conclusion

This study proposes a deep learning-based intelligent detection system for pavement diseases based on a depth camera. First, 200 pavement disease images are taken and transformed into 300 x 300 pixel blocks by cropping and are used as the training set. Then the RGB images are trained for defect detection, defect extraction, and classification. In the defect detection stage, a convolutional encoder is constructed to extract the disease map from a large number of pavement images. Then in the defect extraction stage, the pavement disease features are extracted using the threshold segmentation method. Moreover, in the pavement disease classification stage, the ResNet structure is used to train the model to determine the classification to which class the pavement disease belongs.

Additionally, the RGB images are transferred into depth point cloud data and YOLOv5 is used to identify the diseases and quantify their volume. First, the RANSAC algorithm is used to identify and segment the road surface based on the position and depth data of the identified road potholes. Using this algorithm, it is possible to identify and quantify multiple road pothole volumes by relying only on the output of the RGB-D sensor, free from the distance between the sensor and the detected target. In addition, a labeled database is created, trained, validated, and tested using YOLOv5.

The results show that the ResNet model takes about 3 minutes for one model training and the classification accuracy is above 90%. With the deep learning network, the sensor achieves planar fit at investigated working distances with an AP value of 90.79%. For each trench measurement volume, the average accuracy error value is less than 10%. Furthermore, both example studies on pavement show that the proposed detection method can save a lot of labor and improve the detection efficiency to a certain accuracy.

In the future, the sensor can be mounted to an unmanned vehicle for data collection in remote or hazardous areas, so the data obtained are more reliable. Since the depth sensor used in this study is relatively inexpensive, replacing it with a more advanced depth sensor may further improve the quantitative accuracy.

Data Availability

The data used in the paper are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Ministry of Transport of China, grant no. 2020-MS5-145, and by the State Archives Bureau of China, grant no. 2021-X-45.

References

[1] Z. F. Qiao and H. Zhu, “Research on disease analysis and maintenance of cement concrete pavement in Henan province,” Highway Engineer, no. 3, pp. 12–16, 1996.
[2] H. L. Zhang and S. F. Chen, “Selection of structures for rehabilitation of asphalt pavements,” Highway Engineer, vol. 33, no. 6, pp. 60–63+68, 2008.
[3] J. H. Zeng and J. Xu, “Damage evaluation index system of Expressway asphalt concrete pavements,” Highway Engineer, vol. 34, no. 3, pp. 120–122, 2009.
[4] X. G. Zheng, Q. Yang, and W. M. Lv, “Distress characteristics and mechanics of moisture damage in asphalt pavement,” Highway Engineer, vol. 31, no. 2, pp. 96–98, 2006.
[5] J. X. Chen and M. R. Zhou, “Application of polymer grouting technology in treatment of cavitation diseases of cement
concrete pavement of a class n highway,” *Highway Engineer*, vol. 44, no. 4, pp. 174–177, 2019.

[6] B. Ding, Y. Lei, J. Zhang, T. Chen, and X. Zuo, “Study on evaluation method of pavement condition in permafrost regions,” *Highway Engineer*, vol. 42, no. 6, pp. 107–109, 2017.

[7] G. X. Jiang, H. W. Peng, C. L. Sun, and H. Yuan, “Investigation and cause analysis on rutting of highway asphalt pavement in Hunan province,” *Highway Engineer*, no. 1, pp. 8–10+14, 2011.

[8] A. Boyko and T. Funkhouser, “Extracting roads from dense point clouds in large scale urban environment,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 6, pp. S2–S12, 2011.

[9] H. Guan, J. Li, Y. Yu, M. Chapman, and C. Wang, “Automated road information extraction from mobile laser scanning data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 194–205, 2015.

[10] B. Wang, X. Wang, C. Fei, Y. He, W. Li, and L. Liu, “Pavement crack recognition based on aerial image,” vol. 37, Article ID 0810004, 2017.

[11] Y. Hu and Z. Chun-xia, “A local binary pattern based methods for pavement crack detection,” *Journal of Pattern Recognition Research*, vol. 1, no. 20103, pp. 140–147, 2010.

[12] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, “Automatic road crack detection using random structured forests,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016.

[13] N. Li, “Research on pavement crack identification based on deep learning framework Caffe,” *Metallurgical Collections*, no. 3, p. 20+28, 2017.

[14] N. Zhang, *Study on Detection Algorithm for Road Surface Disease Based on Yolo V5*, East China Jiaotong University, Nanchang, China, 2019.

[15] H. Guo, W. Xu, and Y. F. Liu, “Steel plate surface defect recognition based on support vector machine,” *East China University of Science and Technology*, vol. 44, no. 4, pp. 635–639, 2018.

[16] Y.-J. Cha, W. Choi, and O. Büyükoztürk, “Deep learning-based crack damage detection using convolutional neural networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[17] N. Snavely, “Scene reconstruction and visualization from internet photo collections: a survey,” *IPSJ Transactions on Computer Vision and Applications*, vol. 3, pp. 44–66, 2011.

[18] M. M. Torok, M. Golparvar-Fard, and K. B. Kochersberger, “Image-based automated 3D crack detection for post-disaster building assessment,” *Journal of Computing in Civil Engineering*, vol. 28, Article ID A4014004, 2014.

[19] M. R. Jahanshahi, F. Jazizadeh, S. F. Masri, and B. Becerik-Gerber, “Unsupervised approach for autonomous pavement defect detection and quantification using an inexpensive depth sensor,” *Journal of Computing in Civil Engineering*, vol. 27, no. 6, pp. 743–754, 2013.

[20] J. K. Chow, “Artificial intelligence-empowered pipeline for image-based inspection of concrete structures,” *Automation in Construction*, vol. 120, 2020.

[21] J. G. Yang, L. Xu, Z. J. Lu, Q. Xiang, B. Liu, and S. Y. Liang, “Textile processing anomaly detection based on time domain features and GA-SVM,” *Applied Mechanics and Materials*, vol. 740, pp. 706–713, 2015.

[22] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. Goh, “AnomalyNet: an anomaly detection network for video surveillance,” *IEEE Transactions on Information Forensics and Security*, vol. 14, 2019.

[23] M. Celik, F. Dadaser-Celik, and A. Dokuz, “Anomaly detection in temperature data using DBSCAN algorithm,” in *Proceedings of the Presented at the INISTA 2011-2011 International Symposium on INnovations in Intelligent Systems and Applications*, Istanbul, Turkey, June 2011.

[24] D. Hendrycks, M. Mazeika, and T. Dietterich, “Deep anomaly detection with outlier exposure,” 2019, https://arxiv.org/abs/1812.04606.

[25] D. Junfeng and Z. Xiaolong, “Deep learning algorithm optimization based on combination of auto-encoders,” *Journal of Computer Applications*, vol. 36, p. 697, 2016.

[26] C. Yuan and H. Cai, “Automatic detection of pavement surface defects using consumer depth camera,” in *Proceedings of the Construction Research Congress 2014*, pp. 974–983, Atlanta, Georgia, May 2014.

[27] A. Kuznetsova, T. Maleva, and V. Soloviev, “Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images,” *Advances in Neural Networks*, pp. 233–243, 2020.

[28] A. Boykovskiy, C.-Y. Wang, and H. Y. M. Liao, “YOLOv4: optimal speed and accuracy of object detection,” 2020, https://arxiv.org/abs/2004.10934.

[29] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: convolutional networks for biomedical image segmentation,” 2015, https://arxiv.org/abs/1505.04597.

[30] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2017, https://arxiv.org/abs/1412.6980.

[31] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.