

## Intelligent Detection Model Based on a Fully Convolutional Neural Network for Pavement Cracks

Duo Ma<sup>1, 2, 3</sup>, Hongyuan Fang<sup>1, 2, 3, \*</sup>, Bingham Xue<sup>1, 2, 3</sup>, Fuming Wang<sup>1, 2, 3</sup>,  
Mohammed A. Msekh<sup>4</sup> and Chiu Ling Chan<sup>5</sup>

**Abstract:** The crack is a common pavement failure problem. A lack of periodic maintenance will result in extending the cracks and damage the pavement, which will affect the normal use of the road. Therefore, it is significant to establish an efficient intelligent identification model for pavement cracks. The neural network is a method of simulating animal nervous systems using gradient descent to predict results by learning a weight matrix. It has been widely used in geotechnical engineering, computer vision, medicine, and other fields. However, there are three major problems in the application of neural networks to crack identification. There are too few layers, extracted crack features are not complete, and the method lacks the efficiency to calculate the whole picture. In this study, a fully convolutional neural network based on ResNet-101 is used to establish an intelligent identification model of pavement crack regions. This method, using a convolutional layer instead of a fully connected layer, realizes full convolution and accelerates calculation. The region proposals come from the feature map at the end of the base network, which avoids multiple computations of the same picture. Online hard example mining and data-augmentation techniques are adopted to improve the model's recognition accuracy. We trained and tested Concrete Crack Images for Classification (CCIC), which is a public dataset collected using smartphones, and the Crack Image Database (CIDB), which was automatically collected using vehicle-mounted charge-coupled device cameras, with identification accuracy reaching 91.4% and 86.4%, respectively. The proposed model has a higher recognition accuracy and recall rate than Faster RCNN and different depth models, and can extract more complete and accurate crack features in CIDB. We also analyzed translation processing, fuzzy, scaling, and distorted images. The proposed model shows a strong robustness and stability, and can automatically identify image cracks of different forms. It has broad application prospects

---

<sup>1</sup> School of Water Conservancy Engineering, Zhengzhou University, Zhengzhou, 450001, China.

<sup>2</sup> National Local Joint Engineering Laboratory of Major Infrastructure Testing and Rehabilitation Technology, Zhengzhou, 450001, China.

<sup>3</sup> Collaborative Innovation Center of Water Conservancy and Transportation Infrastructure Safety, Zhengzhou, 450001, China.

<sup>4</sup> Civil Engineering Department, College of Engineering, University of Babylon, Babylon, Iraq.

<sup>5</sup> Institute of Structural Mechanics, Bauhaus Universität Weimar, Weimar, Germany.

\* Corresponding Author: Hongyuan Fang. Email: 18337192244@163.com.

Received: 13 November 2019; Accepted: 09 January 2020.

in practical engineering problems.

**Keywords:** Fully convolutional neural network, pavement crack, intelligent detection, crack image database.

## 1 Introduction

Pavement plays an important role in transportation. However, whether it is cement or asphalt, cracks, ruts, or other diseases will appear after a period of use, among which cracks are the most common. Pavement cracks not only affect ride comfort but also facilitate the deterioration of pavement function. With the aggravation of vehicle traffic and freeze-thaw cycles, cracks easily expand, causing structural damage to the pavement and reducing its service life. Therefore, once cracked, pavement should be promptly repaired. Preventive maintenance can effectively slow cracks propagation and reduce the maintenance cost of pavement [DonaldN (1996)]. Therefore, a large amount of pavement crack information should be collected to effectively detect its condition and provide a basis for preventive repair [Zhang, Zhao and Rabczuk (2018)].

Crack detection currently relies on subjective judgment. Manual inspection is limited by practical factors and fatigue, while a computer can work steadily for a long time without rest [Zhang, Wang, Lahmer et al. (2016)]. And a computer can detect sub-domains of different materials at high noise levels [Nanthakumar, Lahmer, Zhuang et al. (2016)]. Moreover, a computer's detection speed is fast. Because of its obvious advantages in accuracy, efficiency, and cost, computer vision can come into play [Szeliski (2010); Zhang, Nanthakumar, Lahmer et al. (2017)].

Traditional pavement crack-detection methods can be roughly divided into three categories: 1. The gray value of cracks is lower than that of the background, such as threshold segmentation [Christian, Kristina, Varun et al. (2015); Oliveira and Correia (2009)], histogram segmentation [Kirschke and Velinsky (1992)], color difference [Premachandra, Premachandra, Parape et al. (2015)], minimum path [Avila, Begot, Duculty et al. (2014)], anisotropy [Shabalov, Feldman and Bashirov (2012)], filter [Salman, Mathavan, Kamal et al. (2013); Haris, Muhammad and Khurram (2016); Eduardo, Jaime, Roberto et al. (2014); Zhang, Li, Wang et al. (2013)], and CrackTree [Zou, Cao, Li et al. (2012)]. 2. The gray value at the crack edge changes greatly so complex artificial edge detectors [Albert and Nii (2008); Gunawan, Nuriyanto, Sriadhi et al. (2018); Cubero-Fernandez (2017)] or wavelet analysis [Slamet, Fiddin, Chayadi et al. (2016); Tomiyama and Kawamura (2016); Jian, Huang and Chiang (2006); Wang, Li and Gong (2007)] are adopted for feature extraction. 3. A feature extractor is based on early machine learning methods like random forest [Shi, Cui, Qi et al. (2016)], Adaboost [Wang, Yang, Cheng et al. (2018)], and support vector machine (SVM) [Ai, Jiang, Kei et al. (2018)].

Due to inconsistent directions, irregular textures, and uneven shapes, cracks are difficult to identify. In addition, pavement images contain much noise. Changes in light and shade, dust, and driving speed all have a great impact on detection results. Hence, these traditional methods are incapable of analyzing all the features of the cracks. These methods compare favorably with convolutional neural network (CNN) when using

small-scale data, but have low recognition accuracy and poor generalization ability when dealing with large-scale data.

The continual improvement of computer performance and the popularization of the Internet have produced a large amount of data and provided strong support for deep learning. CNNs are artificial neural networks based on deep learning [Lecun, Bengio and Hinton (2015)]. This network model repeatedly learns image data to obtain comprehensive crack characteristics. Furthermore, it is robust for scaling, translation, rotation, mirroring, contrast enhancement, and other deformation operations of cracks. CNNs have enabled great achievements in the fields of automatic driving [Zhao (2018)] and biometric traits [Rajalakshmi, Rengaraj, Mukund et al. (2018)], including crack identification. Anitescu et al. [Anitescu, Atroshchenko, Alajlan et al. (2019)] proposed a method to solve the second order boundary value problem using artificial neural network, which improved the robustness, saved a lot of computation and had the characteristic of high accuracy. To solve the problem of sheet bending, Guo et al. [Guo, Zhuang and Rabczuk (2019)] proposed a depth configuration method based on deep neural network. This method proves that increasing the number of layers can improve the accuracy and convergence. Chen et al. [Chen and Jahanshahi (2017)] recently used an underwater camera to collect crack videos of nuclear power plant components, and made an eight-layer model to classify tubelets of each frame. A similar method was adopted by Wang et al. [Wang and Zhang (2017)] to segment road images into image cells with a resolution of  $64 \times 64$  pixels, which were detected by a CNN of three convolutional layers and two full-connection layers. Zhang et al. [Zhang, Wang, Li et al. (2017)] preprocessed images using 360 artificial feature extractors and trained a four-layer neural network for classification. Park et al. [Park and Bang (2019)] used dash-cam data to train a fully convolutional network (FCN) to segment road pixels, and trained sub-patches with a resolution of  $40 \times 40$  pixels from the central area. Feng et al. [Feng, Xu and Guo (2019)] used multi-scale feature maps from different positions of a residual network to train a classification network. Zhang et al. [Zhang, Yang, Zhang et al. (2016)] trained a ConvNets classifier with pavement images collected by a mobile phone. Cha et al. [Cha, Choi and Oral (2017)] proposed a seven-layer classification network to detect concrete crack sub-regions. These CNNs have achieved excellent results, but the methods still have the drawback that the relationship between patches and surrounding pixels cannot be considered, so the overall features of cracks cannot be learned. They are just image-level classifications rather than regional-level detection. In addition, the extraction of sub-regions cannot be integrated into CNN, so a picture must be repeatedly calculated. Moreover, pictures collected by mobile phone mainly rely on manual work, which is slow and subjective. Pictures collected by dash-cam have too many redundant targets, and distant and lateral cracks are blurred. Finally, the number of the layers is few, which limits extraction of crack features.

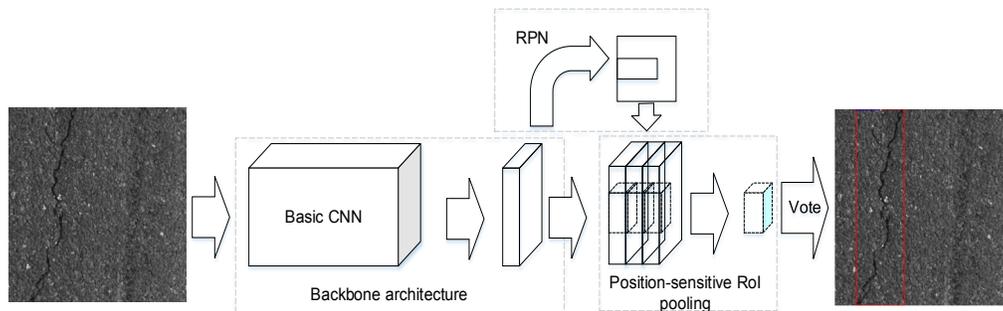
With the continuous development of deep learning, a large number of region-level object-detection methods have been proposed. Those based on CNN can be roughly segmented into two categories: two-stage algorithms based on sliding windows, such as R-CNN [Girshick, Donahue, Darrelland et al. (2014)], SPP Net [He, Zhang, Ren et al. (2014)], Fast R-CNN [Girshick (2015)], and Faster R-CNN [Ren, He, Girshick et al. (2015)], and regression-based one-stage algorithms, such as YOLO [Redmon, Divvala,

Girshick et al. (2015)] and SSD [Liu, Anguelov, Erhan et al. (2015)].

Aiming at the large data volume and complex background environment of pavement-crack images, the proposed method detects the pavement cracks based on a fully convolutional neural network by combining the 100-layer basic residual network and R-FCN [Dai, Li, He et al. (2016)] framework. This improves Faster RCNN by canceling the fully connected sub-network, replacing RoI (Region of Interest) pooling with position-sensitive RoI pooling, which solves the problem of translatability invariance. The work incorporates region proposal network (RPN) to extract the crack region of the feature map, which integrates the sub-region extraction task into the network and greatly improves the computing efficiency. Online hard example mining (OHEM) and data augmentation are used to improve accuracy. The proposed model is trained on the public dataset CCIC and the Crack Image Database (CIDB) collected by vehicle-mounted charge-coupled device (CCD) cameras. The fully convolutional neural network shows higher accuracy and faster processing speed than state-of-the-art methods, and can extract more abstract features. It achieves region-level detection and has a strong generalization ability.

## 2 Methodology

The fully convolutional neural network model adopted in this study includes the basic residual network, RPN, position-sensitive RoI pooling, and an output layer. To reduce the error between the predicted and ground-truth values, gradient descent [Lecun, Bottou, Bengio et al. (1998)] and backpropagation [Lecun, Bottou, Orr et al. (1998)] algorithms are used to update the network's parameters. Fig. 1 shows the structure of this model. The images are input to shared basic CNN to obtain the feature map, which is calculated twice, first to get the region proposal through RPN, and then to get the position-sensitive score map by  $1 \times 1$  convolution kernel. The region proposal and position-sensitive score map is matched, and the final identification result is obtained through position-sensitive RoI pooling and voting.

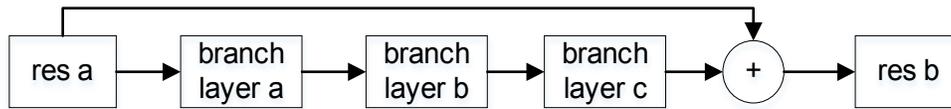


**Figure 1:** Schematic of proposed model

### 2.1 Basic residual network

As the depth increases, it becomes more difficult to train and optimize the network. Simply stacking convolutional layers and pooling layers cannot upgrade the model, but will cause gradient disappearance, which is a degradation problem [He, Zhang, Ren et al.

(2015)]. We propose ResNet to solve this problem. Fig. 2 shows a residual block of the basic network. Res a and res b represent two residual nodes. Between res a and res b are three branches, which are branch layers a, b, and c. Each branch includes a convolution layer, batch normalization layer, and activation function layer. Convolution has two advantages: parameter sharing and sparse connection. Convolution calculation greatly reduces storage and computation requirements. Batch normalization not only speeds up the convergence of the model but also alleviates the problem of gradient disappearance in the deep network to some extent, making it easier and more stable to train the deep network model [Ioffe and Szegedy (2015)]. The activation function is the rectified linear unit (ReLU), which compared with the early activation function sigmoid and tanh, converges faster and avoids gradient explosion and gradient vanishing [Maas, Hannun and Ng (2013)]. In addition, ResNet directly connects two residual nodes through a shortcut connection. Through this connection mode, ResNet solves the degradation problem and improves network performance while increasing network depth.



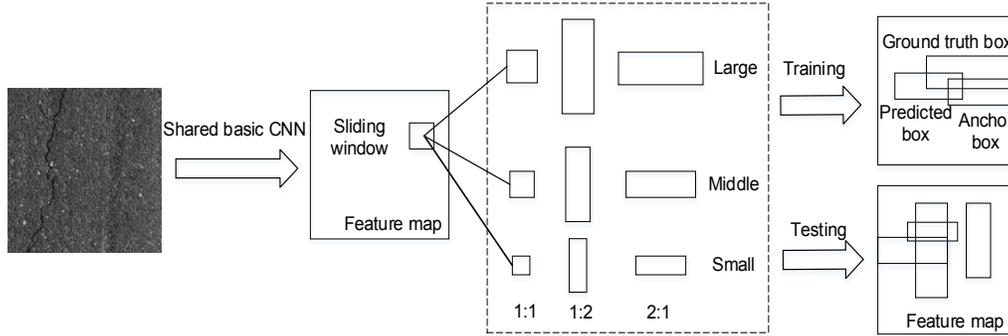
**Figure 2:** Residual block

## 2.2 Region proposal network

The main idea of two-stage CNN based on region proposal is to generate a sparse region of interest (RoI) by a candidate region algorithm, and then to classify the candidate regions. Traditional methods to extract candidate regions are selective search (SS) and edge boxes (EB). Efficient CNN fully exploits the advantages of a graphics processing unit (GPU), but these early candidate-region extraction algorithms can only be carried out on a central processing unit (CPU). These algorithms cannot be integrated into CNN, and the convolution features cannot be shared. This reduces the computational efficiency and accuracy of candidate-region extraction.

As shown in Fig. 3, shared basic CNN outputs a feature map and slides a small  $3 \times 3$  window on it. Anchor points are located in the center of the sliding window. Three scales of 8, 16, and 32, and three aspect ratios of 0.5, 1, and 2 were considered to obtain nine anchor boxes. A convolutional feature map of size  $W \times H$  has a total of  $9 \times W \times H$  anchor boxes. When training RPN, Intersection-over-union (IoU) was used to assign two kinds of anchor boxes to positive labels: the anchor box with the highest IoU value with a truth box, or the anchor box with an IoU value greater than 0.7 with any truth box; assign a type of anchor boxes to a negative label: the anchor box with an IoU value less than 0.3 with all truth boxes. Most images have more negative anchor box frames than positive anchor boxes. Training all anchor boxes at the same time will bias the results toward negative anchor boxes. In this study, 256 anchor boxes are extracted from an image to calculate the loss function of a small batch, where the ratio of positive and negative anchor boxes is 1:1. If the number of positive anchor boxes in an image is less than 128, we fill it with negative anchor boxes. The purpose of RPN learning is to cause the offset

of the predicted box relative to the anchor box close to the offset of the ground-truth box relative to the anchor box, so as to accurately predict the position of the region proposals according to the position of the anchor box. In the test, the scores of predicted boxes were ranked from high to low. The 300 predicted boxes with the highest scores above a fixed threshold were selected as the region proposals.

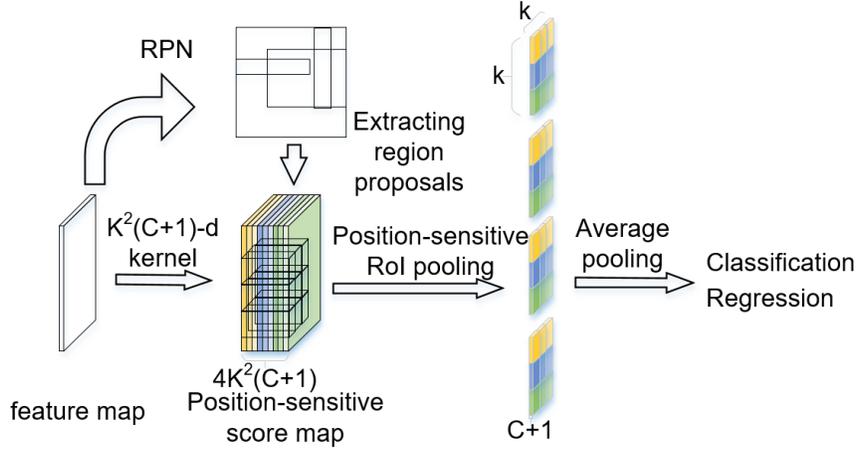


**Figure 3:** Region proposal network

### 2.3 Position-sensitive RoI pooling

The appearance of fully convolutional neural networks, such as GoogLeNet [Szegedy, Vanhoucke, Ioffe et al. (2016)] and ResNet [He, Zhang, Ren et al. (2015)], proves that the use of convolution instead of the full connection not only has better effects but also can adapt to inputs of different sizes. However, if the full connection layer is discarded directly and the feature map is connected to the final classification layer and regression layer, then the effect is poor. This is mainly because these fully convolutional neural networks are image-level classification networks, which prefer position invariance. Position-sensitive RoI pooling can effectively solve this problem.

As shown in Fig. 4, in the last feature map of the shared CNN, RPN and position-sensitive RoI pooling are respectively employed to extract region proposals and solve the problem of translational invariance. First, the 1024-d convolutional kernels of size  $l \times l$  are used to generate the position-sensitive score map of depth  $4 \times k^2 \times (C+1)$  by the convolution calculation of the 2048-d feature map. Similar to the idea of dropout, this reduces dimensions and avoids overfitting to some extent. Then the region proposals are mapped to the position-sensitive score map. The depth of all region proposals on the position-sensitive score map is  $4 \times k^2 \times (C+1)$ , where  $C$  is the number of object types ( $C=1$  in this example),  $+1$  corresponds to the background, 4 corresponds to the four position coordinates required by regression, and  $k^2$  represents the division of the region proposals into  $k \times k$  bins ( $k=3$  in this example). Next, position-sensitive RoI pooling is carried out to divide region proposals with the size  $w \times h$  into  $k \times k$  bins, each with size  $(w/k) \times (h/k)$ . Finally, average pooling is conducted for each region proposal to obtain  $(C+1)$ -dimensional vectors used for classification and regression. The fully convolutional neural network is completely shared and greatly improves the efficiency.



**Figure 4:** Schematic diagram of position-sensitive RoI pooling

## 2.4 Output layer

Traditional neural network classification uses SVM. When the data dimension is low and the data volume is small, it does not show shortcomings. However, in object detection, due to the huge amount of data, the SVM is gradually replaced by the softmax function. A multitask function is proposed by combining softmax with a bounding box regression function to reduce the computational complexity and improve accuracy. The cost function in this work is the sum of cross-entropy loss and boundary box regression loss

$$L(p, t) = L_{cls}(p, c) + \lambda [c > 0] L_{bbreg}(y, v) \quad (1)$$

where, when  $c > 0$ ,  $\lambda$  is equal to 1, and when  $c = 0$ ,  $\lambda$  is equal to 0, and

$$L_{cls}(p, c) = -\log p_c \quad (2)$$

Eq. (2) is the cross-entropy loss, that avoids the problem of slow convergence of sigmoid units. Moreover, the greater the difference between the predicted and ground-truth values the greater the cross-entropy loss. The advantage is that the model tends to bring the predicted value closer to the ground-truth value. The softmax function is defined by

$$p_c = \frac{e^{r_c}}{\sum_{c'=0}^C e^{r_{c'}}} \quad (3)$$

$$r_c = \sum_{i,j} r_c(i, j | \Theta) \quad (4)$$

where  $r_c$  is a  $c$ -dimensional vector generated by position-sensitive RoI pooling, and  $i$  and  $j$  represent a block (bin) in row  $i$  and column  $j$ . The bounding box regression function is defined by

$$L_{bbreg}(t, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L1}(t - v) \quad (5)$$

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (6)$$

where  $t=(t_x, t_y, t_w, t_h)$  and  $v=(v_x, v_y, v_w, v_h)$  represent the position coordinates of the prediction box and ground-truth box, respectively.

### 3 Experiment and result

To train CNN requires much marked image data. A variety of publicly available datasets are available, among which ImageNet [Deng, Dong, Socher et al. (2009)], COCO [Lin, Maire, Belongie et al. (2014)], PASCAL VOC [Everingham and Winn (2006)], and CIFAR-10 are the most commonly used. However, few datasets are available with images of pavement cracks. A public pavement crack dataset, Concrete Crack Images for Classification (CCIC), was trained to compare our model with ConvNet [Zhang, Yang, Zhang et al. (2016)]. The dataset has 1595 images of  $227 \times 227$  pixels. CCIC was labeled artificially to make it conform to the object-detection task. In this work, the sample set was randomly divided into training and test sets at 9:1 ratio. There were 1435 pictures in the training set and 160 in the test set.

A high-performance computer is indispensable for large-scale image-data processing. We used an Intel Core i7-7800 CPU with 64 GB RAM and an NVIDIA Titan V GPU. The model was based on the Ubuntu 16.04 operating system, using the Caffe 1.0 architecture. The computer language was Python 2.7. CUDA 8.0 and CUDNN 9.0 were used to improve computing performance.

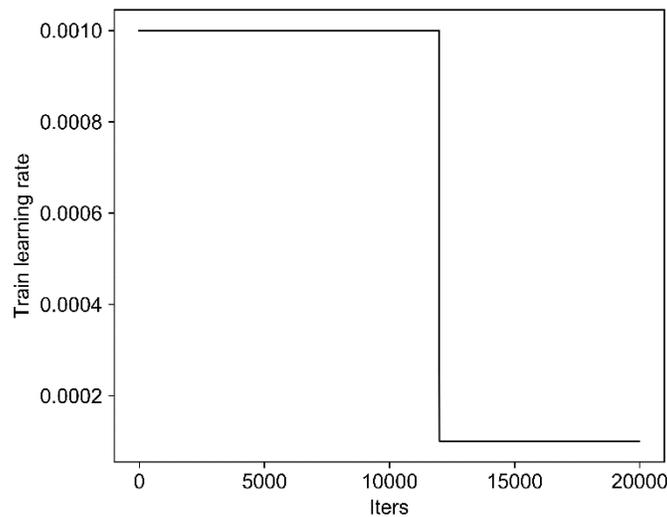
#### 3.1 Experiment on CCIC

##### 3.1.1 Hyperparameters

CNN must set many hyperparameters that affect the model's convergence. Most important are the number of iterations and the learning rate. Too few iterations lead to underfitting, and too many iterations lead to overfitting. With too high a learning rate, the loss function will increase and prevent convergence, and if it is too low, the learning speed is slow and the loss function may not change at all. Hence it is important to set a reasonable number of iterations and an appropriate learning rate. The selection of the iteration number and the learning rate mainly depends on observing the variation graph of the loss function. When the loss function converges, the iteration is stopped. And the learning rate is mainly dependent on the convergence rate of the loss function. The initial learning and the total number of iterations is set to 0.001 and 20000, respectively. The step method is used to adjust the learning rate to make the model converge as fast as possible. The learning rate drops by an order of magnitude after 12000 iterations, as shown in Fig. 5. To accelerate training, the backpropagation method was a mini-batch gradient descent algorithm with simple momentum. As Tab. 1 shown, mini-batch size, weight decay value and momentum coefficient have slight influence on the accuracy of the model. Weight decay value of 0.0005 and momentum coefficient of 0.90 are used. Due to the limitation of computer memory capacity, mini-batch size should be selected the maximum one. In this paper, mini-batch size is 2.

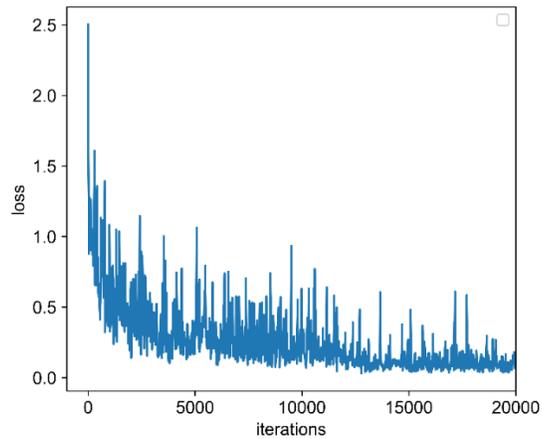
**Table 1:** Models with different hyperparameters

Learning rate	Mini-batch size	Weight decay	Momentum coefficient	Accuracy
0.0010	2	0.0005	0.90	91.40%
0.0010	2	0.0005	0.99	90.91%
0.0010	2	0.0005	0.50	90.85%
0.0010	2	0.0001	0.90	90.91%
0.0010	2	0.0010	0.90	90.80%
0.0010	1	0.0005	0.90	90.91%
0.0050	2	0.0005	0.90	90.80%
0.0005	2	0.0005	0.90	90.91%

**Figure 5:** Learning-rate curve

### 3.1.2 Training and transfer learning

Unlike a traditional crack-detection algorithm, CNN does not need to manually extract the crack features. Instead, it can automatically learn features by updating the weight of the convolutional kernel. The model is initialized according to the above hyperparameters. The transfer learning instead of randomly initializing network parameters is adopted as a starting point for the training model. Transfer learning uses a network that was pre-trained by others and applied to other fields. This can accelerate convergence and improve the accuracy. The loss values were output at the terminal every 20 iterations to draw the loss-value decline curve, as shown in Fig. 6. This could better control the number of iterations and enable us to grasp the convergence rate. After 20000 iterations, the loss value was below 0.1, and hence, the model had converged.



**Figure 6:** Loss function decline

### 3.1.3 Testing and assessment

To show the training effect of the proposed model, 160 images were tested. The testing standard was that a predicted bounding box with an IoU overlap higher than 0.5 with the ground-truth box was correct detection. Fig. 7 shows an example of a testing result. The red box in the figure is the location and marking of cracks, and the label on the top is the red box. The value of 1.000 represents the probability of cracks in the red box. It can be seen that our proposed model can not only distinguish the cracks but also identify their location with a rectangular prediction box. Our model achieved 91.4% accuracy, which exceeded that of ConvNet designed by Zhang et al. [Zhang, Yang, Zhang et al. (2016)]. Counting the load time, it took only 1.5 s to test an image on average. This is because it uses one network calculation to detect the whole image, instead of feeding each sub-region into the network, which saved the computing time.



**Figure 7:** Example of testing result

### 3.1.4 PR curve

Accuracy refers to the prediction result, indicating the ratio of correctly predicted samples (the summation of true positive and false positive) to all samples,

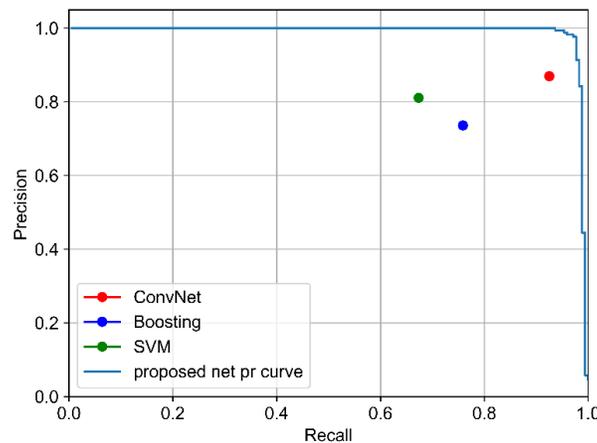
$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (7)$$

Here, TP, FP, FN, and TN are the numbers of true positive, false positive, false negative, and true negative, respectively. However, when the categories are unbalanced, accuracy is not a comprehensive evaluation of the model, and a precision-recall curve should be used as the numerical evaluation index. Recall refers to the ratio of true positive samples to actual positive (true positive and false negative) samples. Precision is the ratio of true positive samples to predicted positive (true positive and false positive) samples. The recall rate and accuracy rate are defined as

$$recall = \frac{TP}{TP + FN} \quad (8)$$

$$precision = \frac{TP}{TP + FP} \quad (9)$$

We can adjust the confidence threshold to obtain different precision and recall values, and we can derive a curve, such as that in Fig. 8. As the confidence threshold decreases, the precision decreases and the recall rate increases. For object detection, the PR curve near the upper-right corner represents accurate and complete recognition. The recall rate and precision of ConvNet, Boosting, and SVM [Zhang, Yang, Zhang et al. (2016)] are marked in the PR curve. The precision and recall rate of ConvNet were 86.96% and 92.51%, respectively. When the precision was the same, the proposed model had a recall rate of 98.27%. This means that 5.76% of cracks would be missed by ConvNet. The curve of the proposed model is closer to the upper-right corner, indicating that this model is not only accurate but more comprehensive, and it can effectively distinguish between cracks and non-cracks.



**Figure 8:** Precision-recall curve

### **3.2 Experiment on CIDB**

The proposed model has shown excellent results in the public dataset CCIC. However, those images were shot with a mobile phone. This collection method is time-consuming and hard sledding. In actual collection, inspection is conducted with a vehicle camera. In addition, the cracks in CCIC are too obvious. In practice, cracks are often small. And when a crack is obvious enough, the pavement is already damaged, which is contrary to the concept of preventive repair. Furthermore, after excessive cutting, each picture only retains local information about cracks. Hence, it is important to establish a labeled pavement-crack image dataset with high definition and a sufficient quantity of images.

#### *3.2.1 Data collection*

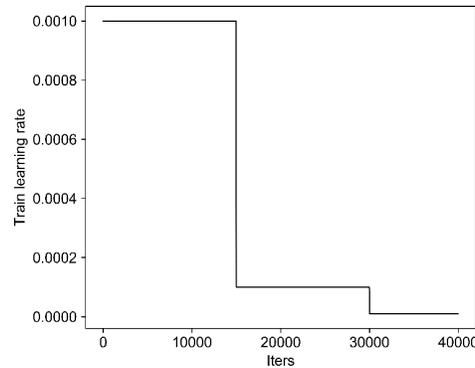
Many countries have carried out this research work, such as PAVUE in Sweden, Komatsu in Japan, the PCES system of the ETC Company in the United States, the PAVUE system of the IMS Company, the CREHOS system of the Swiss Federal Technical Research Institute, and the road-crack detection car developed by WDM in the United Kingdom. Image-acquisition devices using computer vision are generally divided into CMOS (Complementary Metal-oxide Semiconductor) and CCD, which take the subject as the image signal and transmit it to the image-processing system. Although the CCD camera is relatively expensive, its sensitivity and resolution exceed those of the CMOS camera. The CMOS camera is also noisy, which will affect the image quality. A CCD camera robot has been applied to bridge monitoring [Yu, Guo, Wang et al. (2012)], and it can also be used to collect pavement-crack images. To obtain real crack images under different lighting conditions, a vehicle-mounted CCD camera was used to take photos of pavement under natural light in sunny and cloudy weather. High-resolution road images with  $2048 \times 2048$  pixels were recorded. The collecting method used for this work has three advantages over CICC: 1. Automatic collection saves manpower and financial resources. 2. There will be light and dark changes under natural conditions, which better conforms to reality. 3. A CCD camera has high sensitivity and resolution, and can store a large amount of image data, which fits the demands of deep learning and image resolution. The pavement images used in this paper were taken from 10 expressways in Guangdong province between 2016 and 2018.

#### *3.2.2 Image processing and labeling*

The original images were divided into images of  $512 \times 512$  pixels. The positions and ranges of cracks (coordinates of the upper-left and lower right corners) were marked with rectangular boxes. The data layer of Caffe only supports input data in the LevelDB, LMDB, and HDF5 formats. Image datasets must be processed into data types that can be recognized by the network model. Images were converted to the LMDB format. Pavement-crack images were randomly divided, with 90% used as a training set and 10% for testing. The data were augmented to generate new images with a 90-degree rotation. After preprocessing, labeling, and data-augmentation, 6295 images were integrated into the Crack Image Database (CIDB).

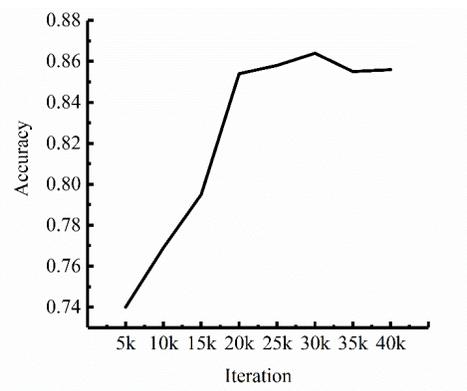
### 3.2.3 Training

The transfer learning method was also adopted in training on CIDB, and the model parameters were initialized with the pre-training model. The learning rate dropped by an order of magnitude after every 15,000 iterations, as shown in Fig. 9. Other hyperparameters were determined to be consistent with the training on CICC. The loss value and accuracy on the training set were saved every 20 iterations, and model parameters were saved after every 5000 iterations.

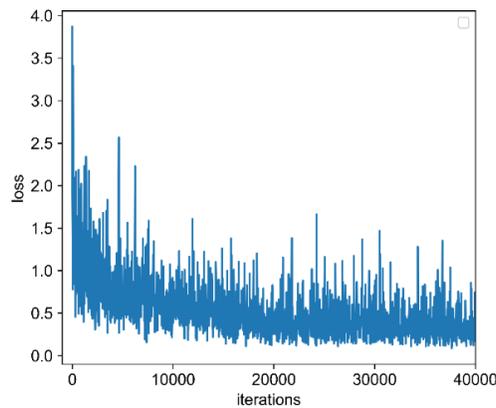


**Figure 9:** Learning-rate curve

The total number of iterations was set to 40,000. To judge whether the proposed model was optimal, the model output for each 5000 iterations was used to detect on the test set of 332 images, and the accuracies of multiple iterations were plotted as a line chart, as shown in Fig. 10. The loss-function decline curve is shown in Fig. 11. The proposed model had the highest accuracy rate, 86.4%, after 30,000 iterations. The model has a strong generalization ability. When the number of iterations is small, the model does not fully learn the features of cracks, and this is the underfitting stage. When the number of iterations is too high, the model learns meaningless noise, and the accuracy rate decreases, which is the overfitting stage. From the loss-function decline curve, it can be seen that after 30,000 iterations, the loss value tends to converge and reaches the local optimal solution.



**Figure 10:** Accuracy of multiple iterations



**Figure 11:** Loss-function decline

### 3.2.4 Testing and assessment

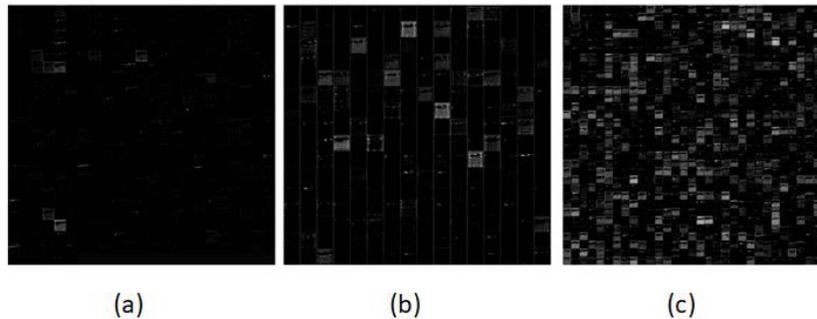
In 2012, AlexNet made a breakthrough that the top-5 error rate was reduced to 16.4% in the top competition ILSVRC (ImageNet Large Scale Visual Recognition Challenge) in the computer vision field, and since then, deep CNN has been widely used in the field of computer vision [Krizhevsky, Sutskever and Hinton (2012)]. In 2014, VGG-16 set a new record that the top-5 accuracy is 90.1% for ILSVRC [Simonyan and Zisserman (2014)]. With the increase of neural network depth, the effect of network feature extraction is improved. However, the deepening of the network also brings the important problem of gradient disappearance, which decreases the network's accuracy. Network depth dilutes the gradient, limiting the network to train effectively. The ResNet framework solves this problem well.

Tab. 2 shows the comparative analysis of different networks when OHEM is not used. The computational efficiency is divided into training efficiency and test efficiency. The training efficiency is represented by the average time of an iteration, while the test efficiency refers to the average test speed of an image, as represented by the average test time. The accuracy is the ratio of the number of correctly predicted samples to the number of samples in the test set. These models were trained through 30,000 iterations without OHEM. The accuracy of Faster RCNN based on ZF was 68.4%, at a test time of 0.027 s. The accuracy of Faster RCNN based on VGG16 was 80.7%, at a test time of 0.055 s. The accuracy of R-FCN based on ResNet-50 was 79.0%, at a test time of 0.051 s. R-FCN based on ResNet-101 achieved the highest accuracy, 84.6%, at a test time of 0.076 s, without affecting the detection efficiency. ResNet-50 has 2.5 times more layers than VGG16. But R-FCN based on ResNet-50 trains faster and their detection accuracy is much the same. This is because the proposed model uses the fully convolutional neural network instead of fully connected layers. ResNet-50 has fewer layers than ResNet-101, hence its faster computation speed, but ResNet-101 is more accurate, and that is worth the slower speed. The accuracy increases when the depth is increased from 50 to 101, but decreases slightly with a depth of 152. The accuracy of the 152-layer model was 82.8%, 1.8% lower than that of the 101-layer network. Although they have similar accuracy. This

may be due to the increase in the number of network layers leading to the overfitting problem [He, Zhang, Ren et al. (2015)]. The proposed model based on ResNet-101 performs best.

**Table 2:** Performance of different depths

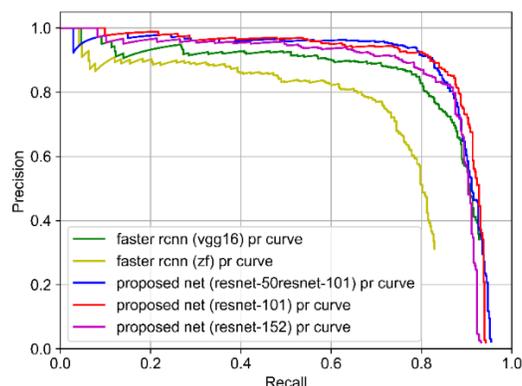
	Number of basic network layers	Training efficiency	Testing efficiency	Accuracy
Faster RCNN based on ZF	5	0.088 s	0.734 s	0.684
Faster RCNN based on VGG16	16	0.213 s	0.674 s	0.807
Proposed model based on ResNet-50	50	0.199 s	0.678 s	0.790
Proposed model based on ResNet-101	101	0.347 s	0.720 s	0.846
Proposed model based on ResNet-152	152	0.492 s	0.530 s	0.828



**Figure 12:** Feature map

To further illustrate the influence of model depth on performance, feature maps with different layers were extracted. Fig. 12(a) is derived from the last feature map of faster RCNN based on VGG16. Fig. 12(b) shows the feature map taken from a middle layer of R-FCN based on ResNet-101, and Fig. 12(c) is taken from the last layer of R-FCN based on ResNet-101. Comparing Figs. 12(a) and 12(c), it can be found that the R-FCN model has more feature maps and more obvious crack features. Therefore, this model can achieve higher accuracy. Comparing Figs. 12(b) and 12(c), it can be seen that with the same model, relatively shallow layers often learn fewer features, and with the increase of network depth, the crack area becomes clearer. Therefore, network depth has a crucial impact on accuracy.

The PR curves shown in Fig. 13 are still used to evaluate CIDB detection. With the precision rate of 89.1%, the proposed model based on ResNet-101 had the highest recall rate, 84.3%. The corresponding figure for the proposed model based on ResNet-152 and ResNet-50 is 76.87% and 82.9%, respectively, and faster RCNN based on VGG16 had a 70.8% recall rate, while Faster RCNN based on ZF had a recall rate of only 16.1%. This means that with the same precision rate, other models will mistakenly identify some cracks as non-cracks. This will overlook pavement that should be repaired preventively, resulting in greater economic losses. The red curve of the proposed model based on ResNet-101 is closer to the upper-right corner of the figure, and contains a larger area than other models. So, it can be considered that this model has better identification results.



**Figure 13:** Precision-recall curve

## 4 Discussion

### 4.1 Online hard example mining (OHEM)

Due to the use of the fully convolutional neural network architecture, OHEM technology can be easily integrated into training. During forward propagation, RPN calculates the loss of 300 region proposals. Only the region proposals with the highest loss are backpropagated, so as to obtain a lower training loss and higher accuracy. When OHEM is used, the accuracy of results can be increased by 1.8 percentage points without increasing the training time and test time, as shown in Tab. 3. Fig. 14 compares the PR curve with and without OHEM. Under the same precision rate of 88.9%, the recall rate is 82.4% without OHEM and 85.3% with OHEM. This shows that OHEM improves the model's accuracy and recall rate.

**Table 3:** Performance of OHEM

	Accuracy	Training efficiency	Testing efficiency
Proposed model without OHEM	0.846	0.347 s	0.720 s
Proposed model with OHEM	0.864	0.348 s	0.712 s

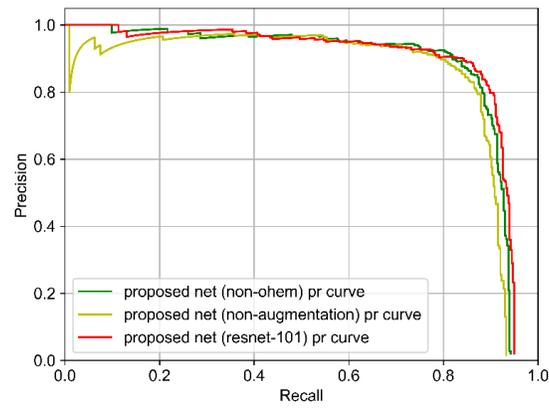
### 4.2 Data-augmentation

Data augmentation refers to the enhancement of data volume through rotation, translation, scaling, contrast enhancement, mirroring, and other operations on existing images. Due to the dark color of the road image, the effect of contrast enhancement is not obvious. In this paper, we rotated the training set image by only 90°. Tab. 4 shows that when OHEM is used, the training model accuracy without data augmentation is only 84.1%, and the accuracy is increased by 2.3 percentage points with data augmentation, with no decrease in training and testing efficiency. This is because data augmentation does not change the number of iterations and test images, and the training and test times do not increase. Fig. 14 compares the PR curve with and without data augmentation. When the precision is 88.9%, the recall rate without data augmentation is only 77.6%,

which is 7.7% less than with dataset enhancement. Data augmentation increases the number of samples and improves the model recognition effect.

**Table 4:** Performance of data augmentation

	Accuracy	Training efficiency	Testing efficiency
Proposed model without data augmentation	0.841	0.353 s	0.628 s
Proposed model with data augmentation	0.864	0.348 s	0.712 s



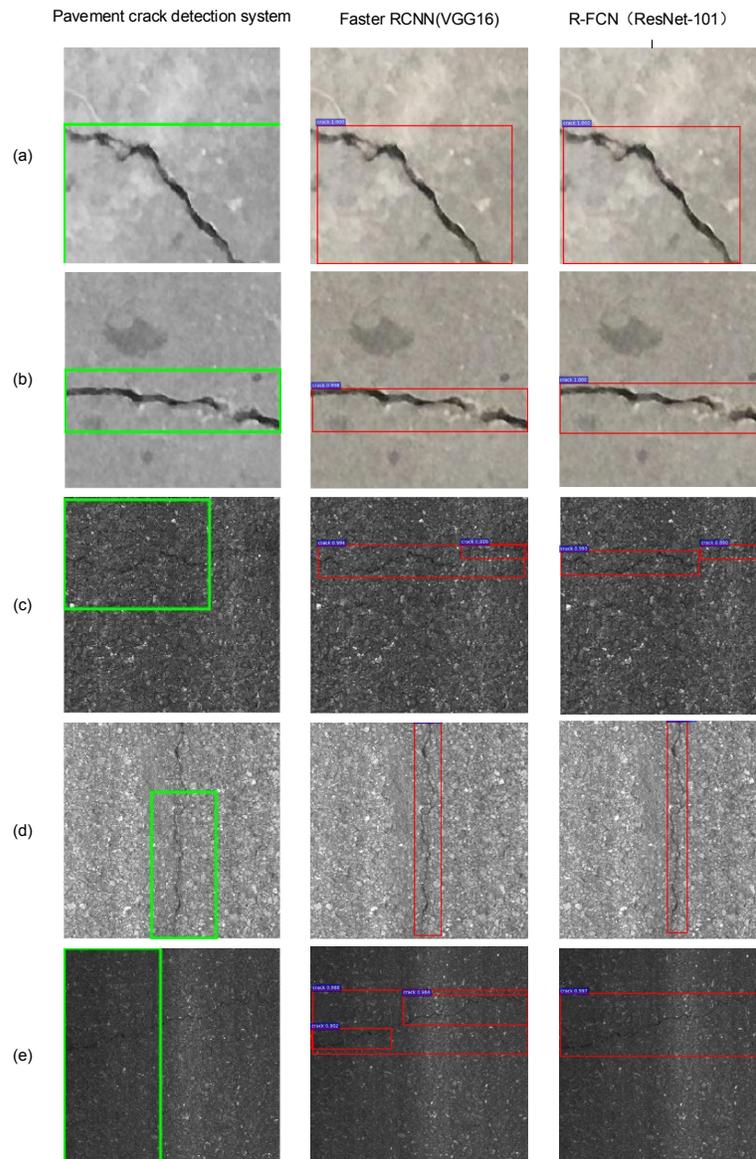
**Figure 14:** Precision-recall curve

#### 4.3 Comparison of model detection effects

To compare the performance of the new and traditional methods, the test sets of CCIC and CIDB were detected by the pavement-crack detection system, Faster RCNN, and the proposed model. The pavement-crack detection system is a traditional method of histogram equalization and binary processing. Figs. 15(a) and 15(b) show the detection effects of the three models on the CCIC. The traditional method identifies large regions and fails to effectively identify crack boundaries. Cracks of Figs. 15(c), 15(d), and 15(e) is detected by CIDB. The traditional method can no longer completely identify the crack. The images in (c) are of pavement detection under normal light. The traditional algorithm detects the left crack with a large detection area, and fails to detect the right crack. Faster RCNN cannot effectively identify the boundary between the left and right cracks. The proposed model effectively separates the two cracks and completely identifies them. The brightness of the image collected by the detection vehicle is often inconsistent, and the traditional method is not sensitive to light. The images in (d) are the results detected under bright light. The traditional algorithm detects only partial cracks. The detection results of Faster RCNN and the proposed model are similar, but the predicted box of the proposed model is closer to the crack boundary. The images in (e) are the results detected under dark light. The traditional method wrongly detects the dark part as a crack, and Faster RCNN identifies the crack as two segments, generating three prediction boxes.

The proposed model generates a more suitable prediction box.

The above comparison shows that: 1. the traditional method of manually extracting crack features can only detect cracks through one or a few of the features, such as brightness, color, and boundary. The crack features extracted by CNN are more accurate and complete, for a better detection effect. 2. The proposed model has strong robustness to the change of light and dark, and can be better used in practical detection. 3. The cracks in the public dataset are obvious and simple, and the model with fine detection on CCIC cannot be used in practical applications.



**Figure 15:** Recognition effects of different models

#### **4.4 Robustness**

Robustness is an important basis for judging model performance. In object detection, robustness refers to the model's ability to recognize an object with translation, scale scaling, blur processing, and deformation processing. The robustness of the proposed model is demonstrate and evaluated with respect to translation, fuzzy processing, scale transformation, and deformation of the image.

##### *4.4.1 Translation processing*

As shown in Fig. 16(a), these pictures are cut from the upper-left, lower-left, upper-right, and lower-right corner of the same original image; i.e., the position of the same crack on the image is shifted. The image size is  $400 \times 400$  pixels. All cracks in the figure are identified, and the translation of a crack has no impact on its detection. Compared with the image-level detection model designed by Zhang et al. [Zhang, Yang, Zhang et al. (2016)], our proposed model realizes region-level detection. In the actual detection of pavement, it is not necessary to consider the problem of multiple cracks or the location movement of cracks in a picture.

##### *4.4.2 Fuzzy processing*

As shown in Fig. 16(b), the resolution of the same image was directly reduced. From left to right, the resolutions are  $200 \times 200$ ,  $300 \times 300$ ,  $400 \times 400$  and  $512 \times 512$ . There are two cracks in each image. The left crack is fairly obvious, while the right one is somewhat fuzzy. Only the right crack in the image with a resolution of  $200 \times 200$  was not identified. Overall, this model is robust to fuzzy processing. In practice, cracks are not as obvious as those in public datasets, and the proposed model can be applied to actual problems.

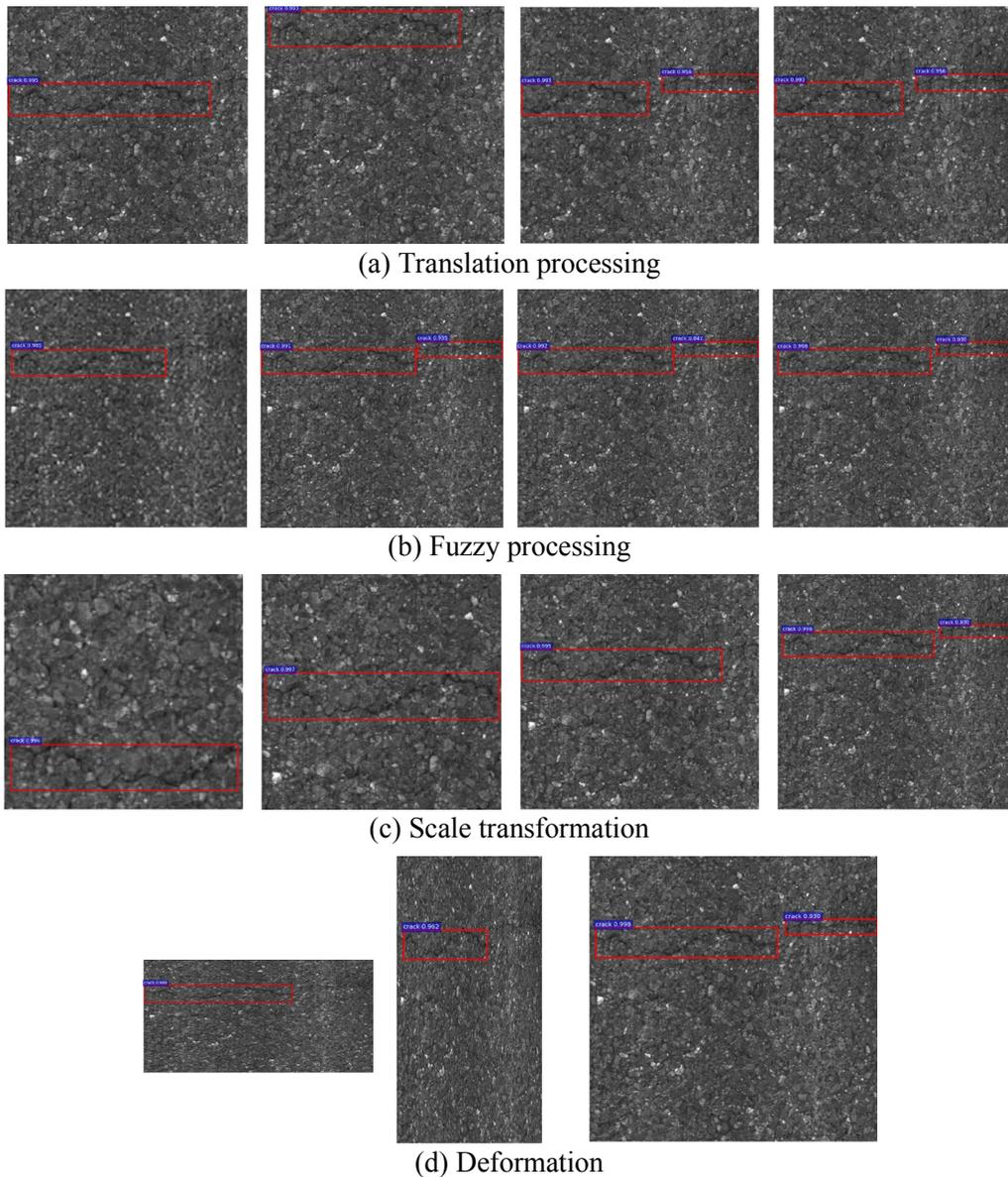
##### *4.4.3 Scale transformation*

As shown in Fig. 16(c), the same image was captured with resolutions of  $200 \times 200$ ,  $300 \times 300$ , and  $400 \times 400$  from the upper-left corner, so that the scales of the same cracks were different. All cracks were detected by our model with high confidence. The scale of cracks did not affect the detection. In actual detection tasks, the proposed model can better adapt to different pavement behavior and task requirements.

##### *4.4.4 Deformation*

As shown in Fig. 16(d), the same image data was shortened horizontally and vertically. From left to right, the resolutions were  $512 \times 256$ ,  $256 \times 512$ , and  $512 \times 512$ . The crack on the left side could be identified by our model, and that on the right side could not. This may be because the right crack is fuzzy, and after deformation processing, it is more difficult to identify. In the actual collection of pictures, if the speed is too fast, the image may be distorted. The proposed model avoids this to a certain extent.

These analysis results show that the proposed model has strong robustness and stability, which can not only perfectly identify cracks of stretching or translation, but also cracks in blurred and distorted images that are distinguished with difficulty by human eyes. Hence it can better adapt to practical problems.



**Figure 16:** Robustness analysis diagram

## 5 Conclusion

We proposed an intelligent detection model for pavement cracks using a fully convolutional neural network. Different from traditional CNNs, the proposed model consists of a shared fully convolutional basic subnetwork, RPN, position-sensitive RoI pooling, and an output layer. RPN integrates the region extraction into CNN, and position-sensitive RoI pooling solves the problem of translation invariance.

Using 1595 images of concrete cracks from the public dataset CCIC, the proposed model was trained with 20000 iterations and obtained an accuracy of 91.4%. It performed better than ConvNet, boosting and SVM in terms of PR curve. The detection in this study is region-level object detection, which is a demanding task. However, CCIC takes pictures by mobile phone, which cannot be well integrated into automatic detection. Moreover, the cracks in the dataset are too obvious, which is not consistent with the actual road detection. Therefore, 4T image data were collected by vehicle CCD camera. After selection, segmentation, and preprocessing, 3313 image data were labeled, and 2982 images were added with data augmentation, further building the Crack Image Database. Using a single high-performance GPU, the training set was trained for 30000 iterations. On the test set, the proposed model obtained 86.4% accuracy, which was better than Faster RCNN based on ZF (68.4%) and Faster RCNN based on vgg-16 (80.7%). The PR curve of the proposed model is better.

OHEM and data augmentation increased the model accuracy by 1.8% and 2.3% points, respectively. Compared with the traditional algorithm and Faster RCNN, the proposed model can extract more comprehensive and accurate crack features, which are less affected by light and shade changes. The CIDB database collected in this study is more challenging to the model and can better reflect the actual road-crack information. Finally, we tested images of translation, fuzzy processing, scale transformation, and deformation, and all cracks were still detected. The proposed algorithm has good stability. Unavoidable practical problems such as driving speed and blur have little impact on the detection results. The proposed model can be applied to pavement detection under different practical conditions, and has strong robustness in actual problems.

The CIDB of 6295 images was collected and trained to obtain the model in this study. It shows the potential of a fully convolutional neural network in pavement-crack detection. However, there are many kinds of pavement diseases, and only cracks are identified in this study. We will collect and train more kinds of pavement-disease image data to broaden the field of intelligent identification.

**Acknowledgment:** The authors are thankful for the support from the National Key Research and Development Program of China (No. 2017YFC1501200), the National Natural Science Foundation of China (No. 51678536), the Scientific and Technological Research Program of Henan Province (No. 171100310100), and the Outstanding Young Talent Research Fund of Zhengzhou University (1621323001).

**Funding Statement:** This research was funded by the National Key Research and Development Program of China (No. 2017YFC1501200), the National Natural Science Foundation of China (Nos. 51678536, 41404096). This project was supported by Department of education's Production-Study-Research combined innovation Funding-"Blue fire plan (Huizhou)" (CXZJHZ01742), the Program for Science and Technology Innovation Talents in Universities of Henan Province (Grant No. 19HASTIT043), the Outstanding Young Talent Research Fund of Zhengzhou University (1621323001). The authors would like to thank for these financial supports.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- Ai, D.; Jiang, G.; Kei, L.; Li, C.** (2018): Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods. *IEEE Access*, vol. 6, pp. 24452-24463.
- Albert, A.; Nii, A.** (2008): Evaluating pavement cracks with bidimensional empirical mode decomposition. *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 1-7.
- Anitescu, C.; Atroshchenko, E.; Alajlan, N.; Rabczuk, T.** (2019): Artificial neural network methods for the solution of second order boundary value problems. *Computers Materials & Continua*, vol. 59, no. 1, pp. 345-359.
- Avila, M.; Begot, S.; Duculty, F.; Nguyen, T. S.** (2014): 2D image based road pavement crack detection by calculating minimal paths and dynamic programming. *IEEE International Conference on Image Processing*, pp. 783-787.
- Cha, Y. J.; Choi, W.; Oral, B.** (2017): Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361-378.
- Chen, F. C.; Jahanshahi, R. M. R.** (2017): NB-CNN: deep learning-based crack detection using convolutional neural network and naïve bayes data fusion. *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392-4400.
- Christian, K.; Kristina, G.; Varun, K.; Burcu, A.; Paul, F.** (2015): A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196-210.
- Cubero-Fernandez, A.** (2017): Efficient pavement crack detection and classification. *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, 1.
- Dai, J.; Li, Y.; He, K.; Sun, J.** (2016): R-FCN: object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems 29*, pp. 370-387.
- Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K. et al.** (2009): ImageNet: a large-scale hierarchical image database. *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 248-255.
- Donald, N. G.** (1996): *Cost-Effective Preventive Pavement Maintenance*. National Academy Press.
- Eduardo, Z.; Jaime, G.; Roberto, M.; José, L.** (2014): Road crack detection using visual features extracted by gabor filters, *Computer-Aided Civil & Infrastructure Engineering*, vol. 29, no. 5, pp. 342-358.
- Everingham, M.; Winn, J.** (2006): The PASCAL visual object classes challenge 2007 (VOC2007) development kit. *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98-136.
- Feng, H.; Xu, G. S.; Guo, Y.** (2019): Multi-scale classification network for road crack detection. *IET Intelligent Transport Systems*, vol. 13, no. 2, pp. 398-405.

- Girshick, R.** (2015): Fast R-CNN. *IEEE International Conference on Computer Vision*, pp. 1440-1448.
- Girshick, R.; Donahue, J.; Darrelland, T.; Malik, J.** (2014): Rich feature hierarchies for object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 580-587.
- Gunawan, G.; Nuriyanto, H.; Sriadhi, S.; Fauzi, A.; Usman, A.** (2018): Mobile application detection of road damage using canny algorithm. *Journal of Physics: Conference Series*, vol. 1019, no. 1, pp. 12-35.
- Guo, H.; Zhuang, X.; Rabczuk, T.** (2019): A deep collocation method for the bending analysis of kirchhoff plate. *Computers Materials & Continua*, vol. 59, no. 2, pp. 433-456.
- Haris, A. K.; Muhammad, S.; Khurram, K.** (2016): Automation of optimized Gabor filter parameter selection for road cracks detection. *International Journal of Advanced Computer Science and Application*, vol. 7, no. 3, pp. 269-275.
- He, K.; Zhang, X.; Ren, S.; Sun, J.** (2014): Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 37, pp. 1904-1916.
- He, K.; Zhang, X.; Ren, S.; Sun, J.** (2015): Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 770-778.
- Ioffe, S.; Szegedy, C.** (2015): Batch normalization: accelerating deep network training by reducing internal covariate shift. *International Conference on International Conference on Machine Learning*, vol. 37, pp. 448-456.
- Jian, Z.; Huang, P. S.; Chiang, F. P.** (2006): Wavelet-based pavement distress detection and evaluation. *Optical Engineering*, vol. 45, no. 2, pp. 409-411.
- Kirschke, K. R.; Velinsky, S. A.** (1992): Histogram-based approach for automated pavement-crack sensing. *Journal of Transportation Engineering*, vol. 118, no. 5, pp. 700-710.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.** (2017): ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, vol. 60, no. 6, pp. 84-90.
- Lecun, Y. A.; Bottou, L.; Bengio, Y.; Haffner, P.** (1998): Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324.
- Lecun, Y. A.; Bottou, L.; Orr, G. B.; Müller, K.** (1998): Efficient backprop. *Neural Networks: Tricks of the Trade*, vol. 1524, no. 1, pp. 9-50.
- Lecun, Y.; Bengio, Y.; Hinton, G.** (2015): Deep learning. *Nature*, vol. 521, no. 7553, pp. 436-444.
- Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P. et al.** (2014): Microsoft COCO: common objects in context. *Computer Vision-ECCV*, vol. 8693, pp. 740-755.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.** (2015): SSD: single shot multibox detector. *Computer Vision*, vol. 9905, pp. 21-37.
- Maas, A.; Hannun, A.; Ng, A.** (2013): Rectifier nonlinearities improve neural network acoustic models. *Proceedings of ICML Workshop on Deep Learning for Audio, Speech,*

*and Language Processing*, vol. 28.

**Nanthakumar, S. S.; Lahmer, T.; Zhuang, X.; Zi, G.; Rabczuk, T.** (2016): Detection of material interfaces using a regularized level set method in piezoelectric structures. *Inverse Problems in Science and Engineering*, vol. 24, no. 1, pp. 153-176.

**Oliveira, H.; Correia, P. L.** (2009): Automatic road crack segmentation using entropy and image dynamic thresholding. *17th European Signal Processing Conference*, pp. 622-626.

**Premachandra, C.; Premachandra, H. W. H.; Parape, C. D.; Kawanaka, H.** (2015): Road crack detection using color variance distribution and discriminant analysis for approaching smooth vehicle movement on non-smooth roads. *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 4, pp. 545-553.

**Rajalakshmi, M.; Rengaraj, R.; Mukund, B.; Kumar, A.; Naren, A. et al.** (2018): An ensemble based hand vein pattern authentication system. *Computer Modeling in Engineering & Sciences*, vol. 114, no. 2, pp. 209-220.

**Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A.** (2015): You only look once: unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788.

**Ren, S.; He, K.; Girshick, R.; Sun, J.** (2015): Faster R-CNN: towards real-time object detection with region proposal networks. *International Conference on Neural Information Processing Systems*, vol. 39, no. 6, pp. 1137-1149.

**Salman, M.; Mathavan, S.; Kamal, K.; Rahman, M.** (2013): Pavement crack detection using the gabor filter. *16th International IEEE Annual Conference on Intelligent Transportation Systems*, pp. 2039-2044.

**Shabalov, A. L.; Feldman, M. S.; Bashirov, M. Z.** (2012): Automatic detection and classification of defect on road pavement using anisotropy measure. *European Signal Processing Conference*, pp. 617-621.

**Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z.** (2016): Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-12.

**Simonyan, K.; Zisserman, A.** (2014): Very deep convolutional networks for large-scale image recognition. *Computer Science*, pp. 1-14.

**Slamet, R.; Fiddin, Y. A.; Chayadi, O.; Kamarul, H. G.** (2016): Road surface crack detection using wavelets features extraction technique. *Indian Journal of Science and Technology*, vol. 9, no. 47.

**Somin, P.; Seongdeok, B.** (2019): Patch-based crack detection in black box images using convolutional neural networks. *Journal of Computing in Civil Engineering*, vol. 33, no. 3.

**Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z.** (2016): Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 2818-2826.

**Szeliski, R.** (2010): *Computer Vision: Algorithms and Applications*. Springer-Verlag, Germany.

**Tomiyama, K.; Kawamura, A.** (2016): Application of lifting wavelet transform for

pavement surface monitoring by use of a mobile profilometer. *International Journal of Pavement Research and Technology*, vol. 9, no. 5, pp. 345-353.

**Wang, K. C. P.; Zhang, A.** (2017): Deep learning for asphalt pavement cracking recognition using convolutional neural network. *Airfield and Highway Pavements*, pp. 166-177.

**Wang, K. C. P.; Li, Q.; Gong, W. G.** (2007): Wavelet-based pavement distress image edge detection with a trous algorithm. *Transportation Research Record*, no. 2024, pp. 73-81.

**Wang, S.; Yang, F.; Cheng, Y.; Yang, Y.; Wang, Y.** (2018): Adaboost-based crack detection method for pavement. *IOP Conference Series: Earth and Environmental Science*, vol. 189, no. 2, 022005.

**Yu, Q.; Guo, J.; Wang, S.; Zhu, Q.; Tao, B.** (2012): Study on a new bridge crack detection robot based on machine vision. *International Conference on Intelligent Robotics & Applications*, vol. 7506, pp. 174-184.

**Zhang, A.; Li, Q.; Wang, K. C. P.; Qiu, S.** (2013): Matched filtering algorithm for pavement cracking detection. *Transportation Research Record*, vol. 2367, no.1, pp. 30-42.

**Zhang, A.; Wang, K. C. P.; Li, B.; Yang, E.; Dai, X. et al.** (2017): Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer Aided Civil & Infrastructure Engineering*, vol. 32, no. 10, pp. 805-819.

**Zhang, C.; Nanthakumar, S. S.; Lahmer, T.; Rabczuk, T.** (2017): Multiple cracks identification for piezoelectric structures. *International Journal of Fracture*, vol. 206, no. 2, pp. 151-169.

**Zhang, C.; Wang, C. X.; Lahmer, T.; He, P.; Rabczuk, T.** (2016): A dynamic XFEM formulation for crack identification. *International Journal of Mechanics and Materials in Design*, vol. 12, no. 4, pp. 427-448.

**Zhang, C.; Zhao, J.; Rabczuk, T.** (2018): The interface strength and delamination of fiber-reinforced composites using a continuum modeling. *Composite Part B: Engineering*, vol. 137, pp. 225-234.

**Zhang, L.; Yang, F.; Zhang, D.** (2016): Road crack detection using deep convolutional neural network. *IEEE International Conference on Image Processing*, pp. 3708-3712.

**Zhao, J.** (2018): Distance control algorithm for automobile automatic obstacle avoidance and cruise system. *Computer Modeling in Engineering & Science*, vol. 116, no. 1, pp. 69-88.

**Zou, Q.; Cao, Y.; Li, Q.; Mao, Q.; Wang, S.** (2012): Cracktree: automatic crack detection from pavement images. *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227-238.