IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 10, 2021, accepted June 23, 2021, date of publication June 28, 2021, date of current version July 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3093210

A Real-Time Bridge Crack Detection Method Based on an Improved Inception-Resnet-v2 Structure

JINKANG WANG, XIAOHUI HE, SHAO FAMING, GUANLIN LU, HU CONG, AND QUNYAN JIANG

Department of Mechanical Engineering, College of Field Engineering, Army Engineering University of PLA, Nanjing 210007, China

Corresponding author: Xiaohui He (gcbhxh@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671470, and in part by the Key Research and Development Program of China under Grant 2016YFC0802900

This work did not involve human subjects or animals in its research.

ABSTRACT Bridge crack detection is essential to ensure bridge safety. The introduction of deep learning technology has made it possible to detect bridge cracks automatically and accurately. In this study, the Inception-Resnet-v2 algorithm was systematically improved and applied to the real-time detection of bridge cracks. We propose an end-to-end bridge crack detection model based on a convolutional neural network. This model combines the advantages of Inception convolution and residual networks, broadening the network width and alleviating the training problem of the deep network. The calculation speed is improved while still ensuring accuracy. Multi-scale feature fusion enables the network to extract contextual information of different scales, which improves the accuracy of crack recognition. The GKA (K-means clustering method based on a genetic algorithm) realizes the accurate segmentation of the target area, greatly enhances the clustering effect, and effectively improves the detection speed. In this model, large fracture datasets are used for training and testing without pre-training. The experimental results show that the performance of this method was improved in all aspects: accuracy, 99.24%; recall, 99.03%; F-measure, 98.79%; and FPS(Frames Per Second), 196.

INDEX TERMS Bridge crack detection, inception-resnet-v2, multiscale feature fusion, GKA.

I. INTRODUCTION

With the development of economic construction, China's road and bridge industry has undergone rapid progress. Modern bridge structures are mostly made of concrete. As time goes by, cracks of different shapes and degrees often form on these concrete surfaces, as shown in Figure 1. Cracks seriously affect the health of bridge structures, and can even endanger the safety of pedestrians [1]. Timely and accurate detection of crack inception and propagation can effectively avert catastrophic accidents [2]. Therefore, crack detection plays an important role in bridge health monitoring and reliability maintenance.

Traditional manual inspection methods are time-consuming and laborious, and cannot be widely evaluated. Moreover, carrying out manual inspections can pose a threat to the safety of inspectors, due to traffic hazards. Many transportation departments have created automatic data collection, detection, and evaluation systems. Classical bridge pavement disease detection systems [3], such as the GERPHO system in France [4], Komatsu system in Japan [5], and ZOYON-RTM intelligent pavement detection system developed in China [6], have contributed to the automatic detection results for pavement cracks becoming more accurate and reliable. Although these detection systems have unique advantages in the detection of their respective objects, most of them are manually operated and must meet specific detection conditions, such as particular brightness threshold. These systems can only identify the existence of cracks; they cannot locate cracks [7].

In recent years, machine learning and computer vision have been applied to crack detection [8], and promising results have been achieved. Wang [9] used an algorithm based on mathematical morphology and image fusion to solve the problem of crack detection in strip steel.

The associate editor coordinating the review of this manuscript and approving it for publication was Taous Meriem Laleg-Kirati.

FIGURE 1. Examples of bridge cracks of different shapes: (a) vertical crack, (b) transverse crack, (c) oblique crack, (d) crack with noise, and (e) reticulation crack.

Chambon *et al.* [7] solved the problem of automatic detection and evaluation of road cracks by using a computer vision algorithm.

Since Hinton *et al.* [10] published a paper on deep learning in the journal Science in 2006, deep learning technology has been extensively studied, which has provided new methods for solving the problem of bridge crack detection. In the past few years, many scholars have applied algorithms of convolution neural networks to crack detection, which has greatly improved the efficiency and accuracy of crack detection [11]. Wang and Hu [12] proposed application of the neural network method to pavement crack detection, and used the principal component analysis (PCA) method to calculate and analyze the data. This method significantly improved the efficiency of computer crack identification. Cha *et al.* [13] proposed a vision-based crack detection method, which used a convolutional neural network to identify the deep structure with higher accuracy. Chaiyasarn *et al.* [14] presented a combined model of a neural network and a support vector, which significantly improved the detection accuracy and recognition speed of cracks.

In actual road environments, the shape, size, and background information of bridge cracks are not always ideal. Additionally, there are many factors that can create challenging conditions for bridge crack detection using a computer system, such as uneven illumination, oil pollution, and bad weather.

Based on the analysis of previous research on crack detection, our strategy was to identify the features of cracks based on the improved Inception-Resnet-v2 algorithm [15]. Our main contributions are as follows.

Using Inception-Resnet-v2 [15] as the backbone network for crack feature recognition, the accuracy and speed of crack detection were both improved. To the best of our knowledge, we introduced Inception-Resnet-v2 into the crack detection field for the first time.

In order to solve the problem of effective information loss after deep convolution, a multi-scale feature fusion method was introduced. We fused the feature maps of different convolution layers, which greatly improved the feature expression of small targets.

The K-means clustering algorithm, based on a genetic algorithm (GKA), was introduced to enhance the clustering effect, and further improve the crack detection rate.

The rest of our paper is organized as follows. Section 2 reviews the research methods of published papers on crack detection using deep learning methods. Section 3 illustrates the network model and innovations used in this paper. Section 4 verifies the effectiveness of our method in improving the comprehensive performance of crack detection with experimental results. Section 5 summarizes the findings, and determines the direction of future work.

II. RELATED WORK

Traditional machine learning [16] aims to discover the laws and patterns contained in a large amount of training data by learning, and then make predictions for new data. Deep learning is an important branch of machine learning. Due to rapid developments in the field of deep learning in the last ten years, several representative algorithms have been proposed. Because of its ability with regard to feature learning and feature expression, deep learning has gradually replaced machine learning algorithms as the mainstream method in the crack detection field [17].

In 2012, Alex Krizhevsky *et al.* [18] proposed an AlexNet network model with five convolution layers (convolution + nonlinear activation + maximum pooling layer) and three fully connected layers. For the first time, this network solved the gradient divergence problem by using the Rectified Linear Unit (ReLU), and proposed the use of the Dropout algorithm in the fully connected layer to avoid over-fitting. In 2014, VGGNet [19] emerged, which included network models with depths ranging from 11 to 19 layers. Among the models, VGG16 and VGG19 were the most commonly used. All the model structures adopted five convolution layers, three full connection layers, and softmax output layers. This type of model verified that the increase in depth was beneficial to the improvement of detection accuracy. In the same year, GoogLeNet [20] proposed the Inception structure [21] based on the idea of the ''Network in Network'' [22], which realized the approximate replacement of the optimal local coefficient structure with dense components. Using a large number of 1×1 convolution kernels in the Inception structure greatly reduced the number of parameters, which improved the training speed and generalization ability of the model. In addition, two auxiliary classifiers were added to the model to conduct the gradient forward, which effectively reduced

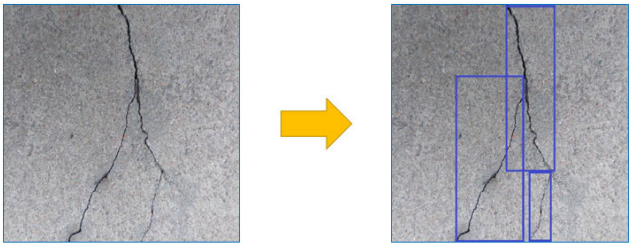


FIGURE 2. Crack detection method based on target detection: judging whether it contains a crack or not and using boundary box to locate the crack in the image.

the phenomenon of gradient disappearance. Inception-v2 [23] changed the 5 × 5 convolution kernel into two 3 × 3 convolution kernels. On the premise of ensuring the same effect, it used convolution integral solution to reduce the parameter quantity and speed up the calculation. Inception-v3 [24] put forward the decomposition idea, which divided the n × n convolution kernel into n × 1 and 1 × n convolution kernels, deepening the network and increasing the nonlinearity of the network. In 2015, ResNet [25] introduced the idea of directly bypassing the input information to the output, and changed the direct learning target value into learning the residual value between the input and the output. To some extent, this skip connection structure solved the problem of information loss and consumption, and simplified the difficulty of the learning target. In 2016, Szegedy C *et al.* put forward Inception-v4 and Inception-Resnet-v2 [15], in which the Inception-Resnet-v2 was more exquisitely designed on the basis of Inception-v4. Inception-Resnet-v2 utilized residual connections instead of filter concatenation, which not only accelerated the training, but also improved the performance. In the same year, DenseNet [26] established the connection relationship between different layers, made full use of feature, and further reduced the problem of gradient disappearance. Compared with Resnet, the training effect was very good. In 2017, Chollet F proposed a convolutional neural network architecture Xception [27] based on the deeply separable convolution layer, in which the introduction of residual connection mechanism significantly accelerated the convergence process of Xception and achieved significantly higher accuracy.

The method of bridge surface crack detection involves taking the crack as the target object [28]. According to the input image, the computer judges whether it contains a crack or not. If so, the boundary box is used to locate the crack in the image, as shown in Figure 2. The current mainstream target detection algorithms can be divided into two categories: two-stage object detection and one-stage object detection. In the two-stage detection algorithm, the algorithm is completed in two stages. Firstly, the candidate regions are extracted, and then the candidate regions are classified and further accurately located. Examples of this include Fast R-CNN [29] and Faster R-CNN [30]. However, the one-stage detection algorithm does not need to extract the candidate region, as it directly generates the class probability and position coordinate value of the object. Compared with the two-stage object detection algorithm, its detection speed is faster. Examples include SSD [31] and YOLO [32].

In 2018, Suh and Cha [33] proposed a multi-type crack detection method based on Faster R-CNN, using ZF-Net [34] instead of VGGNet in the original structure of Faster R-CNN, which accelerated the speed of feature extraction. Their experimental results showed that the improved Faster R-CNN had better robustness, and can essentially realize the real-time detection and location of various types of crack. In 2019, Li *et al.* [35] introduced a more effective and relatively simple detection method based on practical applications, which realized real-time detection of six kinds of crack with an average accuracy as high as 96.3%, by using the improved Faster R-CNN model. Mandal *et al.* [36] proposed an automatic pavement detection and analysis system based on YOLO v2, but the detection accuracy of this method needed to be improved. To solve the problem of poor real-time performance and low accuracy of crack detection, Nie *et al.* [37] proposed a crack detection method based on YOLO v3, which was improved in terms of multi-scale prediction, the basic classification network, and the classifier. Its accuracy reached 88%, meeting the requirements of civil infrastructure monitoring. In addition, the YOLO network architecture model was also applied to the detection of small target cracks in railway tracks. In 2019, Li *et al.* [38] used the improved YOLO to effectively improve the detection accuracy and real-time detection speed of track cracks. However, these methods still have some problems, such as a slow convergence speed, excessive training parameters, which makes it difficult to optimize the model.

III. OVERVIEW OF OUR METHOD

The end-to-end convolutional neural network model proposed in this paper is shown in Figure 3. The network consists of four modules, including a feature extraction backbone network based on Inception-Resnet-v2, a multi-scale context information fusion module, a GKA clustering algorithm module, and a Dropout module. In the structure shown in Figure 3, the Inception-Resnet-v2 backbone network extracts the crack features of the image, the improved Inception structure is used to increase the network width, and the introduction of the residual network is used to prevent gradient divergence. The multi-scale context information fusion module fuses the feature maps after convolution of the Stem module, Inception-Resnet-A module, Inception-Resnet-B module, and Inception-Resnet-C module, making it easier to detect small cracks. Its structure will be described in detail in Section 3.2. In addition, the GKA module can accurately identify the target area and reduce the computational complexity, improving the network detection rate. The Dropout module effectively alleviates the occurrence of over-fitting, and achieves the regularization effect to a certain extent.

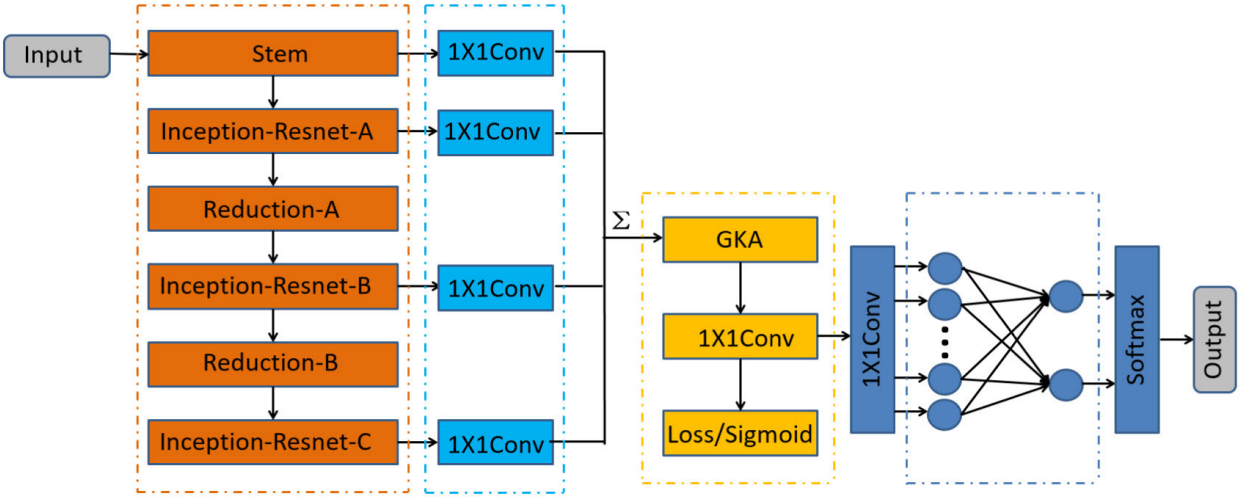


FIGURE 3. The four main components of our proposed method: Inception-Resnet-v2 baseline, Context information fusion, features Clustering and Dropout.

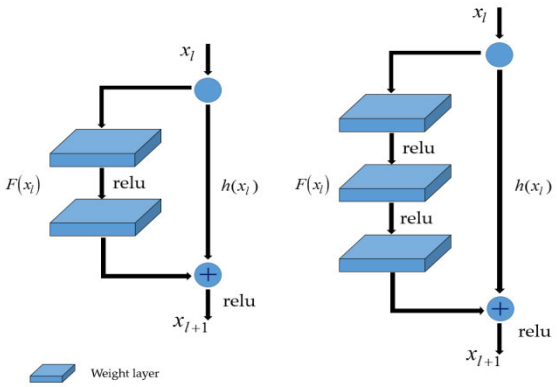


FIGURE 4. Two-layer and three-layer residual network structure.

A. INCEPTION-RESNET-V2 BACKBONE NETWORK

The Inception network structure [21] considers that multiple convolution kernels of different sizes can enhance the adaptability of the network, and extract more abundant features of different scales. At the same time, the Inception network structure can greatly reduce the parameters of the model by adopting the NIN [22] model, so that the network can reduce the number of convolution kernels as much as possible without losing model feature representation, thereby reducing the complexity of the model.

The residual network structure is shown in Figure 4. The signals of different units and layers can be directly transmitted to any layer, forward and backward, which accelerates the network training and parameter optimization. Lu *et al.* [39] proposed the Deep Coupled ResNet, which consists of a backbone network and two branch networks. The backbone network is used to identify object photos with different resolutions, and the two branch networks train high-resolution images and target images to convert them into coupled images with specific resolutions.

In the residual convolution network, the number of feature map of x_l may be different from that of, so it is necessary to use 1×1 convolution to upgrade or reduce the dimension. At this time, the residual operation is expressed as:

$$F(x_l) = w * x_l + \alpha \tag{1}$$

$$y_l = R(F) + h(x_l) \tag{2}$$

$$x_{l+1} = R(y_l) \tag{3}$$

In these equations, x_l is the input; w is the weight; α is the offset; y_l is the sum of two branches; R is the *Relu* function; $F(x_l)$ represents the convolution operation; $h(x_l)$ is a simple transformation for the input; and x_{l+1} is the final output of the residual module. *Relu* is an activation function, shown as equation (4), which is beneficial to the spread of the ladder and the prevention of divergence of the ladder, so as to prevent the ladder from becoming greatly attenuated behind the multilayer convolution [40].

$$R(x) = \max(0, x) \tag{4}$$

When $x > 0$, $R(x) = x$, and its lead is 1; when $x \leq 0$, $R(x) = 0$, with a lead of 0. In the forward calculation, we can input the value x and the threshold of 0, and obtain the output value. In the backward calculation, the gradient is 1 or 0. That is, the gradient decline is small, or does not occur. Compared with functions such as *Tanh* and the *Sigmoid*, *Relu* is simple to calculate, and has a smaller gradient decline, which is beneficial to deepening networks.

The purpose of introducing a residual network learning unit was to avoid the problem of the gradient disappearing completely when training the Inception network model. At the same time, when the performance of the network model reaches a certain saturation, the residual network layer can be mapped identically, which makes the training network faster and easier to converge. *Xi* represents the input of the

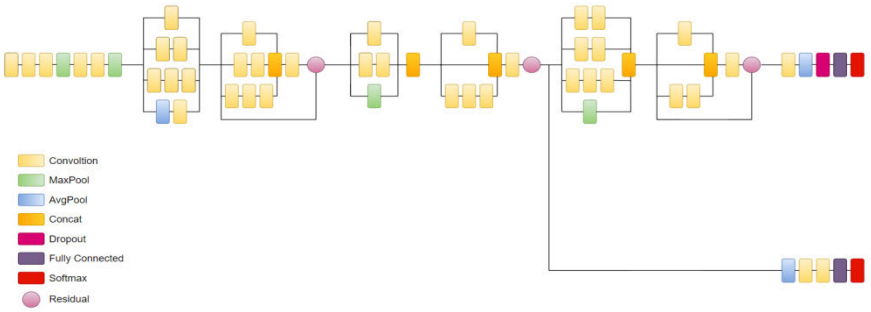


FIGURE 5. Structure of the Inception-Resnet-v2 backbone network.

ith residual unit, and Xn represents the input of the nth unit, and $F(\cdot)$ is the residual function. From the following formula for the learning characteristics from shallow i layer to deep n layer, we can know that no matter how deep the network layers are, the gradient will never approach 0.

$$\frac{\partial Xn}{\partial Xi} = \frac{\partial Xi + F(Xi, \omega i, \alpha i)}{\partial Xi} = 1 + \frac{\partial F(Xn, \omega n, \alpha n)}{\partial Xn} \quad (5)$$

In our research, three-layer residual network learning units were used. In the network of three-layer residual network learning units, 1×1 convolution is first used for dimension reduction, and then 3×3 convolution is performed. The number of network parameters of three-layer residual network units is 17.35 times less than that of two-layer residual network units.

There are shortcomings in the Inception module and its improved algorithms. The basic Inception module has a limited effect towards improving the network performance; its improved algorithms are so complex that the number of parameters and calculations can become a burden, and overfitting occurs frequently. The network has sufficient width but insufficient depth. The imbalance between width and depth leads to a low efficiency of parameter operations.

The ResNet module also has some shortcomings. Although it deepens the network and improves the classification accuracy of the network, the number of parameters and calculations increases rapidly. It is faster than the Inception module and its improved algorithms, but while the network structure is deepened, the width is narrow. The imbalance between width and depth leads to a diversity of feature extraction that is worse than that of the Inception module. If the Residual module is too complex, the training acceleration brought by the skip connection is weaker than the training deceleration brought by the sharp increase in parameters and number of calculations, which leads to training interruption or gradient explosion.

To some extent, the Inception module and the Residual module can take advantage of each other to improve the detection accuracy and reduce the number of calculations. In view of the above preliminary analysis, a fusion network called the Inception-Resnet is proposed, which is composed of the Inception module and the Residual module.

We adopted the Inception-Resnet-v2 as the backbone network of our proposed model. The structure diagrams of Inception-Resnet-v2 are shown in Figure 5.

B. MULTI-SCALE FEATURE FUSION

In the process of studying deep convolutional neural networks, researchers have found that the features extracted by shallow layers and deep layers are different. The shallow layers extract primary or intermediate features, such as edges and textures, while the deep layers extract advanced semantic features beyond human intuitive understanding. The former is beneficial to obtaining the target location, while the latter is beneficial to target detection. This conclusion is not only applicable to a certain convolutional neural network structure, but to all deep convolutional neural networks. For a bridge crack image, there are many small-scale targets in the image, which account for a few tenths or even a few hundredths of the whole image. The lack of positioning information leads to a large deviation in the positioning of small-scale targets, resulting in a decline in the overall performance of the detection. Therefore, a good deep convolution neural network should not only have the ability to distinguish the target from the surrounding environment, eliminate interference, and achieve excellent classification, but also be able to achieve good positioning to ensure detection accuracy.

Since the introduction of the Inception-Resnet-v2 module, the accuracy of detection results has been greatly improved. However, with the deepening of the network, we would inevitably lose a lot of effective positioning information, which reduced the quality of the detection results of small-scale targets. Therefore, it was necessary to introduce multi-scale feature fusion information. Our proposed multi-scale feature fusion structure uses 1×1 convolution kernels with three different sampling rates to obtain multi-scale feature information. When the sampling rate is close to the mapping feature, the 3×3 convolution kernel cannot capture the local details effectively. Therefore, the 1×1 convolution filter is used to extract the details of smaller crack edges on the bridge surface.

In the Inception-Resnet-v2 structure, we use a combination of features from the convolution of the Stem module, Inception-Resnet-A module, Inception-Resnet-B module,

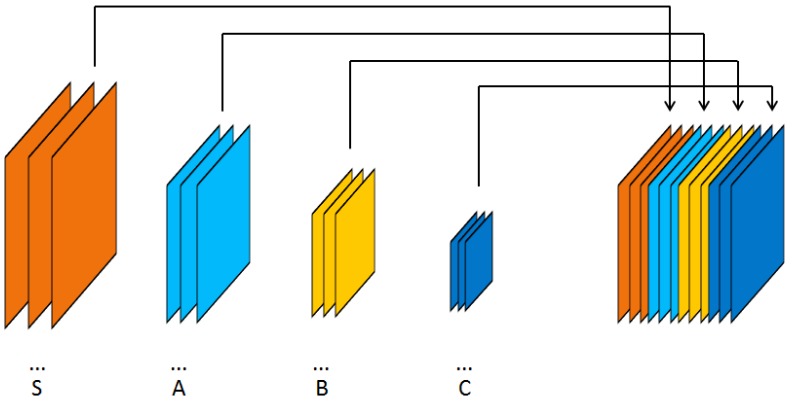


FIGURE 6. Combination of features from the convolution of the Stem module, Inception-Resnet-A module, Inception-Resnet-B module, and Inception-Resnet-C module for multi-scale feature fusion.

and Inception-Resnet-C module for feature fusion, as shown in Figure 6.

C. THE K-MEANS CLUSTERING ALGORITHM BASED ON GENETIC ALGORITHMS

With the deepening of network depth, the number of parameters becomes huge, so that the network computing capacity decreases. In order to solve the problem, we proposed to apply K-means algorithm to the model. K-means clustering algorithm is one of the classical clustering algorithms based on partition. The algorithm takes Euclidean distance as the correlation measure, and finds the corresponding cluster center vector for optimal classification, so as to minimize the evaluation index. In K-means clustering algorithm, the smaller the distance between two data points, the greater the correlation between the two data points.

The K-means algorithm has been widely popularized and applied because it is robust, simple to calculate, easy to understand, and easy to implement. However, it has a high requirement for the selection of clustering centers and easily converges to the local optimal solution, thus missing the global optimal solution. Moreover, the K-means clustering algorithm uses the Euclidean distance as the criterion of correlation between data points, which may cause distance distortion when the K-means clustering algorithm processes data points, greatly affecting the clustering results [41]. In view of this problem, our study improved the K-means clustering algorithm based on genetic algorithms [42].

Genetic algorithm, originated from biological system, is a model to simulate the process of biological evolution. This algorithm permeates and combines with natural genetics and computer. It can simulate many complex problems only by using simple bit string coding, and gradually optimize the coding structure by using simple change rules, so it has a strong global search ability.

The K-means algorithm has a strong local search ability, while genetic algorithms have a strong global search ability. Therefore, we combined them and proposed the GKA. After each generation performs the genetic operation, the operation steps of the K-means algorithm are introduced to optimize each individual in the newborn population, and the optimized individual enters the next generation of genetic operations. The specific improvement steps are as follows.

1) CODING

The floating-point coding method, based on the cluster center, is used to encode. Supposing that the cluster center is m-dimensional, the length of each chromosome is $k \times m$ for k clusters. The chromosomes are $\{x_1, x_2, \cdots x_k\}$ and $x_i = [x_{i1}, x_{i2}, \cdots x_{im}]$. The corresponding chromosome represents the coordinates of the k cluster center.

2) POPULATION INITIALIZATION

k individuals are selected from the sample space, and each individual represents an initial clustering center. The clustering centers are coded into a chromosome according to the basic coding formula. Chromosome initialization is repeated P_{size} times to generate the initial population. (P_{size} is the size of the population.)

3) FITNESS FUNCTION

After the cluster center is determined, the fitness value of each population is calculated as follows:

$$f = \frac{within(k)}{between(k)} \tag{6}$$

4) SELECT OPERATION

The mixed selection operator, combining the rotation gambling and the optimal insurance strategy, is used to calculate the probability of individual selection based on the fitness $f(x_i)$ $(i = 1, 2, \ldots, P_{size})$.

$$P(x_i) = \frac{f(x_i)}{\sum_{i=1}^{P_{size}} f(x_j)} \tag{7}$$

According to the calculated selection probability, the rotation gambling method is used to select individuals to participate in crossover and mutation operations, to generate a new population. The fitness value of each chromosome in the new population is calculated, and the individual with the highest fitness recorded in the previous generation is used to replace the individual with the lowest fitness at present. In this way, the next generation population emerges.

5) CROSS OPERATION

Arithmetic crossover between two individuals, x_1 and x_2, is performed. The new individuals are:

$$x_1' = ax_1 + (1-a)x_2 \tag{8}$$

$$x_2' = ax_2 + (1-a)x_1 \tag{9}$$

where a is a constant. According to the probability of intersection $P(x_i)$, an intersection position j is chosen, and the next generation of individuals x_1' and x_2' are obtained through the intersection.

6) MUTATION OPERATION

The variation points in individual coded strings are specified, and the value range of each gene point $[U_{\min}, U_{\max}]$ is determined. For each variation point, a random number from the corresponding gene value range is taken with mutation probability $P(m)$ to replace the original value. The new gene is:

$$x_i = U_{\min} + \theta(U_{\max} - U_{\min}) \tag{10}$$

θ is a random number within a (0, 1) circle.

When the same optimal individual fitness value appears continuously and exceeds a certain threshold, the algorithm stops running. The algorithm ends, and outputs the final clustering result. The schematic flow diagram of the improved algorithm is as follows:

IV. EXPERIMENTAL RESULTS

A. DATASET

We used three crack datasets as input samples: the CCIC dataset [43], SDNET dataset [44], and OCD dataset. The images in the CCIC dataset have been collected from various concrete buildings, including 40,000 RGB images with a resolution of 227 × 227 pixels, are divided into negative (non-crack) and positive(crack) categories. The SDNET dataset contains more than 56,000 cracked and non-cracked images of concrete bridge surfaces, walls, and sidewalks, and the dataset includes cracks ranging from 0.06 to 25 mm in width. The dataset also includes images with various obstacles, including shadows, surface roughness, scaling, edges, holes, and nicks. The images in the OCD dataset were captured by us, and are of concrete bridge decks and pavements. Similarly to the above two datasets, the OCD dataset contains 2086 images, which are also divided into cracked and non-cracked images. These images were taken in the daytime, night-time, and in different weather conditions, so the images in this dataset are closer to the reality of bridge pavements.

TABLE 1. BSCD image samples from the different datasets.

Number of images	CCIC		SDNET		OCD	
	Crack	Non-crack	Crack	Non-crack	Crack	Non-crack
Train	4000	4000	5000	5000	600	600
Validate	2000	2000	3000	3000	300	300
Test	1000	1000	1000	1000	100	100

Convolutional neural networks extract and learn image features via convolution operations, using massive datasets. Classification errors on training sets are minimized by back propagation to optimize the network parameters together, before finally realizing crack extraction [45]. Therefore, the performance of the convolutional neural network is directly related to the size of the datasets. We randomly selected 14,000 images from CCIC, 18,000 images from SDNET, and 2000 images from OCD to combine into a larger crack detection dataset [46], called the Bridge Surface Crack Dataset (BSCD). Image samples from this database, made up of several different datasets, are shown in Table 1.

In order to investigate the rationality of the BSCD for our proposed method, each dataset was used to train and validate the model under the same conditions. Then, the images from CCIC, SDNET, OCD, and BSCD were tested by the trained models. To form a control experiment, we selected equal numbers of samples in all crack images and non-crack images The performance of each dataset was evaluated in terms of the accuracy of crack detection. The results are shown in Table 2.

As shown in Table 2, the model trained by each dataset achieves the best accuracy when testing its own images. The model trained by CCIC can detect test images in CCIC with 97.91% accuracy. The model trained by SDNET can detect test images in SDNET with 91.05% accuracy. Accordingly, OCD has an accuracy of 96.37% for its own trained model. However, good detection accuracy cannot be achieved when using a model trained by one dataset to test the images of another dataset. The model trained by CCIC has an accuracy rate of 63.16% for SDNET images and 69.15% for OCD images. The model trained by SDNET has an accuracy rate of 61.32% for CCIC images and 75.19% for OCD images. Correspondingly, the model trained by OCD has an accuracy rate of 59.81% for CCIC images and 71.63% for SDNET images. These results reflect that the convolutional neural network can effectively learn a sample domain to detect samples by itself, but the learned knowledge does not translate to accurately detecting cracks in other datasets.

The model trained by the BSCD dataset improved the detection accuracy of CCIC, SDNET, and OCD, as shown in Table 2. The accuracy of crack detection in CCIC reached 99.92%, which is 2.01% higher than that of CCIC itself. The accuracy with SDNET (95.17%) was 4.12% higher than that

TABLE 2. Sample test results with different training datasets.

Datasets	Accuracy			
	Train	CCIC test	SDNET test	OCD test
CCIC	97.82%	97.91%	63.16%	69.15%
SDNET	92.56%	61.32%	91.05%	75.19%
OCD	97.02%	59.81%	71.63%	96.37%
BSCD	99.24%	99.92%	95.17%	98.69%

of SDNET itself. The accuracy of OCD image detection was improved by 2.32%. These experimental results show that BSCD datasets can contain the feature knowledge of different datasets, and have higher accuracy than other separate datasets in model training. In order to visually present the influence of different datasets on crack detection accuracy, a line chart based on the data is shown in Figure 8. The model trained by the BSCD dataset is superior to the other datasets in detection accuracy.

B. EXPERIMENTAL ENVIRONMENT

Our crack detection methods were evaluated using the Pytorch software. The running platform was a Windows 10 64-bit operating system, Intel®Core i7 V6CPU@3.7 GHz, 16 GB of memory, and a single GeForce 2080 Ti GPU, with 11 GB of memory.

C. PERFORMANCE EVALUATION INDEX

In this part of the experiment, five evaluation indices [47], *Accuracy*, *Recall*, *Precision*, *FPS* and $F-measure$, were adopted to comprehensively evaluate the network performance. There are four different states of crack test results. *TP* is the number of crack samples within the sample; *TN* is the number of non-cracked samples that are correctly classified; *FP* is the number of samples that are misclassified without cracks; and *FN* is the number of samples that are misclassified with cracks. Table 3 shows a confusion matrix to illustrate these states more clearly.

Accuracy refers to the proportion of correctly classified images in all test images, which can reflect the learning situation of all test images. The calculation method is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

However, for the false part, the same accuracy may have a good error estimation effect or a poor error estimation effect. In other words, for an example that is false, the accuracy is difficult to measure. For this reason, we introduce *Recall* and *Precision*.

Recall considers the original sample. It indicates how many positive examples in the samples are correctly detected. The calculation method is as follows:

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

Precision considers the prediction result. It indicates how many of the samples predicted to be positive are true positive samples. The calculation method is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

Recall and *Precision* reflect the learning situation of positive samples and negative samples, respectively. They are sometimes contradictory; that is, when the network has a poor learning effect on negative samples, most samples will be classified as positive samples. In this case the *Recall* will be high, but the *Precision* will be very low, and vice versa. We hoped to comprehensively evaluate the learning effect of the network on training samples, so we adopted the index $F-measure$, which is a combination of *Recall* and *Precision*. It can reflect the learning situation of the network more comprehensively. The calculation method is as follows:

$$F-measure = \frac{2Recall \times Precision}{Recall + Precision} \tag{14}$$

In this study, the detection speed of the model refers to the time required for the crack detection process to be completed for each image. The speed is indicated by *FPS*, which is the number of frames detected in one second.

D. MODEL TRAINING

The basic training strategy of the proposed model is shown in Figure 9. In the training process, both the training set and the validation set are unlabeled images. We trained the model on the training set, evaluated the model on the validation set, and adjusted hyperparameters to make the model in the best state. After the model was trained, we input the original images into the trained model to get the prediction results.

In order to select the best learning rate, we compared the changes in the training loss of different learning rates (lr) with the increase in epochs. As shown in Figure 10 below, from the loss curve, we can see that when the learning rate is 0.001 in the training process before 50 epochs, the training loss of the model tends to become stable faster than at the learning rate

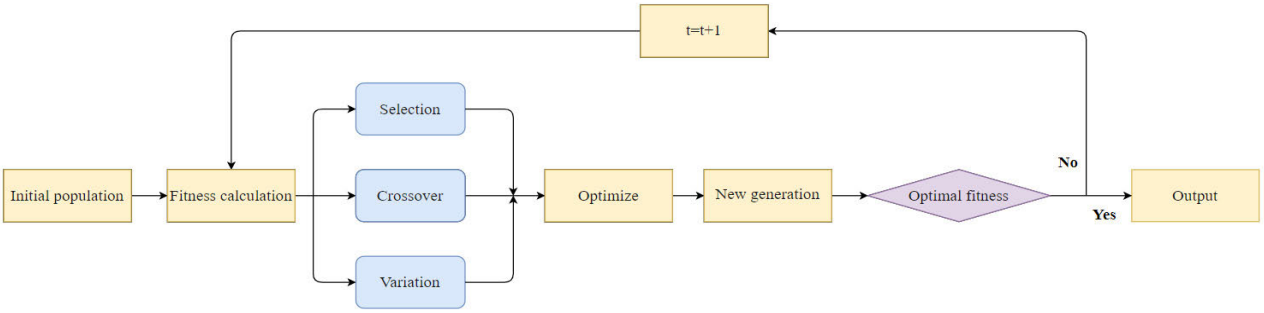


FIGURE 7. Schematic flow diagram of the improved GKA algorithm.

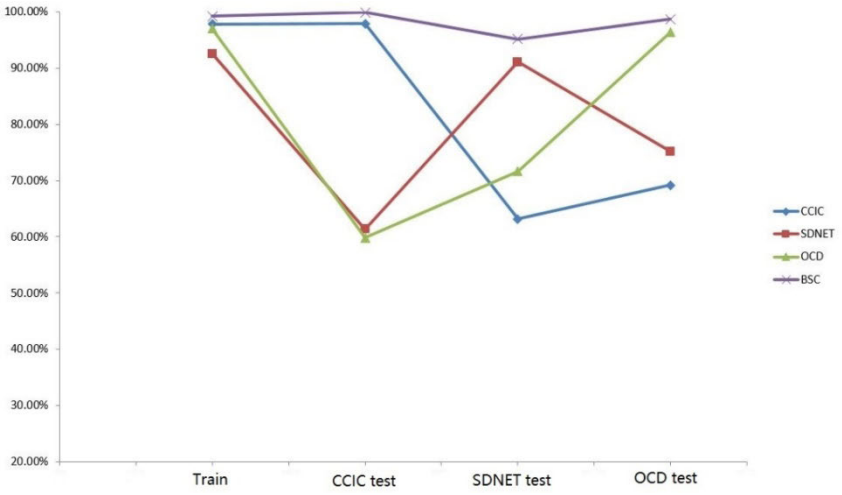


FIGURE 8. Sample test results with different training datasets.

TABLE 3. Confusion matrix of the four detection results.

Ground truth	Precision	
	Crack (True)	Non-crack (False)
Crack (True)	TP (True positive)	FN (False negative)
Non-crack (False)	FP (False positive)	TN (True negative)

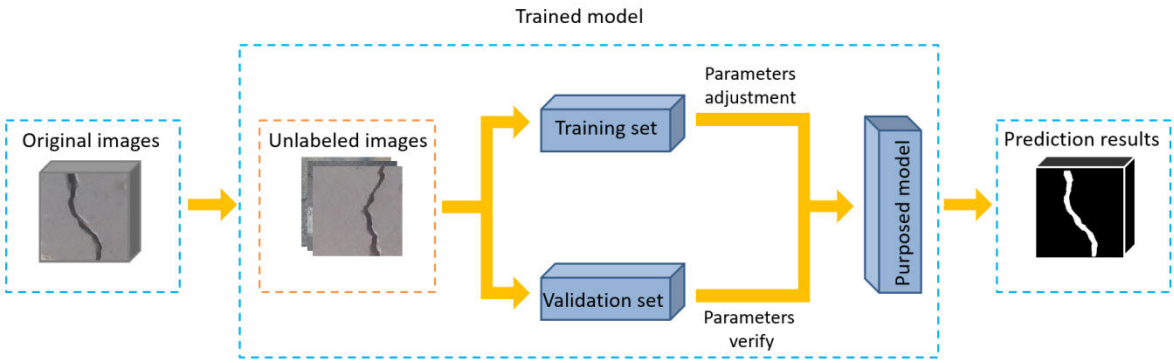


FIGURE 9. Schematic diagram of the model training strategy.

of 0.0001. After 50 epochs, the learning rate of 0.0001 leads to the training loss of the model being smaller. Therefore, combining the respective characteristics of the two different learning rates, we set the initial learning rate as 0.001. After the number of epochs reaches 50, the learning rate drops by 10 times and becomes 0.0001.

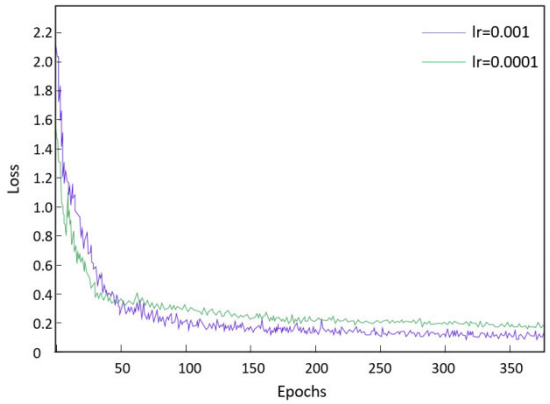


FIGURE 10. The curve of training loss with different learning rates.

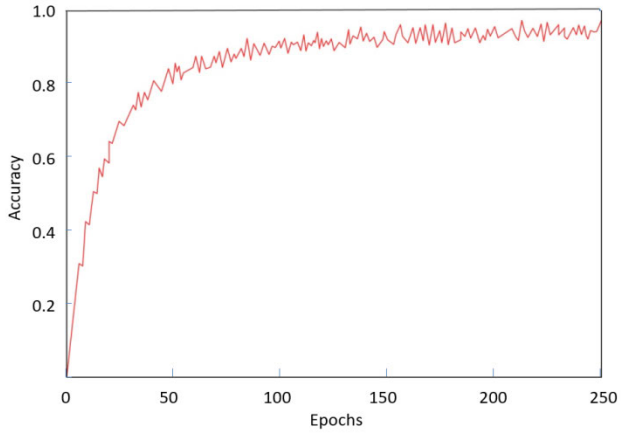


FIGURE 11. The curve of accuracy with different epochs.

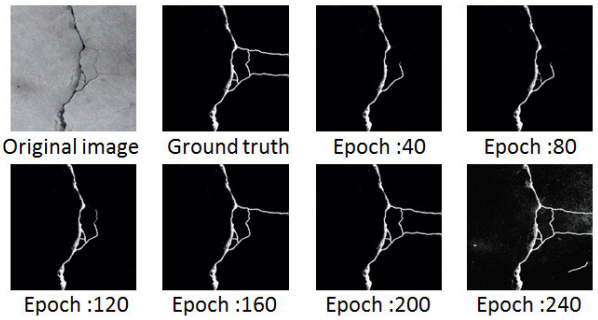


FIGURE 12. The comparison of the ground truth of original image and the output in different epochs.

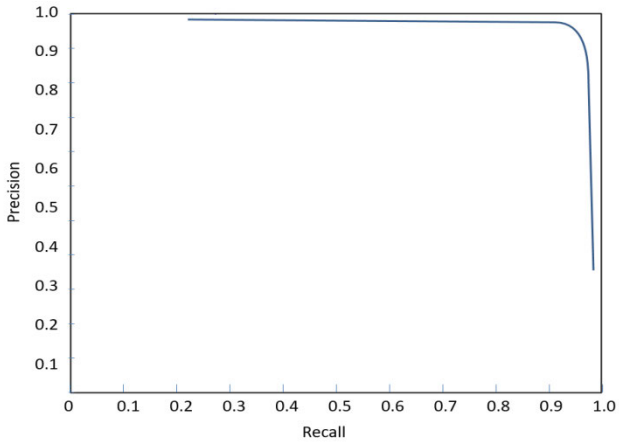


FIGURE 13. The Precision and Recall curve of the proposed model.

Selection of epoch number: As shown in Figure 10, when the number of epochs reaches 200, the training loss is basically stable. Considering the time cost of training, there is no need to continue training after 200 epochs of training. In addition, in order to avoid over-fitting, we did not proceed with higher epochs.

From the training accuracy curve shown in Figure 11, we can see that, when the epoch is around 200, the accuracy reaches its maximum. With an increase in epochs, the accuracy of detection remains essentially unchanged and enters a stable state.

In addition, we verified the selection of epoch number by comparing the outputs of different periods [48], as shown in Figure 12. When the number of epochs is small, such as 50 or 100, the model is under-fitted and the detection performance is poor. There are many tiny cracks that cannot be detected. With an increasing number of epochs, the performance of the crack detection model improves. When it reaches 200, we achieve the best training results. Conversely, the model will be overfitted when the number of epochs is too high. For example, when the epoch number is 240, non-crack images are also detected as a crack images, and a large amount of noise points will appear.

Through the above quantitative experiments, we determined the best parameters, and then crack detection could be carried out according to those parameters. The momentum optimization algorithm was used for training. The initial learning rate was set to 0.001. When the number of epochs reached 50, the learning rate dropped to 0.0001, and then remained unchanged. The momentum parameter was 0.9, the weight attenuation regular term was 0.0005, the batch size was 64, the batch was divided into 313, the epoch dropout was set to 0.5, so as to prevent over-fitting.

After the completion of model training, the *Precision* and *Recall* curve could be obtained by saving and analyzing the training record files in the process of model training, as shown in Figure 13. From the equation of the $F-measure$, we can see that when the *Precision* value is equal to the *Recall* value, the $F-measure$ reaches the maximum value of 98.79%.

E. TEST RESULTS AND ANALYSIS

The test results of the BSCD by network model are shown in Table 4. AlexNet-32 and AlexNet-256 were AlexNet trained when the mini-batch was 32 and 256, respectively. The results show that the training accuracy of the network was not reduced by using the small batch training method. It can also be seen that Xception and AlexNet performed

TABLE 4. Test results of different network verification sets.

Type	Accuracy	Precision	Recall	F-measure	FPS
Inception-v3	0.9808	0.9761	0.9857	0.9809	116
Inception-v4	0.9863	0.9795	0.9884	0.9864	134
DenseNet201	0.9837	0.9808	0.9876	0.9837	67
Xception	0.9633	0.9561	0.9710	0.9651	86
AlexNet-32	0.9667	0.9566	0.9780	0.9672	167
AlexNet-256	0.9605	0.9446	0.9769	0.9613	179
Ours	0.9924	0.9821	0.9903	0.9879	196

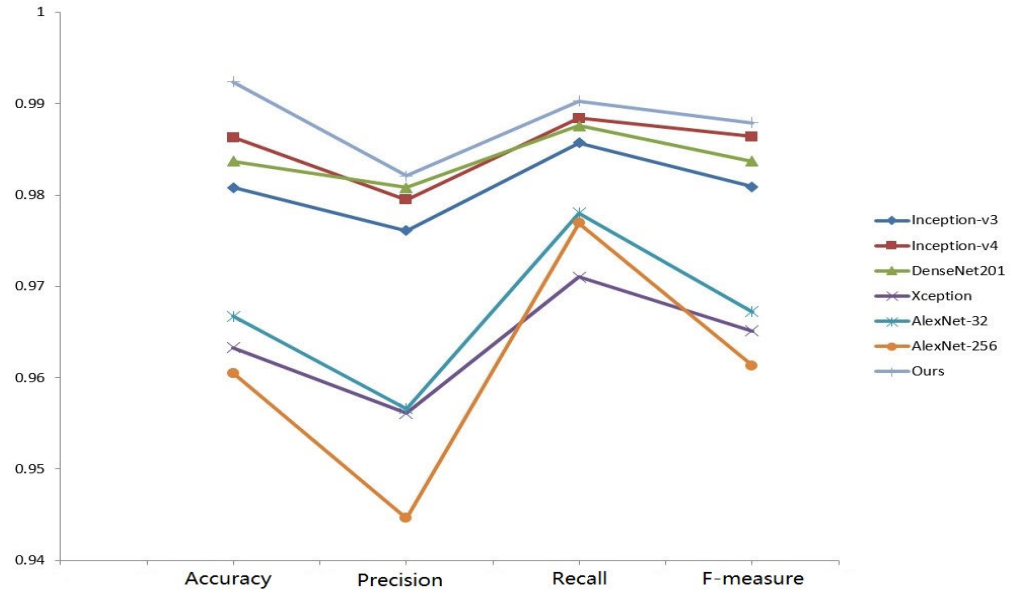


FIGURE 14. Crack detection performance analysis of different networks.

worst in crack detection accuracy under the same training conditions. AlexNet has a simple network structure, few feature extraction layers, a poor ability to learn complex features, and the shortest computing time because of its shallow network depth. The network depth of Xception is second only to DenseNet201, and its detection effect on cracks was the worst, which shows that the method of decoupling channel correlation and spatial correlation in the convolution channel is not conducive to the extraction of crack features.

This method is not adequete to perform crack detection, and does not give full play to the potential of the Xception network. DenseNet201 performed better than the other networks, because the number of feature extraction layer of DenseNet201 is the largest of all the networks. It makes full use of the feature map output with each network module, which enhances its ability to learn image features. Inception-v4 was second, and its performance on the four indices was better than that of Inception-v3. Therefore, adding convolutions of different sizes to the same layer can improve the accuracy of crack detection more effectively than simply deepening the network. At the same time, the large network depth of DenseNet201 and the large network width of Inception-v3 and Inception-v4 took more time to complete the test.

We show the test results of the different networks in Figure 14. From Figure 14, we can see that our proposed network is among the best in the four evaluation indices of *Accuracy*, *Recall*, *Recall*, and $F-$ me*asure*.

From the *FPS* values of each network, we developed a comparison diagram of the average detection time of different networks for a single image, as shown in Figure 15.

It can be seen that the average time for our network to detect a single image was the lowest, at only 0.0051 seconds, which meets the requirements of real-time detection.

Different types of image in the test set were used to test the network. These images were not touched by the network during the training process, so the test results would be closer

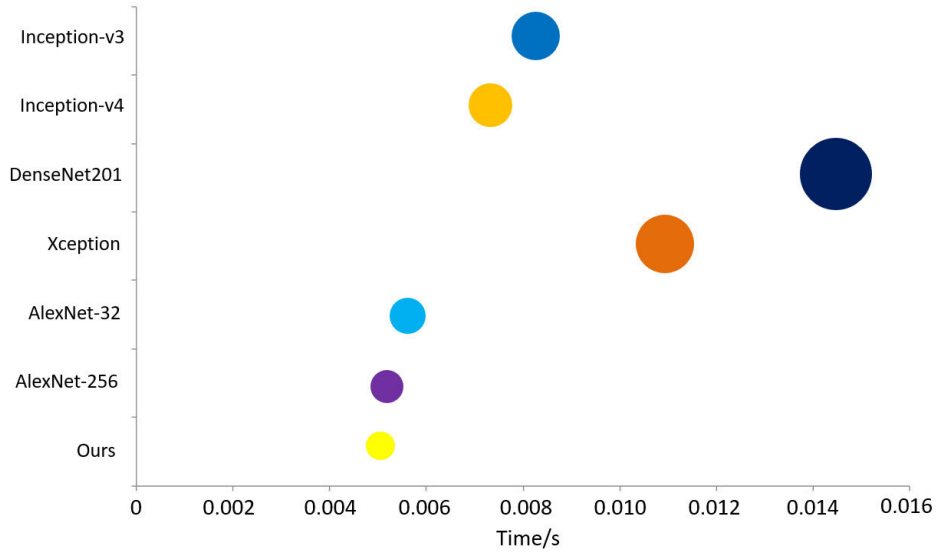


FIGURE 15. Comparison diagram of different network detection times.

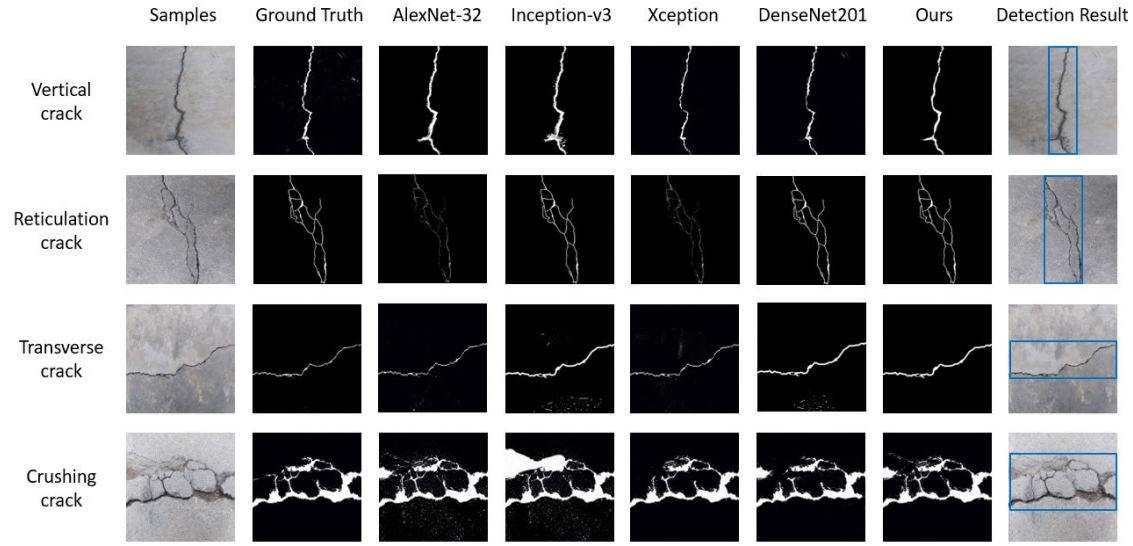


FIGURE 16. Detection results comparison of various methods on different types of cracks.

to the detection effect of different types of crack in a real situation.

Figure 16 shows the results of different types of crack samples detected by AlexNet, Inception-v3, Xception, and DenseNet201, along with our proposed method. There are vertical, reticulation, transverse, and crushing cracks. It can be seen that the detection abilities of Inception-v3, DenseNet201, and our proposed method are the best, while the AlexNet and Xception networks had different degrees of missed detections or false detections. This is consistent with the *Precision* index in Table 4. The detection performance of our method was the best. Our detection result was the closest to the ground truth.

Figure 17 shows the crack detection results under complex background conditions. The background types are shadows, weak light, speckles, road marking lines, and pebbles. This noise can affect crack detection to a certain extent. As the results show, the Inception-v3 and DenseNet201 methods had better detection performance in crack detection with speckles or road marking lines. However, when the background noise of crack images included shadows, pebbles, or weak light, the detection results of these models also had obvious defects. They showed more false detections than the other methods. When the noise included shadows or pebbles, these methods mistook most of the noise for cracks. However, Xception is not very sensitive to a

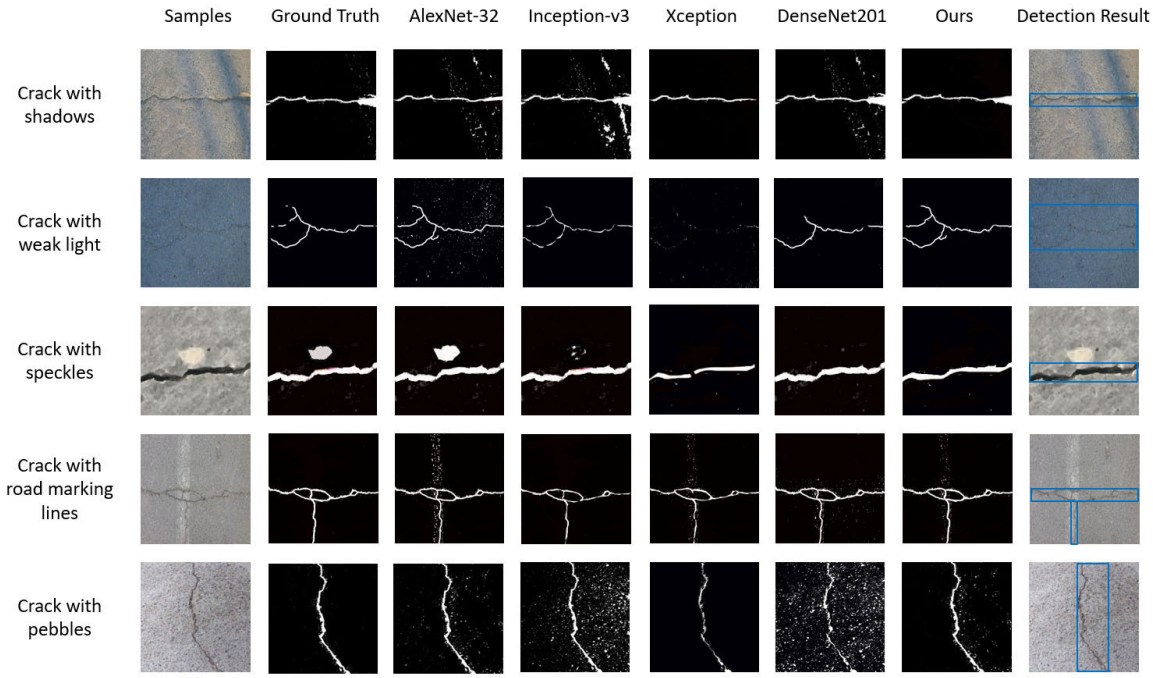


FIGURE 17. Detection results comparison with various methods for cracks in complex conditions.

complex background, and some cracks will be missed with this method. Our proposed method can detect cracks accurately and avoid most noise, such as shadows, speckles, and pebbles.

V. CONCLUSION AND FUTURE WORK

In this paper, a image classification model for crack detection was proposed, which makes use of the advantages of the Inception-Resnet-v2 module, multi-scale feature fusion method, and GKA clustering method. Our method can achieve 99.24% crack detection accuracy without pre-training. At the same time, the model can capture the multi-scale context information of the crack images, improve computational efficiency, and quickly complete crack detection on a dataset. In addition, the model can be embedded into any convolutional network as an effective feature extraction structure. The research results show that our proposed method can be used as an effective crack detection method in bridge pavement crack detection. However, different kinds of cracks have different impacts on bridge health in the actual environment. Our method can only detect cracks, but can not classify them. In future work, the classification of different kinds of cracks will be the focus of our research. For crack images with noise and indistinct background and foreground contrast, we will consider building a deep network to capture more accurate crack information, so as to avoid misjudgment and omissions.

ACKNOWLEDGMENT

(Jinkang Wang, Xiaohui He, Shao Faming, Guanlin Lu, Hu Cong, and Qunyan Jiang are co-first authors.)

REFERENCES

[1] M. Gavilán, D. Balcones, O. Marcos, D. F. Llorca, M. A. Sotelo, I. Parra, M. Ocaña, P. Aliseda, P. Yarza, and A. Amírola, "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, Oct. 2011.

[2] P. Weng, Y. H. Lu, X. B. Qi, and S. Y. Yang, "Pavement crack segmentation technology based on improved fully convolutional networks," *Comput. Eng. Appl.*, 2019.

[3] C. P. W. Kelvin and P. E. Robert, "Investigation of image archiving for pavement surface distress survey," Mack-Blackwell Transp. Center, Univ. Arkansas, Fayetteville, AR, USA, Tech. Rep., 1999.

[4] S. Wang, X. Wu, Y. H. Zhang, and Q. Chen, "Image crack detection with fully convolutional network based on deep learning," *J. Comput.-Aided Des. Comput. Curveics*, vol. 30, no. 5, pp. 859–867, 2018.

[5] L. Sjogren and P. Offrell, "Automatic crack measurement in Sweden," in *Proc. 4th Int. Symp. Pavement Surface Characteristics Roads Airfields*, 2000.

[6] F. Wang, P. Guohua, and H. Xie, "Strip steel defect detection based on morphological enhancement and image fusion," *Laser Infrared*, vol. 48, no. 1, pp. 124–128, 2018.

[7] S. Chambon and J.-M. Moliard, "Automatic road pavement assessment with image processing: Review and comparison," *Int. J. Geophys.*, vol. 2011, pp. 1–20, Jun. 2011.

[8] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, *Temporal Segment Networks: Towards Good Practices for Deep Action Recognition*. Cham, Switzerland: Springer, 2016.

[9] K. C. P. Wang, "Elements of automated survey of pavements and a 3D methodology," *J. Modern Transp.*, vol. 19, no. 1, pp. 51–57, Mar. 2011.

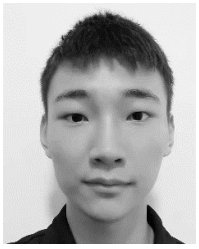
[10] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[11] S. Wu, J. Fang, X. Zheng, and X. Li, "Sample and structure-guided network for road crack detection," *IEEE Access*, vol. 7, pp. 130032–130043, 2019.

[12] X. Wang and Z. Hu, "Grid-based pavement crack analysis using deep learning," in *Proc. 4th Int. Conf. Transp. Inf. Saf. (ICTIS)*, Aug. 2017, pp. 917–924.

[13] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, May 2017.

[14] K. Chaiyasarn, "Crack detection in historical structures based on convolutional neural network," *Int. J. Geomate*, vol. 15, no. 51, pp. 240–251, Nov. 2018.
[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[17] Y.-A. Hsieh and Y. J. Tsai, "Machine learning for crack detection: Review and model performance comparison," *J. Comput. Civil Eng.*, vol. 34, no. 5, Sep. 2020, Art. no. 04020038.
[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105.
[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Comput. Sci.*, 2014.
[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1–9.
[21] J. Liu, C. Li, F. Liang, C. Lin, M. Sun, J. Yan, W. Ouyang, and D. Xu, "Inception convolution with efficient dilation search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 11486–11495.
[22] M. Lin, Q. Chen, and S. Yan, "Network in network," *Comput. Sci.*, 2013.
[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.
[25] Q. He *et al.*, "A survey of machine learning algorithms for big data," *Pattern Recognit. Artif. Intell.*, vol. 27, no. 4, pp. 327–336, 2014.
[26] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 4700–4708.
[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
[28] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, and X. Chen, "Automatic bridge crack detection using a convolutional neural network," *Appl. Sci.*, vol. 9, no. 14, p. 2867, Jul. 2019.
[29] R. Girshick, "Fast R-CNN," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *SSD: Single Shot MultiBox Detector*. Cham, Switzerland: Springer, 2016.
[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[33] G. Suh and Y. J. Cha, "Deep faster R-CNN-based automated detection and localization of multiple types of damage," in *Proc. Sensors Smart Struct. Technol. Civil, Mech., Aerosp. Syst.*, vol. 10598, Mar. 2018, Art. no. 105980T.
[34] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 818–833.
[35] J. Li, X. Zhao, and H. Li, "Method for detecting road pavement damage based on deep learning," *Smart Struct. Mater. Nondestruct. Eval. Health Monit.*, 2019.
[36] V. Mandal, L. Uong, and Y. Adu-Gyamfi, "Automated road crack detection using deep convolutional neural networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018.
[37] M. Nie and C. Wang, "Pavement crack detection based on YOLO v3," in *Proc. 2nd Int. Conf. Saf. Produce Informatization (IICSPI)*, Nov. 2019.
[38] W. Li, Z. Shen, and P. Li, "Crack detection of track plate based on YOLO," in *Proc. 12th Int. Symp. Comput. Intell. Design (ISCID)*, Dec. 2019.
[39] Z. Lu, X. Jiang, and A. Kot, "Deep coupled ResNet for low-resolution face recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 4, pp. 526–530, Apr. 2018.
[40] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853
[41] K. Deeparani and P. Sudhakar, "Efficient image segmentation and implementation of K-means clustering," *Mater. Today, Proc.*, Feb. 2021.
[42] S. Gibb, H. M. La, and S. Louis, "A genetic algorithm for convolutional network structure optimization for concrete crack detection," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2018, pp. 1–8.
[43] Ç. F. Özgenel, "Concrete crack images for classification," *Mendeley Data*, 2018.
[44] S. Dorafshan, M. Maguire, and R. Thomas, "SDNET2018: A concrete crack image dataset for machine learning applications," Tech. Rep., 2018.
[45] S. Y. Zhu *et al.*, "Method for bridge detection based on the U-Net convolutional networks," *J. Xidian Univ.*, vol. 46, no. 4, pp. 35–42, 2019.
[46] Q. Yang, W. Shi, J. Chen, and W. Lin, "Deep convolution neural network-based transfer learning method for civil infrastructure crack detection," *Autom. Construct.*, vol. 116, Aug. 2020, Art. no. 103199.
[47] H. Kim, E. Ahn, M. Shin, and S.-H. Sim, "Crack and noncrack classification from concrete surface images using machine learning," *Struct. Health Monitor.*, vol. 18, no. 3, pp. 725–738, May 2019.
[48] G. Li, X. Li, J. Zhou, D. Liu, and W. Ren, "Pixel-level bridge crack detection using a deep fusion about recurrent residual convolution and context encoder network," *Measurement*, vol. 176, May 2021, Art. no. 109171.

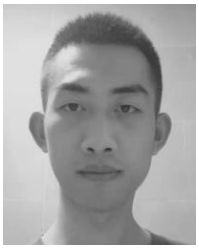


JINKANG WANG received the bachelor's degree in mechanical engineering from the Army Engineering University of PLA, China, in 2020, where he is currently pursuing the master's degree in mechanical engineering. His current research interests include mechanics, machine learning, and computer vision.

XIAOHUI HE was born in 1975. He received the Ph.D. degree from the Army Engineering University of PLA, China. He is an Associate Professor with the Army Engineering University of PLA. His research interests include mechatronics and deep learning.

SHAO FAMING was born in 1978. He received the Ph.D. degree from the Army Engineering University of PLA, China. He is currently an Associate Professor with the Army Engineering University of PLA. His research interests include signal processing, deep learning, and software engineering.

GUANLIN LU is currently pursuing the master's degree with the College of Field Engineering, Army Engineering University of PLA. His research interest includes machine learning.



HU CONG is currently pursuing the master's degree with the College of Field Engineering, Army Engineering University of PLA. His research interests include robotics and machine learning.

QUNYAN JIANG is currently pursuing the master's degree with the College of Field Engineering, Army Engineering University of PLA. Her research interest includes machine learning.