

Received May 20, 2021, accepted June 7, 2021, date of publication June 10, 2021, date of current version June 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3088292

Autonomous Crack and Bughole Detection for Concrete Surface Image Based on Deep Learning

YUJIA SUN, YANG YANG, GANG YAO[✉], FUJIA WEI, AND MINGPU WONG

School of Civil Engineering, Chongqing University, Chongqing 400044, China

Key Laboratory of New Technology for Construction of Cities in Mountain Area, Ministry of Education, Chongqing, China

Corresponding author: Gang Yao (yaocqu@vip.sina.com)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2020CDJQY-A067, in part by the National Key Research and Development Project under Grant 2019YFD1101005, and in part by the National Natural Science Foundation of China under Grant 51608074.

ABSTRACT Cracks and bugholes (surface air voids) are common factors that affect the quality of concrete surfaces, so it is necessary to detect them on concrete surfaces. To improve the accuracy and efficiency of the detection, this research implements a novel deep learning technique based on DeepLabv3+ to detect cracks and bugholes on concrete surfaces. Firstly, in the decoder, the 3×3 convolution of the feature fusion part is improved to a 3-layer depth separable convolution to reduce the information loss during up sampling. Secondly, the original expansion rate combination is changed from 1, 6, 12, 18 to 1, 2, 4, 8 to improve the segmentation effect of the model on the image. Thirdly, a weight value is added to each channel of the Atrous Spatial Pyramid Pooling (ASPP) module, and the feature maps that contribute significantly to the target prediction are learned and screened. To use this method, a database is built containing 16,662 256×256 pixel images of bugholes and cracks on concrete surfaces. The two defects included in those images are labeled manually. The DeepLabv3+ architecture is then modified, trained, validated and tested using this database. A strategy of model-based transfer learning is applied to optimize and accelerate the learning efficiency of the model. The weights and biases of the Xception part of the model are initialized by the pretrained backbones. The results are 97.63% (crack), 93.53% (bughole) Average Precision (AP), 95.58% Mean Average Precision (MAP) and 81.87% Mean Intersection over Union (MIoU). A comparative study is conducted to verify the performance of the proposed method, and the results demonstrate that the proposed approach performs significantly better in crack and bughole detection on concrete surfaces.

INDEX TERMS Crack detection, bughole detection, deep learning, concrete surface, semantic segmentation.

I. INTRODUCTION

Controlling the surface quality of concrete is one of the main challenges faced by the concrete industry today. Defects on the surface of a concrete structure can visually reflect its durability, safety and maintainability. The most common factors affecting the quality of concrete surfaces are cracks and bugholes [1], [2], which are surface defects that are usually scattered randomly on a concrete surface [3]. Research has demonstrated that cracks can affect the safety and sustainability of concrete buildings, while bugholes can reduce the adhesion of fiber reinforced plastic (FRP) material on concrete surfaces [4]. Moreover, if salt accumulates in bugholes, it can lead to the premature degradation of reinforced concrete (RC) structures. Therefore, crack

and bughole detection is of great significance in building maintenance.

Traditionally, a common method for assessing defects on concrete surface is via human visual inspection [5]. However, owing to the inherent subjectivity of human perception, different people usually produce different judgment results for the same concrete surface conditions [6]. In addition, human visual inspection is usually labor-intensive, time-consuming and unable to consistently produce quantitative objective results. Therefore, automatic defect inspection is highly desirable for efficient and objective defect assessment.

Considering the shortcomings of traditional inspection methods based on human visual identification, computer vision-based methods have been widely studied. A large number of damage detection methods based on image processing techniques (IPT) have been proposed. Almost all surface defects can be identifiable, which is a significant

The associate editor coordinating the review of this manuscript and approving it for publication was Luca Cassano.

advantage of IPT. IPT combined with sliding window technology was used by Yeum and Dyke to detect cracks [7]. The potential of IPT is well demonstrated in this study. Liu and Yang [8] used an image analysis approach to detect surface bugholes and the CIB bughole scale, while an OTSU image threshold segmentation technique was adopted to extract the features of the bugholes on the concrete surface. Cheng *et al.* [9] used intensity threshold-based algorithms to identify pavement cracks via setting different threshold values. However, the detection performance is seriously diminished when there are a certain number of noisy pixels with intensities lower than the crack pixels. Since the edges and cracks have a multitude of similarities in morphology, many studies [10], [11] adopt filter-based methods which have been developed for edge detection to detect cracks on pavement images. Though the IPT-based defect detection approach is fast and effective, its robustness is far from adequate when noise, mainly from distortion and lighting, seriously affects the results [12]. Implementing denoising technology is an effective way to overcome these problems. As a well-known technique, total variation denoising can reduce the noise of image data, so as to enhance the edge detectability of the image [13]. However, due to the great changes of image data captured in the real engineering, the applications of prior knowledge in IPT are limited. The drawbacks of these traditional crack and bughole detection methods are obvious: each method is designed for a specific setting or database. If the setting or database is changed, the crack or bughole detectors often fail. In addition, it is arduous to extract semantic information (such as the location and width of cracks) from images. Image processing algorithms are usually designed to help inspectors detect defects and still rely on manual judgment to obtain the final results [14].

With the development of image acquisition equipment and computing capabilities, a host of machine learning algorithms (such as deep learning) has been used for object recognition and have achieved acceptable results [15]–[17]. Deep learning technique is a data-driven approach without requiring rules designed manually. In the process of building the model, we only need to select an appropriate network structure, a function to evaluate the model output and a reasonable optimization algorithm. The Convolutional Neural Network (CNN) has attracted wide attention as an effective recognition method [18] and has been highlighted in object detection and image classification [19]. Liu *et al.* [20] proposed a deep learning algorithm to detect the rebar hyperbolas automatically, and the rebar depth is estimated with a high accuracy by migration of rebar hyperbola. A method based on deep learning was developed to detect concrete bugholes [21], [22], concrete cracks [23]–[27], road cracks [6], [28]–[30] and other defects [31]–[33]. When only one defect type is detected, these methods maybe achieve outstanding performance in realistic situations. Previous research has focused primarily on using CNNs to locate and classify defects, such as using a bounding box to determine the localization of each crack and classify individual cracks in images. The object

detection technology uses a rectangular frame to locate the object, while the crack and bughole distribution and shape of the concrete surface are irregular. Therefore, the recognition accuracy of these methods is limited. Subsequently, a Faster Region-based Convolutional Neural Network (Faster R-CNN) method was applied by Cha *et al.* [34] for multiple defects. This method still detects multiple defects at the grid-cell. In other words, it can effectively detect cracks in the image, but can hardly provide pixel-level concrete crack detections. To improve the detection accuracy, the task of detecting concrete surface defects is regarded as a semantic segmentation task. The goal of semantic segmentation is to classify each pixel in the image. As a deep learning network structure proposed for image semantic segmentation tasks, FCN is an ideal way to do the image semantic segmentation. Li *et al.* [35] proposed a fully convolutional network (FCN) method for multiple damage types (crack, spalling, hole and efflorescence). This method still uses several groups of learning rates for optimal training, and it needs a lot of manpower work before applying the trained model, so it is complicated and time-consuming. Therefore, it is meaningful to find a simple and fast but high precision method.

In this paper, we focus on simultaneous crack and bughole detection at the pixel level. To achieve higher detection performance for crack and bughole defects, a method based on deep learning is proposed for crack and bughole detection on concrete surfaces. DeepLabv3+ [36] is a state-of-the-art framework for semantic segmentation which extends DeepLabv3 with an encoder-decoder structure. This model has a good segmentation effect in the visual field, and it is improved on this basis. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries. In addition, compared with the current training process for defect recognition, the optimization algorithm used in the model of this method is able to auto-adjust the learning rate, avoiding the complexity of using multiple groups of learning rates to achieve optimization. However, there are few applications of DeepLabv3+ in defect detection in the field of civil engineering.

In this paper, the method whose network is improved based on the visual characteristics of cracks and bugholes, is used for object detection and semantic segmentation of cracks and bugholes in concrete surface images. The previous study mainly focused on detecting only one defect. The first contribution of this study is to reduce the information loss during upsampling. In the decoder, the 3×3 convolution of the feature fusion part is improved to a 3-layer depth separable convolution. To avoid the loss of accuracy, a smaller expansion rate combination is used and the global polling layer in the raw image is removed. The original expansion rate combination is changed from 1, 6, 12, 18 to 1, 2, 4, 8 to improve the segmentation effect of the model on the image. This is the second major contribution of the study. Moreover, based on the idea of the channel Attention mechanism of

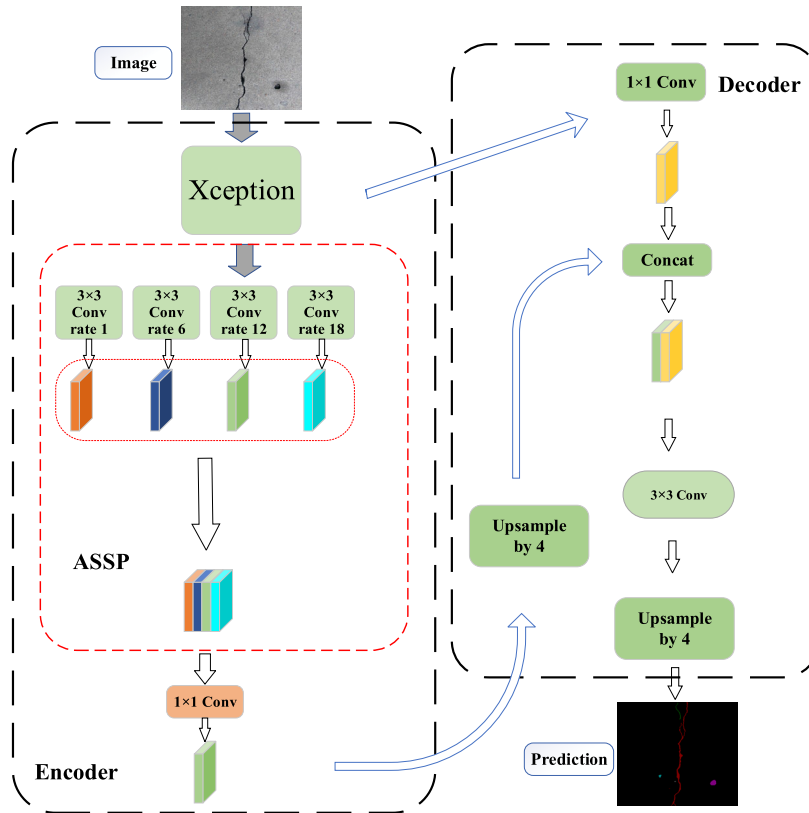


FIGURE 1. Overall schematic flowchart of the original DeepLabv3+.

SENet, a weight value is added to each channel of the Atrous Spatial Pyramid Pooling (ASSP) module, and the feature maps that contribute significantly to the target prediction are learned and screened. This can reduce the burden of processing high-dimensional data and make the network pour more attention into the crucial part of the input information. Besides, it can better judge the mapping relationship between input and output and further improve the model’s prediction accuracy and generalization ability.

II. METHODOLOGY

To detect and locate cracks and bugholes on concrete surfaces, the modified semantic image segmentation framework DeepLabv3+ is applied in this study. The features of cracks and bugholes are extracted simultaneously from concrete images in a database. The overall schematic flowchart of the original DeepLabv3+ is shown in Fig. 1. The input and the output are the original image and the recognition result, respectively. The recognition result shows the class label and location of the cracks and bugholes. The Xception-65 module form, which has achieved accurate calculations and fast results in the latest image classification and object detection, is used as the backbone of the network architecture training in this study [37].

The detailed implementations are described in this section, including the network architecture construction and the improved DeepLabv3+.

A. NETWORK ARCHITECTURE CONSTRUCTION

The Xception network structure is mainly based on the structure of depth wise separable convolution to improve the multi-scale extraction feature Inception v3 convolution method.

1) DEPTHWISE SEPARABLE CONVOLUTION

The main implementation process of depth wise separable convolution is as follows: M represents the number of channels of the input feature and N represents the number of channels of the output feature (also the number of convolution kernels of this layer). Therefore, if the size of the convolution kernel is assumed to be $D_k * D_k * M * N$, and the output is $D_F * D_F * N$, then the amount of computation of the standard convolution is $D_k * D_k * M * N * D_F * D_F$. If $M * N$ is removed from the above formula, it becomes a two-dimensional convolution kernel deconvolving a two-dimensional input feature map. If the size of the output feature map is $D_F * D_F$, since each point of the output feature map is generated by a convolution operation, and each convolution will have $D_k * D_k$ computation, a two-dimensional convolution kernel deconvolving a two-dimensional input feature map has $D_F * D_F * D_k * D_k$ computation. If there are M input feature maps and N convolution kernels, the amount of computation will be $D_F * D_F * D_k * D_k * M * N$.

The algorithm in this paper uses (b) + (c) instead of (a). Assuming that there are N convolution kernels, each

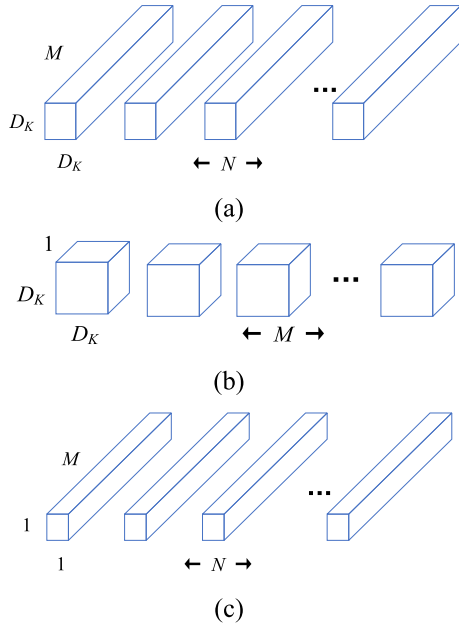


FIGURE 2. Filter principle for standard convolution and depthwise separable convolution: (a) Standard convolutional filters; (b) Depthwise convolutional filters; (c) 1×1 Convolutional filters, referred to as pointwise convolution in the context of depthwise separable convolution.

convolution kernel has a dimension of $D_k * D_k * M$, the number of input feature map channels is M and the output feature map is $D_F * D_F * N$. Thus, (b) means convolution with M convolution kernels of dimension $D_F * D_F * 1$ and the input M feature maps, and M results will be obtained. These M results do not accumulate with each other. (Note that the number of convolution kernels is M instead of N , so there is no N in (b), only M .) Therefore, the amount of computation is $D_F * D_F * D_k * D_k * M$. The result of (b) should be $D_F * D_F * M$. Fig. 2 (b) represents the dimensions of the convolution kernel. Fig. 2 (c) represents the result of convolution (b) with N convolution kernels of dimension $1 * 1 * M$. Namely, the input is $D_F * D_F * M$, the feature map of $D_F * D_F * N$ is finally obtained and the amount of computation is $M * N * D_F * D_F * 1 * 1$. The amount of computation using this algorithm becomes $D_k * D_k * M * N * D_F * D_F + M * N * D_F * D_F$. Compared to (a), the amount of calculation is reduced as follows:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \quad (1)$$

For instance, the size of the convolution kernel is $3 \times 3 \times 3$ and the time of the convolution operation can be reduced to about 1/9 of the original.

2) ATROUS CONVOLUTION

When the image is input into the network, the network will perform convolution and pooling operations on the image. A pooling operation simultaneously reduces the size of the image and increases the receptive field. However, since image segmentation prediction is the pixel-wise output,

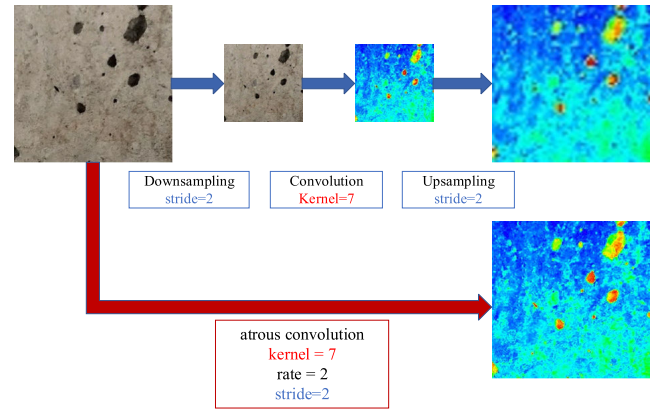


FIGURE 3. Difference between atrous convolution and standard convolution.

the deconvolution operation is usually used for prediction by upsampling the smaller image after pooling to the original image size. The specific process is shown in Fig. 3.

In Fig. 3, the top row uses standard convolution for feature extraction on the low-resolution input feature map, which has a slice of losses. Consequently, the high-resolution input feature map on the bottom row uses the atrous convolution of rate = 2 for dense feature extraction.

3) XCEPTION-65

The Xception-65 network structure in this paper adopts the depthwise separable convolution method to greatly save time and improve the effect of the model without increasing the network complexity. The specific network structure is shown in Fig. 4. The comparison between the Xception-65 network structure and other networks in the model is shown in Table 1.

TABLE 1. Comparison between the Xception-65 network structure and other networks in the model.

	Top-1 accuracy	Top-5 accuracy
VGG-16	0.715	0.901
ResNet-152	0.770	0.933
Inception V3	0.782	0.941
Xception	0.790	0.945

4) ATROUS SPATIAL PYRAMID POLLING (ASSP)

ASSP is affected by the idea of spatial pyramid pooling in the object detection algorithm R-CNN, which shows that by fusing the features extracted from convolutions of multiple different sizes, regions of any size can be effectively and accurately classified. ASSP adopts this idea and atrous convolution is applied to this structure, which is composed of multiple parallel atrous convolution layers with different rates. The features extracted from the atrous convolution of each rate value are first processed in independent branches and finally fused together to obtain the final result (Fig. 5).

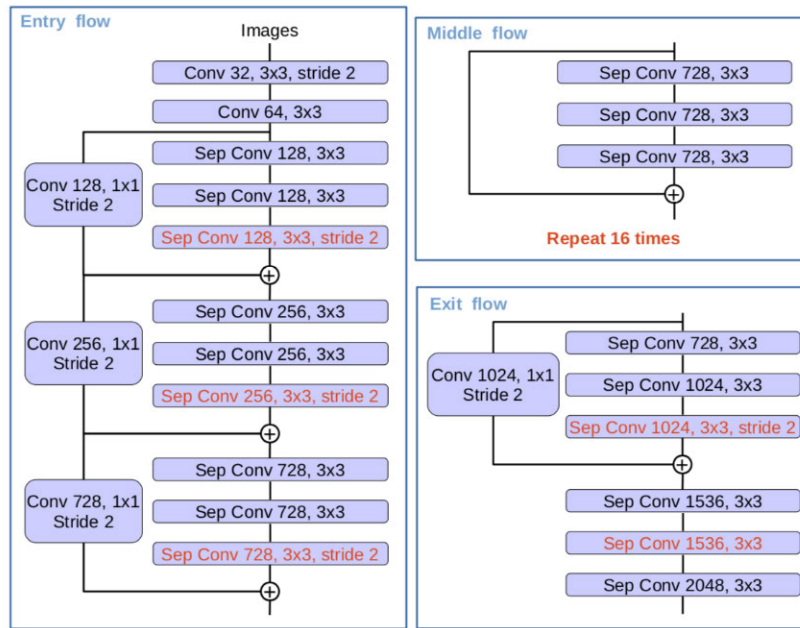


FIGURE 4. Xception-65 network structure diagram [36].

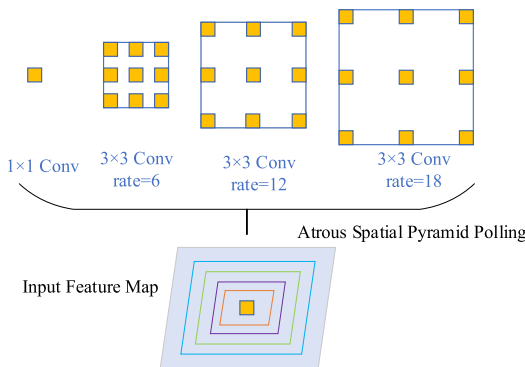


FIGURE 5. Structural diagram of ASSP.

B. IMPROVED DeepLabv3+

In our research, the proposed DeepLabv3+ is specially designed for crack and bughole detection to improve the accuracy and generalization of the model.

First of all, in the decoder, the 3×3 convolution of the feature fusion part is improved to a 3-layer depth separable convolution to maintain the spatial position information and depth information, gradually obtain the fine segmentation results and reduce information loss during upsampling (Fig. 6).

The image is small (256×256 pixels), and the area occupied by cracks and bugholes in the image is close. The scale conversion is not obvious. The detected defects occupy a small area in the image, and if a higher expansion ratio is adopted, the extracted features will not be obvious. Using the expansion ratio combination of the original ASSP module can easily cause a loss of accuracy. Accordingly, a smaller expansion rate combination is used and the global polling

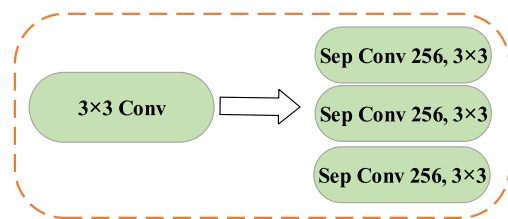


FIGURE 6. Replacing ordinary convolution with separable convolution.

layer in the original image is removed at the same time. The original expansion rate combination is changed from 1, 6, 12, 18 to 1, 2, 4, 8 to improve the segmentation effect of the model on the image. The expansion rate combination 1, 2, 4, 8 is the smallest combination. Finally, based on the idea of the channel Attention mechanism of SENet, a weight value is added to each channel of the ASSP module and the feature maps which contribute significantly to the target prediction are learned and screened (Fig. 7).

The original network does not carry out channel weighting on the feature maps. By default, all channel information is treated equally, and its contributions to the final target prediction are considered to be the same. In fact, with the accumulation of convolutional layers and the enrichment of semantic information, each channel carries different feature information, and the degree of association between the information and the target is different. If the channels of the feature map can be weighted, and the features that contribute significantly to the target prediction are learned and screened, the burden of processing high-dimensional data can be reduced. In addition, this can make the network pour more attention into the crucial part of the input information, better judge the mapping relationship between input and output and further improve the

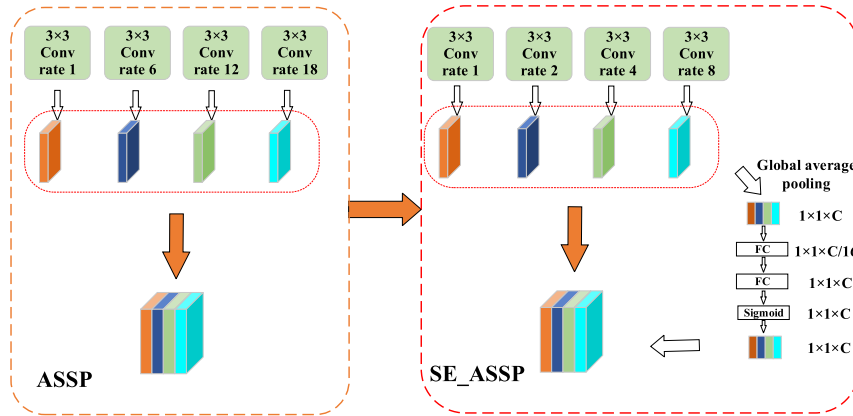


FIGURE 7. Improved SE_ASSP module.

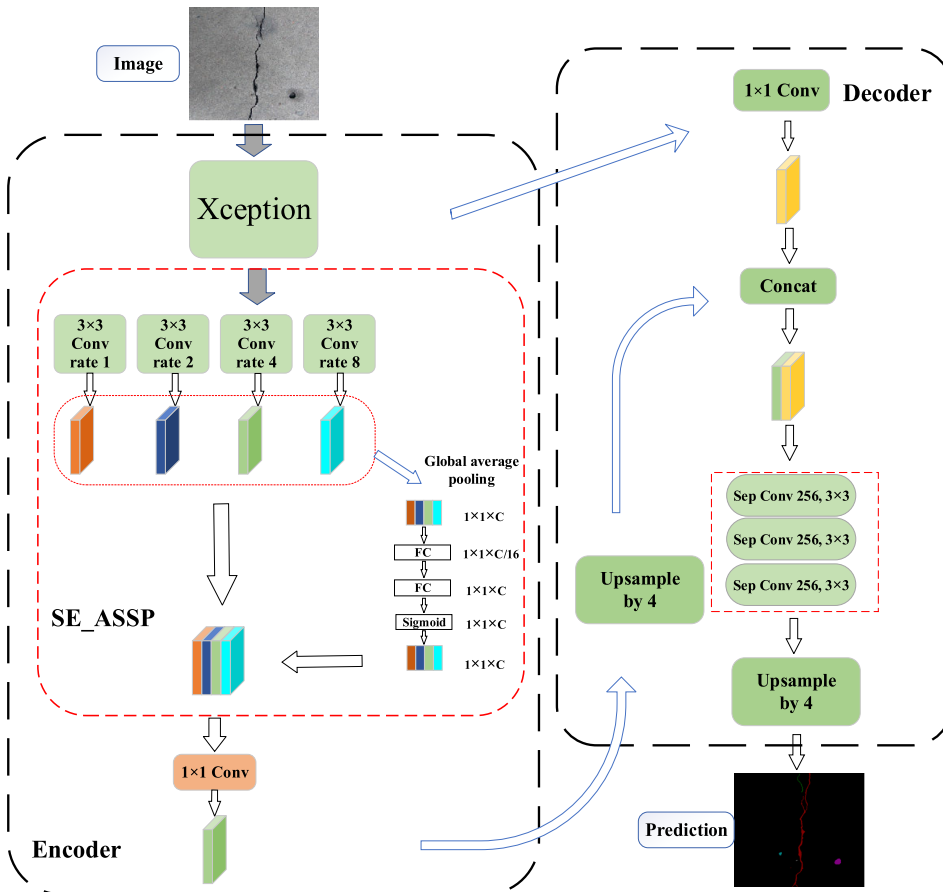


FIGURE 8. Overall schematic flowchart of the proposed method.

model’s prediction accuracy and generalization ability. The overall schematic flowchart of the proposed method is shown in Fig. 8.

III. EXPERIMENTS

A. IMPLEMENTATION DETAILS

The operating system used in this study is Linux Ubuntu 16.04.1, and it is equipped with Intel Core (TM) i7-8750H CPU@2.20GHz, GeForce RTX 2070 graphics and 8GB memory. The system code for simultaneous detection of

cracks and bugholes was written in Python and reproduced using the Tensorflow framework. The Tensorflow framework is an open source software library based on data flow programming through which many machine learning algorithms can be implemented programmatically.

B. IMAGE DATABASE CREATION

In order to facilitate image acquisition, iPhone X with a 12-megapixel wide-angle and telephoto dual-lens camera is used to acquire images. Images were taken under different

lighting conditions to ensure diversity and complexity. To collect images of small defects (cracks and bugholes), all images were taken at a distance of 0.1 meters between the concrete surface and the smartphone. A total of 2000 raw images with dimensions of 3024×3024 pixels were taken from the surfaces of concrete buildings. Considering the training performance of the neural network and the display memory of the computer, the training efficiency is too low when the image is too large. Therefore, we use small images when training features, and we can use original images during testing if necessary. Each raw image can be automatically cropped to generate 139 images with dimensions of 256×256 pixels. However, some cropped images do not contain cracks or bugholes. Consequently, the images containing cracks or bugholes were meticulously selected from the set of cropped images. As a result, a total of 16,662 images that met the requirements were selected to create the database.

There are two kinds of defects (cracks and bugholes) in the images, and the pixels of the background, cracks and bugholes are labeled 0, 1 and 2, respectively. The most fundamental and crucial step is to label the images when creating the database. The image label tool “labelme” is used to label the masks of the objects (cracks and bugholes). Each image in the database has a corresponding annotation file which provides an object contour label and an image class label for each image. In the light of the function, the database can be divided into three parts: the training set, test set and validation set. The training set is mainly used for training the model, the validation set avoids over fitting the model and the test set is primarily used for testing the performance of the model. The trained model gives the detection results of each image in the test set. The accuracy of the model can be calculated by comparing the results given by the model with the manual labeling results.

Due to the large difference between the number of bughole databases and crack databases, the preset data transformation rules are adopted for the bughole data. Geometric operations and color transformation are applied to amplify the data based on the existing data. There are several common types of image geometric transformations: flipping, rotating, cropping, deformation and so on. In this study, we have not used deformation operation, because it will damage the texture characteristics of bugholes and cracks. We mainly use flipping and rotating operation to redistribute the pixels of cracks and bugholes in the image. On the basis of the original color transformation class, some noise is added randomly. Furthermore, owing to the different sizes of the bughole and crack pixels, and the fact that an extremely large proportion of the background is occupied, when training, the training weight of the background: crack: bughole is set to 1:10:15, and the correction of the category imbalance is considered. The definition of a crack or a bughole in this paper is that it should be identifiable in images via the naked human eye.

To evaluate the generalization ability of the modified DeepLabv3+, the database is divided into five parts (80% for training and validating the model and the last 20% for

TABLE 2. Percentage of the training, validation and testing sets.

Defect type	Training	Validation	Testing
Only crack	9186	2000	2796
Only bughole	3086	440	870
Crack + bughole	336	50	96

TABLE 3. Parameters in the training process.

Parameter	Training
Base_lr	0.0001
Lr_decay	0.1
Batch_size	16
Weight_decay	0.0001

testing) according to the fivefold cross-validation principle. Table 2 shows the detailed percentages of the training, validation and testing sets.

The Tensorflow open source framework, which provides a unified input data format (TFRecord format), has two main advantages. One is that all of a sample’s information can be stored together, even if it includes different data types. The “protocol buffer” binary data encoding scheme is applied to the memory. The other advantage is that the multi-threaded operation of the file queue can be used to make the data reading speed and batch processing faster and more convenient. Accordingly, the database is transformed into the training format required by the open source framework.

C. MODEL INITIALIZATION

When training DeepLabv3+, a strategy of model-based transfer learning [38], [39] is applied to optimize and accelerate the learning efficiency of the model. In this paper, transfer learning refers to the method of the feature extraction weights in other fields based on the main framework Xception of DeepLabv3+ as the pretraining weight for crack and bughole detection. According to this strategy, the biases and weights of the Xception part of the DeepLabv3+ model are initialized by the pretrained backbones. In this paper, we considered the momentum algorithm and the Adam algorithm, and found that the loss function value of the momentum algorithm dropped to the lowest during training. Therefore, the momentum algorithm has good performance. We also considered the parameters in Table 3 during training. The detection system was trained 400,000 times with the momentum algorithm, and the training batch size was 16. Table 3 shows the detailed parameters of the training process. The curve of the learning rate in the training process and the loss curve of the training and validation processes are shown in Fig. 9 and Fig. 10, respectively.

As shown in Fig. 9, the initial learning rate is 10^{-4} and the optimal learning rate is 1.15×10^{-6} for 400,000 iterations.

TABLE 4. Input and output for crack and bughole detection on concrete surface.

Category	Input	Output
Only crack		
Only bughole		
Crack + bughole		

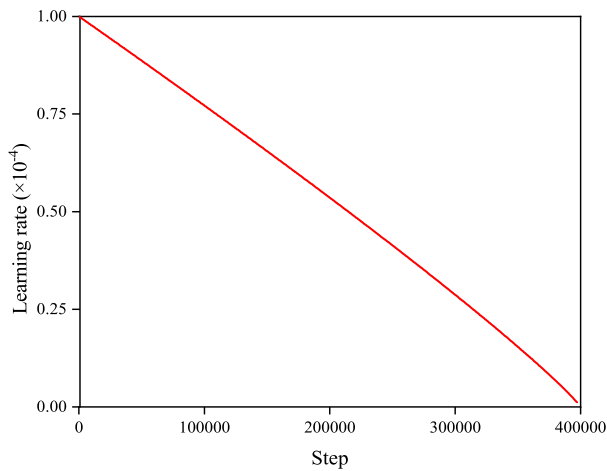


FIGURE 9. Learning rate.

Fig. 10 depicts the record training and validation losses of the modified DeepLabv3+ under the optimal learning rate (1.15×10^{-6}). It can be seen that the training loss decreases rapidly at the beginning and then converges around 0.15.

D. ACCURACY EVALUATION METRICS

To evaluate the accuracy of the technique for semantic segmentation, many evaluation standards have been proposed and used. To measure the effect of per-pixel labeling approaches on performance, the most common current metrics for semantic segmentation are Average Precision (AP),

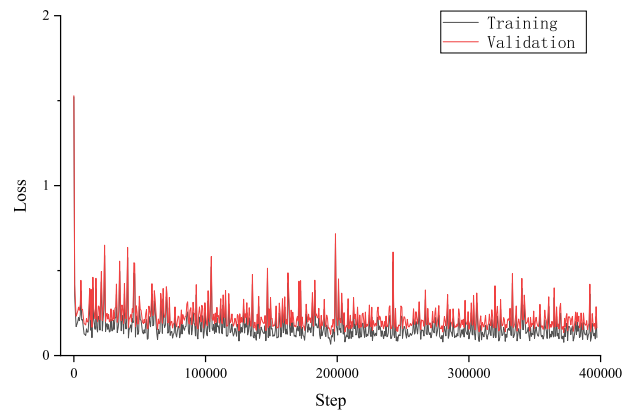


FIGURE 10. Loss curve of training and validation processes.


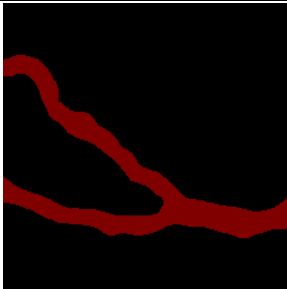
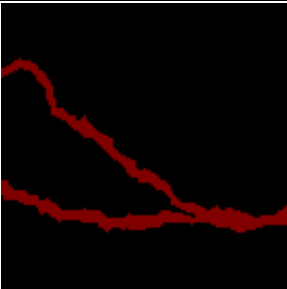
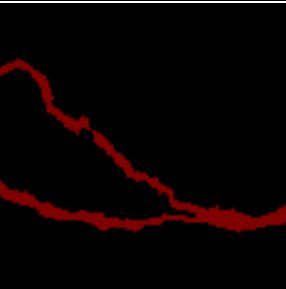



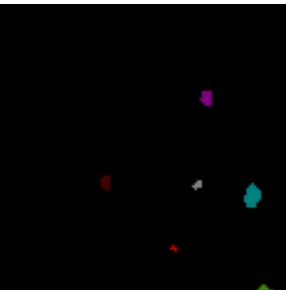

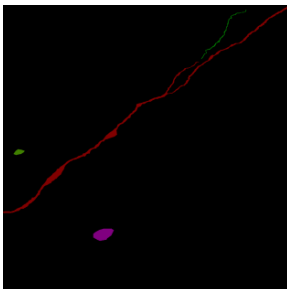
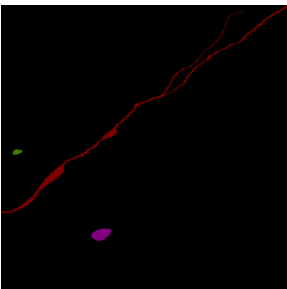
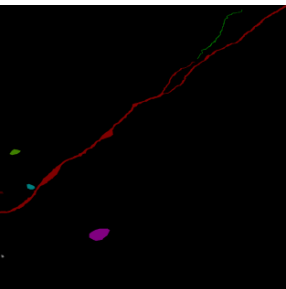
Mean Average Precision (MAP) and Mean Intersection over Union (MIoU). The evaluation metrics can be formulated as follows:

$$AP = \frac{1}{2} \sum_{r=0}^1 \max_{r':r'>r} Precision(r') \tag{2}$$

$$mAP = \frac{1}{m} \sum_{n=1}^m AP_n \tag{3}$$

$$MIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{P_{ii}}{\sum_{j=0}^k P_{ij} + \sum_{j=0}^k P_{ji} - P_{ii}} \tag{4}$$

TABLE 5. Comparison of prediction results for the proposed DeepLabv3+, FCN and original DeepLabv3+.

Original image	Prediction by original DeepLabv3+	Prediction by FCN	Prediction by proposed DeepLabv3+
			
			
			

where i represents the true value and j represents the predicted value. p_{ij} indicates that i is predicted as j . As shown in Fig. 11, red represents the true value and yellow represents the predicted value. The orange part is the intersection of the two circles, which is MIoU.

E. LOSS FUNCTION OF TRAINING

The output of the network is the pixel-wise softmax, that is:

$$p_k(x) = \exp(a_k(x)) / (\sum_{k=1}^K \exp(a_k(x))) \quad (5)$$

where x is the pixel position on the two-dimensional plane, $a_k(x)$ represents the value of the channel k corresponding to x in the pixel in the last output layer of the network. $p_k(x)$ represents the probability that the pixel x belongs to the class k .

The loss function uses negative-class cross entropy to solve the class imbalance of cracks, bugholes and background. That is:

$$E = \sum_x w(x) \log(p_{l(x)}(x)) \quad (6)$$

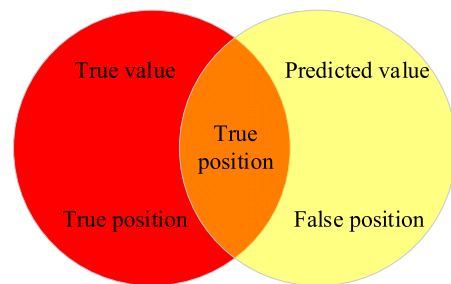


FIGURE 11. MIoU metrics.

where $p_i(x)$ represents the output probability of x on the channel where the real label is located and $w(x)$ is the weight of each pixel. It is unreasonable to treat the loss of each pixel equally when a certain kind of image is dominant in the cross entropy loss of semantic segmentation. Different pixel losses are weighted to give higher weight to the pixels at the boundary of the segmentation target. The idea of weighted loss enables the DeepLabv3+ model to provide a segmentation map with a clear boundary when detecting concrete cracks and bugholes, which makes it easier to distinguish between each bughole and crack.

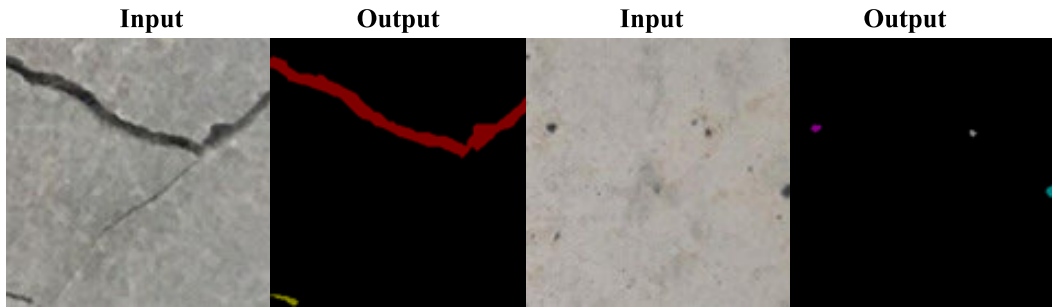


FIGURE 12. Quintessential instances of incorrectly predicted results.

TABLE 6. Comparison results of the proposed DeepLabv3+, FCN and original DeepLabv3+.

Method	AP(%)	MAP(%)	MIoU(%)
Proposed DeepLabv3+	Crack: 97.63	95.58	81.87
	Bughole: 93.53		
FCN	Crack: 89.33	84.90	67.87
	Bughole: 80.46		
Original DeepLabv3+	Crack: 95.43	93.58	78.85
	Bughole: 91.73		

F. RESULTS

To understand more intuitively that the trained DeepLabv3+ is to identify cracks and bugholes on a concrete surface, the input and output images are shown in Table 4.

As shown in Table 4, different kinds of defects (only crack, only bughole or crack + bughole) can be accurately identified. No matter how complex the cracks and bugholes were and how many cracks and bugholes the image contained, the modified DeepLabv3+ was effective, simple, and able to generalize. Satisfactory accuracy was achieved. However, it is worth noting that there were still cases of failure, as shown in Fig. 12. We observed that the low resolution of the images leads to minor errors in defect recognition in the cases of certain images with narrow cracks and tiny bugholes. In addition, these minor errors may be caused by the small training database. Despite the minor errors, the results illustrate the robust performance of the proposed method in detecting cracks and bugholes in concrete surface images. Accordingly, a larger database with higher resolution images of cracks and bugholes on concrete surfaces under various conditions will be established to improve the capacity and generalization of the method in future research.

IV. COMPARATIVE STUDY

To compare the performance of the proposed method based on DeepLabv3+ and a state-of-the-art semantic segmentation approach, the established database, including cracks and bugholes on concrete surfaces, is used to train the original DeepLabv3+ [36] and FCN [40] model. The selected comparison algorithm should be well-known in the field of semantic segmentation, and the segmentation effect should

be satisfactory. Moreover, the comparison algorithm should have open source code and parameters set by the author. When the semantic segmentation algorithm is applied in the field of civil engineering, the FCN algorithm is widely used and its effect is also satisfactory, so it is often used as a comparison algorithm. In addition, our algorithm is improved on the basis of DeepLabv3+, so it is necessary to compare with the original DeepLabv3+ algorithm to prove our contribution. The results of the comparative study are shown in Table 5 and Table 6.

In TABLE 5, there are only cracks in the first original image. The results predicted by the improved DeepLabv3+ are closer to the crack shape in the original image, and the performance is better. There are six visible surface bugholes in the second original image, and all six bugholes can be identified in the prediction results obtained by improved DeepLabv3+, but the other two methods can only identify four obvious bugholes. One crack and three bugholes are visible to the naked eye in the third original image. The prediction results obtained by improved DeepLabv3+ can be correctly identified, but one bughole is missed by the other two methods. From the comparison of these three methods, we concluded that the proposed DeepLabv3+ has better performance in detecting concrete cracks and bugholes because all of its accuracy evaluation metrics (AP, MAP, and MIoU) are superior to those of the other two methods. Furthermore, this method can successfully detect tiny bugholes, and its prediction areas are the closest to the true areas, while the other two methods cannot detect tiny bugholes or thin cracks. Compared with the other algorithms, which require a fixed learning rate to train repeatedly for the best results,

a crucial advantage of the proposed method is the fact that the optimization algorithm used in the model can auto-adjust the learning rate very simply.

V. CONCLUSION

A defect detection method based on DeepLabv3+ is proposed to detect two concrete surface defects: cracks and bugholes. A smartphone was used to collect 2000 original images (with dimensions of 3024×3024 pixels) from the surfaces of concrete buildings. The images were cropped to 256×256 pixels to reduce the computation of the training process. Owing to the large difference between the number of databases of bugholes and cracks, the preset data transformation rules were adopted for the bughole data. After data augmentation, the number of images used for training, validation and testing were 12,608, 2,490 and 3,762, respectively. The image annotation tool “labelme” was applied to annotate the labels and the masks of the cracks and bugholes. The DeepLabv3+ architecture was modified to better adapt to the simultaneous crack and bughole detection, which is described in detail in Section II.B. A model-based transfer learning strategy is applied to optimize and accelerate the learning efficiency of the model. The biases and weights of the Xception part of the model are initialized by the pretrained backbones. The proposed method is able to auto-adjust the learning rate. The results show an AP of 97.63% (crack) and 93.53% (bughole), a MAP of 95.58% and an MIoU of 81.87%. The testing images which were not used to train and validate were used to test the robustness of the trained model. The performance of the proposed method was also compared with the original DeepLabv3+ and FCN-based methods. The comparative study showed that the proposed method can provide excellent defect detection results. It can detect tiny cracks and bugholes at the same time, and the results are more detailed.

The proposed method effectively detects defects (cracks and bugholes on concrete surfaces) and exhibits low level of noise. A common drawback of almost all vision-based methods (including approaches based on CNNs, FCNs and IPTs) is that they are unable to detect the depth of the defects as a consequence of the nature of flattened photographic images.

In the future, more kinds of defect images across more complex backgrounds will be collected to expand the database, so as to improve the accuracy and robustness of the proposed method. The application research can be popularized and applied in actual engineering in the field of civil engineering. The improved algorithm in the follow-up research can be used to develop the detection equipment and applied it to practical engineering.

REFERENCES

- [1] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, “DeepCrack: A deep hierarchical feature learning architecture for crack segmentation,” *Neurocomputing*, vol. 338, pp. 139–153, Apr. 2019, doi: [10.1016/j.neucom.2019.01.036](https://doi.org/10.1016/j.neucom.2019.01.036).
- [2] V. Hedda, *State of the Art—Quality of Concrete Surfaces*, 1st ed. Trondheim, Norway, 2007.
- [3] T. Ozkul and I. Kucuk, “Design and optimization of an instrument for measuring bughole rating of concrete surfaces,” *J. Franklin Inst.*, vol. 348, no. 7, pp. 1377–1392, Sep. 2011, doi: [10.1016/j.jfranklin.2010.04.004](https://doi.org/10.1016/j.jfranklin.2010.04.004).
- [4] A. S. Kalayci, B. Yalim, and A. Mirmiran, “Effect of untreated surface disbands on performance of FRP-retrofitted concrete beams,” *J. Compos. Construct.*, vol. 13, no. 6, pp. 476–485, Nov-Dec. 2009, doi: [10.1061/\(asce\)cc.1943-5614.0000032](https://doi.org/10.1061/(asce)cc.1943-5614.0000032).
- [5] D. Ai, G. Jiang, L. S. Kei, and C. Li, “Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods,” *IEEE Access*, vol. 6, pp. 24452–24463, 2018, doi: [10.1109/access.2018.2829347](https://doi.org/10.1109/access.2018.2829347).
- [6] C. Laofor and V. Peansupap, “Defect detection and quantification system to support subjective visual quality inspection via a digital image processing: A tiling work case study,” *Autom. Construction*, vol. 24, pp. 160–174, Jul. 2012, doi: [10.1016/j.autcon.2012.02.012](https://doi.org/10.1016/j.autcon.2012.02.012).
- [7] C. M. Yeum and S. J. Dyke, “Vision-based automated crack detection for bridge inspection,” *Comput.-Aided Civil Infrastruct. Eng.*, vol. 30, no. 10, pp. 759–770, Oct. 2015, doi: [10.1111/micc.12141](https://doi.org/10.1111/micc.12141).
- [8] B. Liu and T. Yang, “Image analysis for detection of bugholes on concrete surface,” *Construct. Building Mater.*, vol. 137, pp. 432–440, Apr. 2017, doi: [10.1016/j.conbuildmat.2017.01.098](https://doi.org/10.1016/j.conbuildmat.2017.01.098).
- [9] H. D. Cheng, X. J. Shi, and C. Glazier, “Real-time image thresholding based on sample space reduction and interpolation approach,” *J. Comput. Civil Eng.*, vol. 17, no. 4, pp. 264–272, Oct. 2003, doi: [10.1061/\(asce\)0887-3801\(2003\)17:4\(264\)](https://doi.org/10.1061/(asce)0887-3801(2003)17:4(264)).
- [10] E. Zalama, J. Gómez-García-Bermejo, R. Medina, and J. Llamas, “Road crack detection using visual features extracted by Gabor filters,” *Comput.-Aided Civil Infrastruct. Eng.*, vol. 29, no. 5, pp. 342–358, May 2014, doi: [10.1111/micc.12042](https://doi.org/10.1111/micc.12042).
- [11] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, “Pavement crack detection using the Gabor filter,” in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, The Hague, The Netherlands, Oct. 2013, pp. 2039–2044, doi: [10.1109/ITSC.2013.6728529](https://doi.org/10.1109/ITSC.2013.6728529).
- [12] M. Kozlarski and B. Cyganek, “Image recognition with deep neural networks in presence of noise—Dealing with and taking advantage of distortions,” *Integr. Comput.-Aided Eng.*, vol. 24, no. 4, pp. 337–349, Sep. 2017, doi: [10.3233/ica-170551](https://doi.org/10.3233/ica-170551).
- [13] A. Beck and M. Teboulle, “Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems,” *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009, doi: [10.1109/TIP.2009.2028250](https://doi.org/10.1109/TIP.2009.2028250).
- [14] J.-K. Oh, G. Jang, S. Oh, J. H. Lee, B.-J. Yi, Y. S. Moon, J. S. Lee, and Y. Choi, “Bridge inspection robot system with machine vision,” *Autom. Construct.*, vol. 18, no. 7, pp. 929–941, Nov. 2009, doi: [10.1016/j.autcon.2009.04.003](https://doi.org/10.1016/j.autcon.2009.04.003).
- [15] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, “Multi-column deep neural network for traffic sign classification,” *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012, doi: [10.1016/j.neunet.2012.02.023](https://doi.org/10.1016/j.neunet.2012.02.023).
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: [10.1109/TPAMI.2015.2389824](https://doi.org/10.1109/TPAMI.2015.2389824).
- [18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: [10.1038/nature14539](https://doi.org/10.1038/nature14539).
- [19] 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, doi: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031).
- [20] H. Liu, C. Lin, J. Cui, L. Fan, X. Xie, and B. F. Spencer, “Detection and localization of rebar in concrete by deep learning using ground penetrating radar,” *Autom. Construct.*, vol. 118, Oct. 2020, Art. no. 103279, doi: [10.1016/j.autcon.2020.103279](https://doi.org/10.1016/j.autcon.2020.103279).
- [21] F. Wei, G. Yao, Y. Yang, and Y. Sun, “Instance-level recognition and quantification for concrete surface bughole based on deep learning,” *Autom. Construct.*, vol. 107, Nov. 2019, Art. no. 102920, doi: [10.1016/j.autcon.2019.102920](https://doi.org/10.1016/j.autcon.2019.102920).
- [22] G. Yao, F. Wei, Y. Yang, and Y. Sun, “Deep-learning-based bughole detection for concrete surface image,” *Adv. Civil Eng.*, vol. 2019, pp. 1–12, Jun. 2019, doi: [10.1155/2019/8582963](https://doi.org/10.1155/2019/8582963).

- [23] R. S. Adhikari, O. Moselhi, and A. Bagchi, "Image-based retrieval of concrete crack properties for bridge inspection," *Autom. Construct.*, vol. 39, pp. 180–194, Apr. 2014, doi: [10.1016/j.autcon.2013.06.011](https://doi.org/10.1016/j.autcon.2013.06.011).
- [24] Y.-J. Cha, W. Choi, and O. Büyükköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, May 2017, doi: [10.1111/mice.12263](https://doi.org/10.1111/mice.12263).
- [25] F.-C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018, doi: [10.1109/tie.2017.2764844](https://doi.org/10.1109/tie.2017.2764844).
- [26] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Autom. Construct.*, vol. 99, pp. 52–58, Mar. 2019, doi: [10.1016/j.autcon.2018.11.028](https://doi.org/10.1016/j.autcon.2018.11.028).
- [27] 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, doi: [10.1016/j.autcon.2020.103199](https://doi.org/10.1016/j.autcon.2020.103199).
- [28] A. Zhang, K. C. P. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, and C. Chen, "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 10, pp. 805–819, Oct. 2017, doi: [10.1111/mice.12297](https://doi.org/10.1111/mice.12297).
- [29] Z. Qu, J. Mei, L. Liu, and D.-Y. Zhou, "Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model," *IEEE Access*, vol. 8, pp. 54564–54573, 2020, doi: [10.1109/access.2020.2981561](https://doi.org/10.1109/access.2020.2981561).
- [30] W. Song, G. Jia, D. Jia, and H. Zhu, "Automatic pavement crack detection and classification using multiscale feature attention network," *IEEE Access*, vol. 7, pp. 171001–171012, 2019, doi: [10.1109/access.2019.2956191](https://doi.org/10.1109/access.2019.2956191).
- [31] Y.-Z. Lin, Z.-H. Nie, and H.-W. Ma, "Structural damage detection with automatic feature-extraction through deep learning," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 12, pp. 1025–1046, Dec. 2017, doi: [10.1111/mice.12313](https://doi.org/10.1111/mice.12313).
- [32] L. Zhang, G. Zhou, Y. Han, H. Lin, and Y. Wu, "Application of Internet of Things technology and convolutional neural network model in bridge crack detection," *IEEE Access*, vol. 6, pp. 39442–39451, 2018, doi: [10.1109/access.2018.2855144](https://doi.org/10.1109/access.2018.2855144).
- [33] Y. Dong, J. Wang, Z. Wang, X. Zhang, Y. Gao, Q. Sui, and P. Jiang, "A deep-learning-based multiple defect detection method for tunnel lining damages," *IEEE Access*, vol. 7, pp. 182643–182657, 2019, doi: [10.1109/access.2019.2931074](https://doi.org/10.1109/access.2019.2931074).
- [34] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyükköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, Sep. 2018, doi: [10.1111/mice.12334](https://doi.org/10.1111/mice.12334).
- [35] S. Li, X. Zhao, and G. Zhou, "Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 34, no. 7, pp. 616–634, Jul. 2019, doi: [10.1111/mice.12433](https://doi.org/10.1111/mice.12433).
- [36] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," 2018, *arXiv:1802.02611*. [Online]. Available: <http://arxiv.org/abs/1802.02611>
- [37] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807, doi: [10.1109/cvpr.2017.195](https://doi.org/10.1109/cvpr.2017.195).
- [38] Y. Gao and K. M. Mosalam, "Deep transfer learning for image-based structural damage recognition," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 748–768, Sep. 2018, doi: [10.1111/mice.12363](https://doi.org/10.1111/mice.12363).
- [39] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: [10.1109/tkde.2009.191](https://doi.org/10.1109/tkde.2009.191).
- [40] E. Shelhamer, J. Long, and T. Darrell, *Fully Convolutional Networks for Semantic Segmentation*. Washington, DC, USA: IEEE Computer Society, 2017.



YUJIA SUN received the bachelor's degree in civil engineering from Chongqing University, in 2017, where he is currently pursuing the Ph.D. degree in structural engineering. His research interests include building structure construction and information and deep learning algorithms and their application in structural health monitoring.



YANG YANG received the Ph.D. degree in civil engineering from Chongqing University, China, in 2016.

Since 2016, she has been with the Faculty of the Key Laboratory of New Technology for Construction of Cities in Mountain Area, Ministry of Education, Chongqing University, where she is currently a Lecturer. She serves as the Director for several projects related to structural health monitoring for bridges. Her research interests include

structure health monitoring of infrastructures, construction safety, and application of building information model.



GANG YAO received the Ph.D. degree in civil engineering from Chongqing University, China, in 2002.

Since 1998, he has been with the Faculty of the Key Laboratory of New Technology for Construction of Cities in Mountain Area, Ministry of Education, Chongqing University, where he is currently a Professor. He serves as the Director for several projects related to civil construction technology. His research interests include building

structure construction and information, engineering project management, building industrialization and green construction, and deep learning algorithms and their application in structural health monitoring.



FUJIA WEI received the Ph.D. degree in civil engineering from Chongqing University, China, in 2020.

His research interests include building industrialization and green construction, engineering project management, and deep learning algorithms and their application in structural health monitoring.



MINGPU WONG received the bachelor's degree in civil engineering from Sichuan University, in 2017, and the master's degree in civil engineering from Chongqing University, in 2020, where he is currently pursuing the Ph.D. degree in structural engineering. His research interests include engineering project management and deep learning algorithms and their application in structural health monitoring.

• • •