FCN-SFW: Steel Structure Crack Segmentation Using a Fully Convolutional Network and Structured Forests

SEN WANG¹, YUNLONG PAN¹, MINGFANG CHEN², YINHUI ZHANG¹, AND XING WU¹,²

1Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, China
2Yunnan Vocational College of Mechanical and Electrical Technology, Kunming 650500, China

Corresponding author: Mingfang Chen (mfchen111@sina.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 52065035, Grant 51965029, and Grant 62061022; in part by the National Key Research and Development Plan Project under Grant 2017YFC1702503; in part by the Talent Development Project of Yunnan Province under Grant KKSY201801018; and in part by the Yunnan Provincial Department of Education Science Research Fund Project under Grant 201900045.

ABSTRACT

Tiny cracks that exist in steel beams have poor continuity and low contrast in images, posing a huge challenge to crack detection using image-based approaches. When complex backgrounds exist, the existing deep learning methods are usually unable to perform effective feature transfer and fusion for crack feature mapping, and they cannot accurately distinguish crack features from similar backgrounds. In this article, we propose a fusion segmentation algorithm, using the fully convolutional network (FCN) and structured forests with wavelet transform (SFW) to detect tiny cracks in steel beams. First, five neural networks based on the FCN framework are constructed to extend the global characteristics of tiny cracks. Second, a fine edge detection approach using multi-scale structured forests and wavelet maximum modulus edge detection to refine the characteristics of tiny cracks are proposed. Here, a competitive training strategy is used to address the SFW parameter optimization problem. Finally, we fuse the multiple probability maps, acquired from both the optimal FCN model and the SFW classifier, into a merged map, which can segment tiny cracks with robustness better than the comparison approaches. The experimental results show that compared with state-of-the-art algorithms and other segmentation approaches, the proposed algorithm realizes better segmentation in terms of quantitative metrics.

INDEX TERMS

Steel structure crack segmentation, fully convolutional network, structured forests, wavelet transform, maximum modulus edge detection.

I. INTRODUCTION

Tiny cracks, which have a length of less than 5mm and a width of less than 0.2mm, on the surface of a steel beam are an early symptom of deterioration. The traditional method that depends on an inspector’s judgement and expertise is still commonly used, but this kind of subtle defect is difficult to observe via manual visual inspection. Nondestructive detection techniques, such as ultrasonic testing, are suitable for detecting plane defects but they demand sound coupling. Magnetic particle inspection and eddy current testing have unique advantages in tiny crack detection, but the use of these technologies is limited by the properties of the material. Therefore, a considerable amount of research using image processing algorithms to improve detection has been performed. This research is indeed conducive to increasing industrial applications of computer vision inspection.

A wide variety of techniques, such as threshold processing [1], edge detection [2], image filtering [3], and regional texture description [4], have been proposed to solve crack segmentation problems. However, the complex environment and diverse cracks that are constantly encountered in the defect detection field, create a crucial yet non-trivial problem that should not be over-estimated, and tiny cracks on steel beams have poor continuity and low contrast in images, which pose considerable challenges for crack detection using image-based approaches in complex backgrounds. Recently, deep learning algorithms, such as FCN [5], SegNet [6],
ResNet [7] and DeepLABv3 [8], have been applied to the PASCAL VOC 2012 database to detect multiple targets, and these approaches have been found to be better than shallow segmentation approaches at crack detection. A multi-scale edge detection algorithm, such as fast edge detection using structured forests (SFD) [9], is an effective method for obtaining ideal target edges. These algorithms have some serious flaws. First, in the depth feature mapping stage, tiny cracks in larger resolution images are regularly omitted and classified as background, although attempts have been made to avoid this flaw by stacking higher-level deconvolution operations and using skip connections. Second, during multi-scale feature extraction, it is difficult to detect tiny cracks in uneven lighting because the filter leads to edge blur. Third, the detailed configurations of tiny cracks are frequently discontinuous because of the lack of effective feature transfer and fusion of feature mapping. Thus, the algorithms cannot accurately distinguish crack features from similar backgrounds. Moreover, using a lower camera resolution and shooting distance can effectively magnify the information of local cracks in steel beams and improve the recognition of tiny cracks. However, this method makes the camera unable to capture the global information of steel beams, leading to a considerable increase of time for image acquisition and crack detection.

To overcome these limitations of tiny crack detection, this article develops a provable approach to accurately distinguish crack features from similar backgrounds. The proposed FCN-SFW approach consists of three steps. The first step is an ablation study of the FCN framework, which is conducted to extend the global characteristics of microcracks. The second step is optimization of the parameters of the SFW classifier, which is conducted to distinguish refined tiny crack characteristics. The third step is the fusion of the feature mapping of the FCN and SFW, which can segment tiny cracks with robustness higher than the comparison approaches, as shown in our experimental studies. The flow of our FCN-SFW method is shown in Figure 1. The contributions and novelties of our paper are summarized as follows.

1. To improve the limitation that tiny cracks are easily overlooked as background and achieve a better classification performance with a limited dataset, five FCN-based neural networks are proposed to tackle the problems that ignore local refinement information. These neural networks are assessed using an ablation study that reduces or enlarges the depth of the network framework. We gradually reduce or deepen the depth of the FCN model and even deepen the convolutional component in the current convolution layer by duplicating the upper connection weights. The deconvolution component at lower or higher scales is utilized to expand the local fine features of tiny cracks. These FCN-based models are pre-trained on the PASCAL VOC 2012 dataset to obtain the initial parameters, and then these pre-trained models are further fine-tuned on a steel beam dataset to determine the optimal method.

2. This article proposes a fine edge detection approach using multi-scale structured forests and wavelet maximum modulus edge detection, to effectively distinguish refined tiny crack characteristics in images with uneven illumination. The maximum modulus of the wavelet half-reconstruction, instead of Gaussian filtering and gradient derivation, is adopted in our approach. Additionally, we propose a competitive training strategy for crack segmentation parameters to optimize the selection of the splitting parameters, classifier parameters, and characteristic parameters.

3. FCN models are typically good at obtaining the global information of tiny cracks, and SFD classifiers are suitable for extracting the fine details of tiny cracks. Utilizing these heterogeneous features may produce better a segmentation effect. We combine these advantages of both the optimal FCN-based model and the SFW classifier through probability mapping fusion.

The remainder of this article is organized as follows: we summarize related work on crack detection in Section II before we describe the FCN-SFW approach process in Section III. We compare the performances of algorithms in Section IV. We conclude our paper in the last section.

II. RELATE WORK
A. IMAGE PROCESSING-BASED APPROACH
Spatial filtering, wavelet transform, graph theory, and multi-scale image processing are popular approaches that extract local changes in image intensity to detect cracks. These approaches perform well for structure surfaces when the crack is clearly distinguished from the background. The bottom-hat transform [10] has made significant progress at reducing noise in crack images, but effective implementation of this method requires that the intensities of crack pixels are higher than those of noisy pixels. The Hough transform [11] is only suitable for the measurement of a curve or line in a fixed regular range. The crack width transform [12] and morphological operation with different sizes and multiple structural operators [13] exploit the connectivity in a specific shape for feature extraction, but they require parameter selection and adjustment. The non-subsampled shearlet transform [14] is suitable for detecting magnetic tile cracks in variable low-contrast images. The beamlet transform [15] was designed to extract line crack features, but the large amount of time- required is the primary flaw of the multi-scale geometric analysis method. CrackTree [16] and minimal path selection [17] use different endpoints to extract crack-like curves in noisy images, but these approaches cannot handle complex crack topologies. Multi-scale neighborhood information [18] and multi-scale image fusion [19] can solve the limitations of the poor crack location ability and sensitivity to clutter. However, these approaches are unable to realize complete detection of tiny cracks in low contrast images.

VOLUME 8, 2020
B. SUPERVISED LEARNING-BASED APPROACH

One promising solution method for crack detection is to utilize supervised learning for crack prediction. These methods include support vector machines (SVMs) [20], [21], artificial neural networks [22], the hybrid chromosome genetic algorithm [23], CrackIT [24], CrackBT [25], sketch tokens [26], and CrackForest [27], [28]. Exploiting effective feature descriptors, such as texture features [29], standard deviation, and mean parameters [30], is necessary to distinguish crack blocks from non-crack blocks. The subimages extracted from the trained crack images are used to generate feature vectors. However, these supervised training strategies require accurately labelled data to adapt to different scene variations. Recently, a combination of supervised learning with image processing that combines k-means clustering with the Canny operator [31] was implemented to extract...
Although attempts have been made to avoid this limitation by using convolutional neural network (CNN) technology [35], [36], several studies have sought to improve supervised learning using deep network structures [37]–[40]. Typical approaches applying a structured autoencoder or attention-guided CNN network are known as block-wise CNN methods, which include the DCNN [41], GoogleNet [42], CrackNet [43], and CrackNet II [44]. Although the CNN has powerful abilities, using image patches with a sliding window to locate cracks is unable to effectively improve the crack detection performance. To solve these problems, several networks, such as multi-scale classification networks [45], the NB-CNN [46], the faster RCNN [47], the mask-RCNN [48], and Crack-pot [49], have been used to enhance the robustness of crack classification. Nevertheless, all of the previous CNN-based networks using bounding boxes only mark the locations of cracks, and the rough prediction fails to identify cracks at the pixel-level. To improve the accuracy and efficiency of the object detection performance, a segmentation procedure has been developed to detect surface cracks of structures. This semantic crack segmentation approach is known as the pixel-wise CNN approach, and examples include the DeepLabv3-based network [50], the FCN-based network [51], [52], the encoder-decoder network [53], DeepCrack [54], U-Net [55], [56], FPCNet [57], and residual connections [58]. However, these approaches require a considerable amount of time to mark cracks at the pixel level. Moreover, the feature transfer between the convolutional layers of the shallow network becomes increasingly coarse in the convolution-pooling pipeline because of the lack of effective feature transfer and mapping fusion, and small objects, such as tiny cracks, are easily eliminated during down-sampling.

**C. DEEP LEARNING-BASED APPROACH**

Inspired by convolutional neural network (CNN) technology [35], [36], several studies have sought to improve supervised learning using deep network structures [37]–[40]. Typical approaches applying a structured autoencoder or attention-guided CNN network are known as block-wise CNN methods, which include the DCNN [41], GoogleNet [42], CrackNet [43], and CrackNet II [44]. Although the CNN has powerful abilities, using image patches with a sliding window to locate cracks is unable to effectively improve the crack detection performance. To solve these problems, several networks, such as multi-scale classification networks [45], the NB-CNN [46], the faster RCNN [47], the mask-RCNN [48], and Crack-pot [49], have been used to enhance the robustness of crack classification. Nevertheless, all of the previous CNN-based networks using bounding boxes only mark the locations of cracks, and the rough prediction fails to identify cracks at the pixel-level. To improve the accuracy and efficiency of the object detection performance, a segmentation procedure has been developed to detect surface cracks of structures. This semantic crack segmentation approach is known as the pixel-wise CNN approach, and examples include the DeepLabv3-based network [50], the FCN-based network [51], [52], the encoder-decoder network [53], DeepCrack [54], U-Net [55], [56], FPCNet [57], and residual connections [58]. However, these approaches require a considerable amount of time to mark cracks at the pixel level. Moreover, the feature transfer between the convolutional layers of the shallow network becomes increasingly coarse in the convolution-pooling pipeline because of the lack of effective feature transfer and mapping fusion, and small objects, such as tiny cracks, are easily eliminated during down-sampling.

**III. METHODOLOGY**

**A. THE NETWORK MODELS BASED ON FCNs**

Experimental results demonstrate that three FCN models can be effectively utilized to segment tiny cracks from steel beam images. The primary merit of FCN models is that these networks accept the whole crack image as an end-to-end input, but the classification performance is not high enough. Therefore, these models have several serious problems. First, an FCN network needs to set a predefined receptive field in advance. A tiny crack that is different from the region of the receptive field may be fragmented into multiple objects. Although attempts have been made to avoid this limitation by stacking higher-level deconvolution operations and using skip connections, tiny cracks in larger resolution images are often omitted and classified as background. Second, FCN models lack effective feature transfer in the convolution-pooling pipeline, and this deficiency can lead to the fine details being lost in complex and similar backgrounds. The partial segmentation results of the FCN-32, FCN-16, and FCN-8s are shown in Figure 2. To overcome these limitations, we build five neural networks based on the FCN framework; modify the FCN architecture, as shown in Figure 3 c), such that it yields more precise crack detections; and then select the optimal network model for crack segmentation, as illustrated in Figure 3.

In Figure 3 a), the FCN-8s-Conv4-Conv5 architecture indicates that the Conv4 and Conv5 convolution layers are removed on the basis of the FCN-8s architecture. The new feature channels, the conv1_3 and conv2_3 cells with ReLUs, are added before the max pooling of Conv1 and Conv2. In addition, the output map of the final deconvolution dec3 is integrated with the newly generated characteristic map after max pooling of Conv2 and Conv3. Figure 3 b) shows the deletion of Conv5 on the basis of the FCN-8s-Conv4-Conv5 framework, and dec3 is integrated with the characteristic map after max pooling of Conv3 and Conv4. In the FCN-8s+2conv architecture, as shown in Figure 3 d), we add the same conv1_3 and conv2_3 of FCN-8s-Conv4-Conv5 to the FCN architecture. One important modification of the FCN-8s+2conv+CONV6 architecture, as shown in Figure 3 e), is that we allow the FCN-8s+2conv network to propagate feature information to deeper convolution layers. Although the dropout technology can effectively alleviate the occurrence of overfitting, the use of this technology obviously increases the training time, and it is easy to lose detailed information when segmenting tiny cracks. Therefore, in the process of building the fully connected layer, we remove the dropout technology and add a higher scale deconvolution layer to expand the local-fine information choices. The FCN-8s+2conv+Conv6+Dec4 model is shown in Figure 3 f). We show the detailed process of the FCN-8s+2conv+Conv6+Dec4 architecture in Figure 4, and the parameters of each layer are shown in Table 1. We combine the initial models pre-trained on the
FIGURE 3. The six architectures.

FIGURE 4. The network architecture of the FCN-8s+2conv+Conv6+Dec4.
TABLE 1. The parameters of the FCN-8s+2conv+Conv6-Dec4 architecture.

| Layer | Stage | Filter size | Pad | Stride |
|-------|-------|-------------|-----|--------|
| 1     | conv1_1      | [3,3,3,64]  | [1,1,1,1] | [1,1]  |
| 1     | conv1_2      | [3,3,64,64] | [1,1,1,1] | [1,1]  |
| 1     | conv1_3      | [3,3,64,64] | [1,1,1,1] | [1,1]  |
| 1     | pooling1     | [2,2]       | [0,1,0,1] | [2,2]  |
| 2     | conv2_1      | [3,3,64,128]| [1,1,1,1] | [1,1]  |
| 2     | conv2_2      | [3,3,128,128]| [1,1,1,1] | [1,1]  |
| 2     | conv2_3      | [3,3,128,128]| [1,1,1,1] | [1,1]  |
| 2     | pooling2     | [2,2]       | [0,1,0,1] | [2,2]  |
| 3     | conv3_1      | [3,3,128,256]| [1,1,1,1] | [1,1]  |
| 3     | conv3_2      | [3,3,256,256]| [1,1,1,1] | [1,1]  |
| 3     | pooling3     | [2,2]       | [0,1,0,1] | [2,2]  |
| 4     | conv4_1      | [3,3,256,512]| [1,1,1,1] | [1,1]  |
| 4     | conv4_2      | [3,3,512,512]| [1,1,1,1] | [1,1]  |
| 4     | conv4_3      | [3,3,512,512]| [1,1,1,1] | [1,1]  |
| 4     | pooling4     | [2,2]       | [0,1,0,1] | [2,2]  |
| 5     | conv5_1      | [3,3,512,1024]| [1,1,1,1] | [1,1]  |
| 5     | conv5_2      | [3,3,1024,1024]| [1,1,1,1] | [1,1]  |
| 5     | conv5_3      | [3,3,1024,1024]| [1,1,1,1] | [1,1]  |
| 6     | conv6_1      | [3,3,1024,2048]| [1,1,1,1] | [1,1]  |
| 6     | conv6_2      | [3,3,2048,2048]| [1,1,1,1] | [1,1]  |
| 6     | conv6_3      | [3,3,2048,2048]| [1,1,1,1] | [1,1]  |
| 7     | fc7          | [7,7,2048,4096]| [3,3,3,3] | [1,1]  |
| 7     | fc8          | [1,1,4096,4096]| [0,0,0,0] | [1,1]  |
| 7     | fc9          | [1,1,4096,2]  | [0,0,0,0] | [1,1]  |
| 8     | dec1         | [4,4,2,2]    | [1,1,1,1] | [2,2]  |
| 8     | skip1        | [1,1,512,2]  | [0,0,0,0] | [1,1]  |
| 8     | dec2         | [4,4,2,2]    | [1,1,1,1] | [2,2]  |
| 8     | skip2        | [1,1,256,2]  | [0,0,0,0] | [1,1]  |
| 8     | dec3         | [4,4,2,2]    | [1,1,1,1] | [2,2]  |
| 8     | skip3        | [1,1,128,2]  | [0,0,0,0] | [1,1]  |
| 8     | dec4         | [8,8,1,2]    | [2,2,2,2] | [4,4]  |

PASCALVOC 2012 dataset with fine-tuning to train six FCN-based networks.

B. STRUCTURAL FOREST WITH WAVELET TRANSFORM

In the literature [9], the SFD method uses the crack training set for multi-channel feature extraction, and then performs discretization feature mapping with the generated features. The information gain theory can be used to train the deep nodes of different decision trees. Finally, a structured forest classifier composed of multiple trees is utilized for crack detection of steel structure images. It is an effective crack feature extraction algorithm that obtains edge contour information by using Gaussian filtering and the gradient derivative. However, in the tiny crack detection experiment, we found that the SFD method easily experiences crack loss or information redundancy when extracting crack edge features.

In this section, we propose a fine edge detection approach using multi-scale structured forests and wavelet maximum modulus edge detection to effectively distinguish refined tiny crack characteristics in uneven illuminative images. The anti-symmetric property of the two-dimensional discrete wavelet has been confirmed to be superior for edge detection than traditional detection algorithms [59]; the maximum modulus of the wavelet half-reconstruction can not only detect the position of a mutation and slow change but can also detect the singularity of a signal change [60]. Therefore, we adopt this edge detection algorithm to replace convolution filtering and gradient derivation. Additionally, we propose a competitive training strategy for crack segmentation parameters to optimize the selection of the splitting parameters, model parameters, and characteristic parameters. Moreover, we divide the characteristic attributes of cracks into 8 types of common eigenvectors (‘−’, ‘|’, ‘\’, ‘ς’, ‘L’, ‘T’, ‘Y’ and ‘◦’) by using the multi-scale SFW classifier. The feature vectors and structured labels are shown in Figure 5, and the flow chart of the SFW classifier is shown in Figure 6.

Generating random variables to maximize the information gain must provide enough characteristic channels. We select 13 feature channels for feature extraction of the structured segmentation mask. These channels include 3 colour channels, wavelet channels and 4 HOG orientation channels at two scales. We can obtain the regular output features and self-similarity output features after image filtering of 13 feature channels. The process of extracting the 26 feature channels of a crack image is shown in Figure 7.

Suppose a structured 16 × 16 mask is predicted from a larger 32 × 32 image patch. We blur the channel features with a triangle filter and down-sample these results by a fixed factor, which results in 32 × 32 × 13/4 + 300 × 1300 = 7228 candidate features. Then, we use the bagging algorithm to randomly choose some features from these candidate features and the corresponding ground truth (GT) masks to form the feature vectors and discretized mapping labels. These feature vectors and mapping labels can be combined into a constructed training set. We train each decision tree independently in a recursive manner. Each branch is constructed...
until all the splitting attributes of the node have been properly classified in the training phase. The attributes of these trees are integrated into the SFW classifier. The training flow of the SFW classifier is shown in Table 2.

The parameters of the SFW classifier mainly include split parameters (nSample), feature parameters (nCell, norRad, simSmooth and chSmooth), and classifier parameters (imWidth, gtWidth, fracFtrs, maxDepth, minChild, sharpen and nTree). We use the competitive training strategy, as shown in Figure 8, to optimize the selection of these parameters. Given a training image, we utilize the split function to vote on different parameter values in the belonging class. The binary classification results are judged by the 10 threshold values of the decisions from the SFW classifier:

\[
Q = \begin{cases} 
1, & \text{if } T_1 \leq p(c|x) \leq T_2 \\
0, & \text{else}
\end{cases}
\]  
(1)

\[
p(c|x) = \frac{1}{N} \sum_{n} p_n(c|x)
\]  
(2)

where \(p(c|x)\) is the posterior distribution obtained by the \(k\)th tree and \(c \in C = \{0, 1\}\) represents the discrete binary labels. \(T_1, T_2 \in [0.01, 1]\), and stride \(= 0.01\). We select the optimal values of each parameter \(K(n)\) in turn by using the relationship between the maximum recall (R) rate and the minimum mean absolute error (MAE):

\[
R = \frac{TP(Q)}{TP(Q) + FN(Q)}
\]  
(4)

\[
MAE = \frac{1}{J \times k} \sum_{j=1}^{J} \sum_{k=1}^{K} |Q(j, k) - G(j, k)|
\]  
(5)

where \(m\) is the number of verification pictures, \(n\) is the selected value of each parameter, \(j\) and \(k\) are the size parameters of the image, \(TP\) is the number of samples that are correctly classified as positive, \(FN\) is the number of samples that are incorrectly classified as negative, \(Q(j, k)\) represents the binary mapping result of the output, and \(G(j, k)\) represents the standard result image GT corresponding to the binary mapping.

### C. PROBABILITY MAPPING FUSION

FCN-based models are typically good at obtaining the global information of cracks, and SFD classifiers are suitable for extracting the fine details of tiny cracks. Moreover, the
classification error of the FCN-based model is lower than that of the SFW classifier. Utilizing these heterogeneous features may produce a better segmentation effect. We combine these advantages of both the optimal FCN-based model and the SFW classifier through probability mapping fusion. Therefore, we develop a mapping fusion method, called the FCN-SFW, to combine the two kinds of mapping results. Given the two probability maps of a tiny crack image calculated independently by the optimal FCN-based model and SFD-based classifier, we solve the union of both output maps with the same image resolution to obtain the final detection.

IV. CRACK DETECTION RESULTS AND ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

We comprehensively evaluate our FCN-SFW algorithm on the two types of crack datasets. In the first experiment, to simulate an industrial site to detect tiny cracks of steel beams, we designed a simulation experimental device based on field measurement data, as shown in Figure 9. The experimental device mainly includes seven parts: steel beams, two industrial cameras (DMK 72BUC02, CMOS, 1/2.5) with light sources (Deloda-60mm), a motor, a positioning sensor (Sharp-GP2Y0A02YK0F and NI9234 signal acquisition card), a crack collection device, and a movement device. The movement device is installed inside the positioning device, and two steel beams with tiny cracks are placed in parallel directly above the positioning device. We apply the positioning sensor to record the time passing through each baffle. When the motor drives the ball screw to rotate, the collection device undergoes a uniform linear motion. The positioning sensor triggers the industrial camera to collect the images.
of the steel beam. We set the optimal camera parameters (pixel resolution=4 million, focal length=0.22, and exposure time=0.001953125) to obtain high-definition crack images. We extract 1800 crack images from the acquired images to construct a steel beam crack database. These images in the database were separated into subsets for model training, validation, and testing at a ratio of 1:1:1. The size of the input images is 1280×960×3. In the second experiment, the homemade structural dataset includes a total of 2156 crack images with a size of 550×410×3. The numbers of images for training, validation and testing are 800, 580 and 776, respectively. To validate our FCN-SFW approach, we compare five neural networks with the FCN-8s in subsection IV-B. Parameter optimization of the SFD-based classifiers is performed on the same dataset, and the performance comparison of the six SFW classifiers with the SFD classifiers are given in the same subsection. We test six FCN-based models and six SFD-based classifiers in subsection IV-C and compare the 12 fusion algorithms to five neural networks, six SFW classifiers, an FCN model, SFD, SegNet, the DeepLab V3 model, ResNet models, multi-scale down-sampled normalized cut with wavelet edge detection (MDW Ncut) [60], the graph cut method (GCM) [61] and two traditional edge detection algorithms in subsections IV-D and IV-E.

The calculations for the FCN-SFW, FCN-based models, SFD-based classifiers, MDW Ncut and GCM were performed on computer 1 (an Intel (R) Core(TM)i5-4670 T6570 @ 3.40 GHz CPU, 64GB of RAM, Windows 7-64 bit, and MATLAB (R2018b)), and the calculations for SegNet, DeeplabV3, and ResNet were executed on computer 2 (an Intel (R) Core(TM)i7-9700F @3.00 GHz CPU, a GeForce RTX 2070S GPU, 32GB of RAM, Windows 7-64-bit, and TensorFlow). Similar to [5,62], we compare our results with those of other algorithms by using the mean accuracy (MACC), pixel accuracy (PACC), mean intersection over union (MIOU), recall (R), precision (P), F-measure (F) and mean absolute error (MAE).

### B. TRAINING AND VERIFICATION OF THE FCN-BASED MODELS AND SFD-BASED CLASSIFIERS

In this subsection, we evaluate the crack detection performance of six FCN-based models. Note that the FCN-8s model has been pre-trained on the PASCALVOC 2012 dataset to obtain the initial parameters, and the other five models are built on the basis of the FCN-8s model. The training parameters of the six models are set as learningRate=0.0001, batchSize=4, and numSubBatches=2. If the MIOU value stops growing within 500 epochs, then network training is terminated.

The training and validation results of the same epochs are shown in Figure 10. The FCN-8s model is able to maintain a fair objective during the whole training and verification stage. Since these FCN-based models are fine-tuned on the basis of the pre-trained FCN-8s model, the MIU increases slowly, starting at approximately 0.5. Furthermore, due to the elimination of the Conv4 convolution layer and the Conv5 convolution layer, the accuracy rate of the FCN-8s-Conv4-Conv5 model has a small fluctuation range, and the loss rate decreases the fastest. The fluctuation of the remaining three models is more serious than that of the FCN-8s model because of the increasing untrained component, but the loss rate of the FCN-8s+2conv+Conv6+Dec4 model in training is decreased the most among in these models. It can be seen from the comparison results that
as the training and verification times increase, eliminating part of the convolution layer (FCN-8s-Conv and FCN-8s-Conv4-Conv5) and integrating the front-end network feature map (FCN-8s+2conv+Conv6+Dec4) can effectively increase the global contour information; furthermore, only adding the convolution unit with the duplicated connection weights (FCN-8s+2conv) and the convolution layer (FCN-8s+2conv+Conv6) will make them lose the global characteristics in the pursuit of an excessively detailed classification.

According to the edge detection theory based on the maximum modulus of the wavelet (half-reconstruction of reverse biorthogonal 1.1 (hrbio1.1), rbio1.1, dyadic, sym2, coif1 and dmey) transform, we analyse the performances of the six SFD-based classifiers by using the competitive training strategy. We set all the initial parameters to their minimum and use the recall and MAE to evaluate the training and validation performance. Training and verification of the next parameter are conducted on the premise of selecting the last optimal parameter. The performance comparisons of the splitting parameters, feature parameters, and classifier parameters are shown in Figure 11. After using our competitive training strategy, as the value and type of the training parameters change, the classifier training accuracy begins to gradually improve. In the training process of most SFD-based algorithms, the nSample, nCell, chSmooth, imWidth, and nTree parameters need to be smaller, while the normRad and fracFtrs parameters should be larger. The simSmooth and sharpen parameters are relatively insensitive to multiple parameter settings. Among these SFD-based classifiers, the overall improvement effect of the first four classifiers in parameter training is better than that of the SFD classifier, while the SFW-coif1 classifier and SFW-dmey classifier have

**FIGURE 11.** The comparison results of different split parameters, model parameters, and characteristic parameters.
insignificant improvements in their training accuracy compared with the SFD classifier. These findings indirectly prove that the edge detection performances of rbio1.1, dyadic, and sym are superior to those of coif1 and dmey in crack edge detection. We can see from the optimization of nTree that the SFW-hrbio1.1 classifier has the highest detection efficiency compared with the other six classifiers.

C. COMPARISON OF THE FCN-BASED MODELS AND SFD-BASED CLASSIFIERS

To illustrate that the proposed algorithm can result in an effective improvement, we compare our FCN-based models and SFD-based classifiers with two original algorithms on the testing set of the steel beam dataset. We select the optimal parameters of the FCN-based models and SFD-based classifiers from subsection IV-B. The same dataset, training parameters and termination conditions are selected to validate the performance of the FCN-8s model and SFD classifier.

Qualitative comparisons of different algorithms are illustrated in Figure 12. By eliminating some convolution layers and adding convolution units, the FCN-8s-Conv4-Conv5 model, FCN-8s-Conv4-Conv5 model, and FCN-8s+2conv+Conv6+Dec4 model can obtain a higher classification accuracy. The SFW-hrbio1.1, SFW-rbio1.1, and SFW-dyadic classifiers yield much cleaner crack detection than the SFD, SFW-dyadic, SFW-sym2, SFW-coif, and SFW-dmey classifiers. Note that the suggested SFW-hrbio1.1 classifier can reliably obtain the tiny cracks of interest in a shorter computational time. Take the last column in Figure 12 as an example. The visual comparison results are shown in Figure 13. Compared with the standard result of manual annotation (Figure 13 b), the smaller network model can obtain more true-positive samples and fewer fracture cracks (Figure 13 c), and the gradual increase of the convolution layer does not produce an obvious improvement effect (Figure 13 d∼ Figure 13 g) unless more deconvolution layers are added into the network framework (Figure 13 h)). The better edge detection ability is an advantage of SFD-based classifiers (Figure 13 o)). Although the SFW-hrbio1.1 classifier can detect the most real positive samples (Figure 13 i)), these classifiers still cannot identify hidden cracks in darker areas (Figure 13 j∼ Figure 13 n)). We conclude that the FCN-8s-Conv4-Conv5 model is typically the best model at obtaining global information and the SFW-hrbio1.1 classifier is suitable for extracting the local details of tiny cracks, but neither method can effectively and completely detect tiny cracks.

The ROC curves shown in Figure 14 and the quantitative mean comparison provided in Table 3 illustrate that the FCN-8s-Conv4-Conv5 model outperforms all the other FCN-based models, and the only drawback of eliminating the convolution layers is a higher false-positive rate (Figure 13 c)) and computational load than those of the SFD-based classifier. We calculated the average running time of the FCN-based algorithms and SFD-based algorithms in Table 3. We found that although the SFD algorithm leads to the lowest average time-consumption, the accuracy of this algorithm is less than 6 percent compared with the FCN-8s-Conv4-Conv5 model, and the error rate is also higher than 0.68 percent compared with the FCN-8s+2conv+Conv6+Dec4 model. Moreover, the FCN-8s+2conv+Conv6+Dec4 model outperforms the FCN-8s model not only in identifying tiny cracks but also in reducing misclassifications. SFW-hrbio1.1 outperforms other six
TABLE 4. The results of Recall, Precision, F-measure, MAE and time.

| Algorithm   | R      | P      | F      | MAE   | time(s) |
|-------------|--------|--------|--------|-------|---------|
| FCN-8s-Conv4-Conv5+SFW-hrbio1.1 | 0.9095 | 0.6463 | 0.6925 | 0.0218 | 1.8821  |
| FCN-8s-Conv4-Conv5+SFW-hrbio1.1 | 0.9110 | 0.6467 | 0.6931 | 0.0234 | 2.0241  |
| FCN-8s-Conv4-Conv5+SFW-ysdadic | 0.8975 | 0.6483 | 0.6954 | 0.0180 | 2.1015  |
| FCN-8s-Conv4-Conv5+SFW-sym2 | 0.9025 | 0.6445 | 0.6900 | 0.0145 | 2.1654  |
| FCN-8s-Conv4-Conv5+SFW-cof1 | 0.8832 | 0.6395 | 0.6830 | 0.0155 | 2.1986  |
| FCN-8s-Conv4-Conv5+SFW-dmey | 0.8666 | 0.6330 | 0.6768 | 0.0126 | 2.1701  |
| FCN-8s-Conv4-Conv5+SFW-hrbio1.1 | 0.8627 | 0.6340 | 0.6753 | 0.0187 | 1.9127  |
| FCN-8s+SFW-hrbio1.1 | 0.8552 | 0.6319 | 0.6725 | 0.0182 | 2.7302  |
| FCN-8s+2conv+SFW-hrbio1.1 | 0.8434 | 0.6287 | 0.6679 | 0.0184 | 3.0853  |
| FCN-8s+2conv+Conv6+SFW-hrbio1.1 | 0.8336 | 0.6259 | 0.6641 | 0.0184 | 3.5673  |
| FCN-8s+2conv+Conv6+Dec4+SFW-hrbio1.1 | 0.8613 | 0.6336 | 0.6748 | 0.0181 | 3.8918  |
| FCN-8s+SFD | 0.8313 | 0.6253 | 0.6632 | 0.0114 | 3.2374  |
| FCN-8s | 0.6647 | 0.5784 | 0.5999 | 0.0049 | 2.6746  |
| SFD | 0.8031 | 0.6170 | 0.6519 | 0.0111 | 0.5488  |
| SegNet | 0.6970 | 0.5828 | 0.6057 | 0.0040 | 2.3581  |
| DeeplabV3 | 0.5944 | 0.5434 | 0.5544 | 0.0067 | 1.0257  |
| Resnet-152 | 0.5903 | 0.5417 | 0.5522 | 0.0066 | 0.9348  |
| Resnet-30 | 0.5351 | 0.5170 | 0.5211 | 0.0058 | 0.7274  |

FIGURE 13. Visual comparison results in the last column of Figure 12.

methods in terms of recall/precision. Compared with the FCN-8s model and the SFD classifier, the recall of the proposed FCN-8s-Conv4-Conv5 model can attain improvements of more than 18 percent and 6 percent, respectively. Therefore, the comprehensive performance of the FCN-8s-Conv4-Conv5 model and SFW-hrbio1.1 classifier is better than those of other algorithms.

FIGURE 14. The quantitative comparison results of the ROC curves.

D. COMPARISON OF THE FCN-SFW ALGORITHMS AND OTHER MODELS

This subsection reports the comparative experimental results of different fusion methods on the testing set of the steel beam dataset. We use the optimal algorithm (the FCN-8s-Conv4-Conv5 model and the SFW-hrbio1.1 classifier) from subsection IV-C as one of the FCN-SFW algorithms and integrate them with other FCN-based methods or SFD-based methods to form 12 FCN-SFW algorithms. For example, the name FCN-8s-Conv4-Conv5+SFW-hrbio1.1 represents the feature mapping fusion of the FCN-8s-Conv4-Conv5 model and the SFW-hrbio1.1 classifier. We also compare our FCN-SFW algorithms with several deep learning models on the same test dataset. We select the optimal parameters of the FCN-based models and SFD-based classifiers from subsection IV-B. The same dataset, training parameters and termination conditions are used to validate the performance of the DeepLabV3 model, and ResNet model. The experimental results prove that the DeepLabV3 models at 149100 epochs obtain the best performance. The best
numbers of epochs of the ResNet-50 model and ResNet-152 model are 149200 and 149600, respectively.

Some qualitative comparison results of different algorithms are shown in Figure 15. We can see that the results of the 12 fusion algorithms are closer to the corresponding standard results of manual segmentation. These fusion algorithms produce a higher classification accuracy than the other 6 deep learning algorithms. Therefore, the combination of the FCN-based model and SFD-based classifier, FCN-SFW, is consistently better than the other algorithms on various challenging tiny crack images. Figure 16 and Table 4 show that, the SFD algorithm also produces the lowest average time-consumption, but the accuracy of the FCN-8s-Conv4-Conv5+SFW-dyadic algorithm is 11 percent higher than the accuracy of the SFD algorithm. Among these fusion algorithms, the recall rate of the FCN-SFD algorithm with the worst fusion effect is also 3 percent higher than that of the SFD algorithm. The superiority of the FCN-8s-Conv4-Conv5+SFW-dyadic algorithm is not only reflected in the area covered by the ROC curve but also in the various quantitative accuracy indicators. Compared with the FCN-8s model and SFD classifier, the recall of the proposed FCN-8s-Conv4-Conv5+SFW-dyadic algorithm can attain improvements of more than 23 percent and 11 percent, respectively. Although the fusion of algorithms requires a certain amount of running time, it can greatly improve the crack segmentation accuracy.

E. COMPARISON OF THE TEST RESULTS WITH THE STRUCTURAL CRACK DATASET

This subsection reports the performance comparison on the structural crack dataset. The experiment is conducted to demonstrate the generalization and practicability of the proposed FCN-SFW algorithm. The parameters of the FCN-based models and SFD-based classifiers are set to the optimal values. For example, the number of epochs of FCN-8s-Conv4-Conv5 is set to 460; and the principal parameters of the SFW-hrbio1.1 classifier are selected as nSample = 256, nCell = 5, chSmooth = 16, gtWidth = 4, imWidth = 8, maxDepth = 32, minChild = 8, and nTree = 8. We choose the optimal models of DeepLab V3 (epochs = 56100), ResNet-50 (epochs = 59700), ResNet-152 (epochs = 60900), and SegNet (epochs = 6720) to compare the crack detection performance. The optimal parameter settings of the MDW Ncut algorithm, the GCM algorithm and the Canny operator are the same as those in the literature [60].
The partial qualitative comparison results of the 20 algorithms are shown in Figure 17. Compared with four deep network models (DeepLab V3, ResNet-50, ResNet-152, and SegNet), our FCN-SFW algorithm also achieves remarkable performance. The remaining algorithms generally have noticeably poor performance. Additionally, the proposed algorithm obtains higher recall and precision in Figure 18 and Table 5. These comparisons verify that applying the fusion strategy to detect cracks is indeed an effective method that deserves further consideration. The only disadvantage is that it increases the false positive errors and time-consumption for the calculations.

V. CONCLUSION

In this article, an image segmentation algorithm combining fully convolutional neural networks and multi-scale structured forests is proposed to effectively identify tiny cracks in steel beam images. To improve tiny crack detection performance.
accuracy and reduce misclassifications of real samples, five FCN-based network architectures are built to overcome the problem of networks ignoring local information. To effectively distinguish refined tiny crack characteristics in images with uneven illumination, we combine the multi-scale structural forests with wavelet maximum modulus edge detection to enhance the edge information of tiny cracks, and construct a competitive training strategy to optimize the selection of the splitting parameters, classifier parameters, and characteristic parameters. Finally, we fuse these heterogeneous features to obtain the best segmentation results. Experimental verification on two crack image databases proves that the proposed FCN-SFW algorithm can realize high-quality tiny crack segmentation.

Nevertheless, we can improve the performance of the algorithm in future research. First, future research will study backbones pretrained with ImageNet or other large datasets. Second, our crack segmentation work will focus on continuous dense predictions within small target boxes, to achieve more refined micro predictions. Third, we will use semi-supervised learning methods to improve the defects of insufficient crack samples and the time required for labelling. Fourth, convex optimization and feature reduction are effective techniques to reduce the computational complexity of a network model.

REFERENCES

[1] H. Kim, E. Ahn, S. Cho, M. Shin, and S.-H. Sim, “Comparative analysis of image binatization methods for crack identification in concrete structures,” Cement Concrete Res., vol. 99, pp. 53–61, Sep. 2017.

[2] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino, “Concrete crack detection by multiple sequential image filtering,” Comput.-Aided Civil Infrastruct. Eng., vol. 27, no. 1, pp. 29–47, Jan. 2012.

[3] 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.

[4] S. S. Bharathi, N. Redhakrishnan, and L. Priya, “Surface defect detection of rubber oil seals based on texture analysis,” in Proc. IEEE 4th Int. Conf. Signal Image Process. (CISP), Coimbatore, India, 2013, pp. 207–216.

[5] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 60–65, Apr. 2017.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 834–848, Apr. 2018.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[9] P. Dollar and C. L. Zitnick, “Fast edge detection using structured forests,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 8, pp. 1558–1570, Aug. 2015.

[10] E. Asovinu, S. Divya, S. Kardhepan, and T. Manikandan, “Mathematical morphology and bottom-hat filtering approach for crack detection on relay surfaces,” in Proc. Int. Conf. SMART Struct. Syst. (ICSSS), Chennai, India, Mar. 2013, pp. 108–113.

[11] L. Yao and J. Xiao, “Pantograph slide cracks detection method based on fuzzy entropy and Hough transform,” J. China Railway Soc., vol. 36, no. 5, pp. 58–63, 2014.

[12] H. Cho, H.-J. Yoon, and J.-Y. Jung, “Image-based crack detection using crack width transform (CWT) algorithm,” IEEE Access, vol. 6, pp. 60100–60114, 2018.

[13] D.-M. Tsai and D. E. R. Molina, “Morphology-based defect detection in machined surfaces with circular tool-mark patterns,” Measurement, vol. 134, pp. 209–217, Feb. 2019.

[14] C. Yang, P. Liu, G. Yin, and L. Wang, “Crack detection in magnetic tile images using nonsubsampled shearlet transform and envelope gray level gradient,” Opt. Laser Technol., vol. 90, pp. 7–17, May 2017.

[15] Z. Lin and X. Zhao, “Geometrical flow-guided fast beamlet transform for crack detection,” IET Image Process., vol. 12, no. 3, pp. 382–388, Mar. 2018.

[16] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, “CrackTree: Automatic crack detection from pavement images,” Pattern Recognit. Lett., vol. 33, no. 3, pp. 227–238, Feb. 2012.

[17] R. Ambhaz, S. Chambon, J. Idier, and V. Baltazar, “Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 10, pp. 2718–2729, Oct. 2016.

[18] D. Ai, G. Jiang, L. Siew Kei, and C. Li, “Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods,” IEEE Access, vol. 6, pp. 24452–24463, 2018.

[19] H. Li, D. Song, Y. Liu, and B. Li, “Automatic pavement crack detection by multi-scale image fusion,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 6, pp. 2025–2036, Jun. 2019.

[20] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme, “An innovative methodology for detection and quantification of cracks through incorporation of depth perception,” Mach. Vis. Appl., vol. 24, no. 2, pp. 227–241, Feb. 2013.

[21] K. Fernandes and L. Ciobanu, “Pavement pathologies classification using graph-based features,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2014, pp. 793–797.

[22] L. Li, L. Sun, G. Ning, and S. Tan, “Automatic pavement crack recognition based on BP neural network,” PROMET Traffic Transp., vol. 26, no. 1, pp. 11–22, Jan. 1970.

[23] H. Hu, Y. Liu, M. Liu, and L. Nie, “Surface defect classification in large-scale strip steel image collection via hybrid chromosome genetic algorithm,” Neurocomputing, vol. 181, pp. 86–95, Mar. 2016.

[24] H. Oliveira and P. L. Correia, “CrackIT—An image processing toolbox for crack detection and characterization,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2014, pp. 798–802.

[25] P. Shi, X. Fan, and G. Wang, “A novel underwater dam crack detection algorithm based on sonar images,” in Proc. 5th Int. Conf. Comput. Sci. Autom. Eng., 2016, pp. 452–456.

[26] F. Meng and A. Li, “Pavement crack detection using sketch token,” Procedia Comput. Sci., vol. 139, pp. 151–157, Jan. 2018.

[27] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, “Automatic road crack detection using random structured forests,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 12, Dec. 2016.

[28] Q. Luo, B. Ge, and Q. Tian, “A fast adaptive crack detection algorithm based on a double-edge extraction operator of FSM,” Construct. Building Mater., vol. 204, pp. 244–254, Apr. 2019.

[29] A. Cord and S. Chambon, “Automatic road defect detection by textural pattern recognition based on AdaBoost,” Comput.-Aided Civil Infrastruct. Eng., vol. 27, no. 4, pp. 244–259, Apr. 2012.

[30] H. Oliveira and P. L. Correia, “Automatic road crack detection and characterization,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 1, pp. 155–168, Mar. 2013.

[31] D. Lattanzi and G. R. Miller, “Robust automated concrete damage detection algorithms for field applications,” J. Comput. Civil Eng., vol. 28, no. 2, pp. 253–262, Mar. 2014.

[32] L. Wu, S. Mokhtari, A. Nazef, B. Nam, and H.-B. Yun, “Improvement of crack-detection accuracy using a novel crack defragmentation technique in image-based road assessment,” J. Comput. Civil Eng., vol. 30, no. 1, Jan. 2016, Art. no. 04014118.

[33] G. P. Bu, S. Chanda, H. Guan, J. Jo, M. Blumenstein, and Y. C. Loo, “Crack detection using a texture analysis-based technique for visual bridge inspection,” Electron. J. Struct. Eng., vol. 14, no. 1, pp. 41–48, Jan. 2015.

[34] R. Davoudi, G. R. Miller, and J. N. Kutz, “Structural load estimation using machine vision and surface crack patterns for shear-critical RC beams and slabs,” J. Comput. Civil Eng., vol. 32, no. 4, Jul. 2018, Art. no. 04018024.

[35] D. Yuan, X. Li, Z. He, Q. Liu, and S. Lu, “Visual object tracking with adaptive structural convolutional network,” Knowl.-Based Syst., vol. 194, Apr. 2020, Art. no. 105554.

[36] D. Yuan, N. Fan, and Z. He, “Learning target-focusing convolutional regression model for visual object tracking,” Knowl.-Based Syst., vol. 194, Apr. 2020, Art. no. 105526.
[37] P. Feng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, “Structured AutoEncoders for subspace clustering,” IEEE Trans. Image Process., vol. 27, no. 10, pp. 5076–5086, Oct. 2018.

[38] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, “Deep clustering with sample-assignment invariance prior,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4857–4868, Nov. 2020, doi: 10.1109/TNNLS.2019.2958324.

[39] C. Tian, Y. Xu, and W. Zuo, “Image denoising using deep CNN with batch renormalization,” Neural Netw., vol. 121, pp. 461–473, Jan. 2020.

[40] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, “Attention-guided CNN for image denoising,” Neural Netw., vol. 124, pp. 117–129, Apr. 2020.

[41] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2016, pp. 3708–3712.

[42] S. J. Schmugge, L. Rice, N. R. Nguyen, J. Lindberg, R. Grizzi, C. Joffe, and H.-W. Huang, Q.-T. Li, and D.-M. Zhang, “Deep learning based image wavelet reconstruction,” Acta Automatica Sinica, vol. 36, no. 4, pp. 475–487, 2010.

[43] S. Wang, X. Wu, Y. H. Zhang, and Q. Chen, “Image crack detection with multi-scale down-sampled normalized cut,” Chin. J. Sci. Instrum., vol. 38, no. 11, pp. 2788–2796, 2017.

[44] A. Zhang, K. C. P. Wang, Y. Fei, Y. Liu, S. Tao, C. Chen, J. Q. Li, and B. Li, “Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet,” J. Comput. Civil Eng., vol. 32, no. 5, Sep. 2018, Art. no. 04018041.

[45] H. Feng, G.-S. Xu, and Y. Guo, “Multi-scale classification network for road crack detection,” IET Intell. Transp. Syst., vol. 13, no. 2, pp. 398–405, Feb. 2019.

[46] 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.

[47] J. Li, X. Zhao, and H. Li, “Method for detecting road pavement damage based on deep learning,” Proc. SPIE, vol. 10972, Apr. 2019, Art. no. 109722D.

[48] R. Kalfarisi, Z. Y. Wu, and K. Soh, “Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization,” J. Comput. Civil Eng., vol. 34, no. 3, May 2020, Art. no. 04020010.

[49] S. Anand, S. Gupta, V. Darbari, and S. Kohli, “Crack-pot: Autonomous road crack and pothole detection,” in Proc. Digit. Image Comput., Techn. Appl. (DICTA), Dec. 2018, pp. 1–6.

[50] Q. Song, Y. Wu, X. Lin, Y. Yang, M. Yang, H. Chen, C. Liu, M. Hu, X. Chai, and J. Li, “Real-time tunnel crack analysis system via deep learning,” IEEE Access, vol. 7, pp. 64186–64197, 2019.

[51] H.-W. Huang, Q.-T. Li, and D.-M. Zhang, “Deep learning based image recognition for crack and leakage defects of metro shield tunnel,” Tunneling Underground Space Technol., vol. 77, pp. 166–176, Jul. 2018.

[52] 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.

[53] S. Zhang, S. Park, H. Kim, and H. Kim, “Encoder–decoder network for pixel-level road crack detection in black-box images,” Comput.-Aided Civil Infrastruct. Eng., vol. 34, no. 8, pp. 713–727, Aug. 2019.

[54] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, “DeepCrack: Learning hierarchical convolutional features for crack detection,” IEEE Trans. Image Process., vol. 28, no. 3, pp. 1498–1512, Mar. 2019.

[55] 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.

[56] M. David Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, and G. Morison, “A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks,” in Proc. 26th Eur. Signal Process. Conf. (EUSIPCO), Sep. 2018, pp. 2120–2124.

[57] W. Liu, Y. Huang, Y. Li, and Q. Chen, “FPCNet: Fast pavement crack detection network based on encoder-decoder architecture,” 2019, arXiv:1907.02248. [Online]. Available: http://arxiv.org/abs/1907.02248.

[58] J. Konig, M. David Jenkins, P. Barrie, M. Mannion, and G. Morison, “A convolutional neural network for pavement surface crack segmentation using residual connections and attention gating,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2019, pp. 1460–1464.

**SEN WANG** received the Ph.D. degree in mechanical and electronic engineering from the Kunming University of Science and Technology, Kunming, China, in 2017. He is currently a Lecturer with the Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology. His research interests include image processing, computer vision, and deep learning.

**YUNLONG PAN** received the B.S. degree from the City College, Donggung University of Science and Technology, in 2019. He is currently pursuing the M.S. degree with the Kunming University of Science and Technology. His main research interests include machine vision detection and deep learning algorithms.

**MINGFANG CHEN** received the Ph.D. degree in mechanical manufacturing and automation from Northeastern University, Shenyang, China, in 2015. He is currently an Associate Professor with the Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology. His research interests include complex electromechanical system integration and application, circuit analysis and engineering applications, and intelligent control theory and applications.

**YINHUI ZHANG** received the Ph.D. degree in mechanical design and theory from the Kunming University of Science and Technology, Kunming, China, in 2010. He is currently a Professor with the Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology. His research interests include image processing, computer vision, and machine intelligence.

**XING WU** received the Ph.D. degree in equipment condition monitoring and fault diagnosis from Shanghai Jiao Tong University, Shanghai, China, in 2005. He is currently a Professor with the Kunming University of Science and Technology and the Yunnan Vocational College of Mechanical and Electrical Technology. His research interests include fault diagnosis, modern signal processing technology and its application in fault diagnosis, intelligent diagnosis theory, and system of equipment.