IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 5, 2022, accepted February 27, 2022, date of publication March 8, 2022, date of current version March 11, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3156606

MSCNet: A Framework With a Texture Enhancement Mechanism and Feature Aggregation for Crack Detection

GUANLIN LU, XIAOHUI HE, QIANG WANG, FAMING SHAO, JINKANG WANG, AND XIAOKANG ZHAO

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

Corresponding author: Xiaohui He (gcbhxh314@163.com)

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

ABSTRACT Bridge crack is one of the critical optical and visual information to judge the health state of bridges. The bridge crack detection methods based on artificial intelligence are essential in this field, but the current approaches are not satisfactory in terms of speed and accuracy. This study proposes a novel multi-scale crack detection network, called MSCNet, comprising a texture enhancement mechanism and feature aggregation to enhance the visual saliency of the objects in the background for bridge crack detection. We use Res2Net as the backbone network to improve the depth information expression ability of the cracks itself. Because the edge property of bridge cracks is prominent, to make full use of this visual feature, we use a texture enhancement module based on group attention to capturing the detailed information of cracks in low-level features. To further mine the depth information of the network, we use a cascade fusion module to capture crack location information in high-level features. Finally, to fully utilize the characteristic information of the deep network, we fuse the low- and high-level features to obtain the final crack prediction. We evaluate the proposed method compared with other state-of-the-art methods on a large-scale crack dataset. The experimental results demonstrate the effectiveness and superiority of the proposed method, which achieves a precision of 93.5%, recall of 94.2%, and inference speed of over 63 FPS.

INDEX TERMS Crack detection, deep learning, feature aggregation, grouping attention mechanism, texture enhancement.

I. INTRODUCTION

Bridges are an important part of traffic lines and mainly used for railways, highways, channels, pipelines, and people to cross rivers, valleys, or other obstacles. However, bridges suffer from damages due to various natural or human factors. Among the damages, crack formation is a common problem affecting bridge services. Cracks in a bridge accelerate corrosion of the armature, resulting in deterioration of the bridge structure [1]. Furthermore, cracks affect the integrity, durability, and seismic performance of a bridge and considerably reduce the bridge quality [2]. Hence, prompt detection and repair of cracks are essential for the engineering community, national government administrative services, and bridge construction companies to maintain the bridges in a healthy state. Among the nondestructive evaluation technologies for bridge health monitoring, visual inspection is the most used method [3]. However, because the visual detection technology alone is not sufficient to evaluate the internal condition of bridge structural members, other in-depth methods should be introduced for a more comprehensive inspection. Research has shown that the surface crack is the most obvious index of possible deterioration or damage of structures; therefore, the detection of surface cracks is essential for timely evaluation of the health status of bridges [4].

The conventional visual inspection method depends on the naked-eye observation by maintenance personnel. This manual crack detection method is not only time-consuming and labor-intensive but also lacks in safety. In addition, the inspection results are highly dependent on the maintenance personnel's subjective judgment, possibly leading to oversight or inaccurate inspection [5], [6]. With the vigorous development

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

of digital cameras and image processing technology, many image-based crack detection methods have been proposed. Sharma *et al.* [7] presented a model based on the support vector machine to detect concrete cracks. Talab *et al.* [8] used several simple image filters to design a multi-sequential image filter to observe cracks. Santos *et al.* [9] presented a mathematical morphology-based model for detecting cracks on concrete structures. Prasanna *et al.* [10] developed an automatic crack detection algorithm called STRUM (spatially tuned robust multifeatured) classifier to detect cracks on the bridge surface. Although these methods improve the detection accuracy and speed compared with traditional crack detection technologies, they cannot cover all the unforeseen circumstances in complex environments.

Convolution neural networks (CNNs) have made significant progress in object detection [11]–[15]. As an important application of computer vision, object detection mainly aims to locate an object of interest in an image and accurately judge the object's specific category. Many researchers have applied the CNN to crack detection. Chen et and Jahanshahi [16] presented a CNN combined with naive Bayes data fusion (NB–CNN) to detect cracks; however, the crack detection accuracy of this algorithm is low. To achieve higher detection accuracy, Cha *et al.* [17] proposed a network based on the Faster region-based CNN [18] to detect cracks. Dung and Anh [19] proposed a fully convolutional network to detect cracks. This paper designs a new visual recognition network called MSCNet to detect surface cracks. The research focuses on improving the accuracy and speed of existing surface crack detection methods, which is convenient to excellent good performance in practical detection tasks.

The main contributions of this study are as follows:

- We present a simple but effective crack texture enhancement module (TEM) to capture the details of cracks in low-level features, which will increase the anti-noise capability of the network.
- To balance accuracy and consumption of computation resources, we introduce a cascaded fusion module (CFM) in the framework to reduce the complexity of the deep aggregation network and collect location information of cracks from high-level features.
- In addition, we adopt a feature aggregation module (FAM) to aggregate the high- and low-level features obtained from the CFM and TEM and consequently express the higher-order relationship between the features of both levels.

The rest of the paper is organized as follows. In section 2, we review related work. In section 3, we present the details of the proposed method. In section 4, experiments are designed to evaluate the proposed network, and the results are presented. Finally, in section 5, a brief conclusion for the paper is presented.

II. RELATED WORKS

Crack formation is the most common degradation phenomenon and the primary problem of concrete bridges; thus, accurate, and timely detection of bridge cracks is critical in the daily maintenance of bridges. Based on crack detection, the structural stability of bridges can be scientifically and effectively evaluated and high-risk parts can be repaired. Bridge crack detection methods based on image recognition improve the efficiency of bridge crack visual detection.

Traditional bridge crack detection methods based on image recognition mainly employ image preprocessing technology and machine learning classification algorithms. Chanda *et al.* [20] proposed a detection method of concrete surface cracks based on an image penetration model. Zalama *et al.* [21] proposed a method based on the Gabor filter to detect longitudinal and transverse cracks. Shi *et al.* [22] proposed a road crack detection framework based on random structured forests. Although these traditional crack detection methods based on image recognition are considerably effective compared with manual inspection, they are highly dependent on complex classifiers and image processing, leading to low efficiency and weak generalization ability of the methods.

With the development of artificial intelligence technology, especially the breakthrough of deep learning technology in computer vision, image-based crack detection has a new development opportunity. The convolution layer in the deep CNN can automatically learn the crack characteristics from the image by using the error back propagation of the gradient descent, which effectively separates the crack from its background. Many crack detection algorithms based on deep learning have been recently proposed. Zhang *et al.* [23] designed a framework that can be directly applied to the original crack image for automatic feature extraction and classification, and the framework achieved superior performance compared to traditional handcrafted methods. Pauly *et al.* [24] improved classification accuracy and recognition by adopting a deeper neural network to classify crack and non-crack patches. Qiao *et al.* [25] proposed a framework called DFANet with a strong anti-interference capability and improved robustness; the framework consists of deep feature aggregation and attention mechanism. Li *et al.* [26] processed CliqueNet, a network to distinguish cracks rapidly and accurately from the background. Yang *et al.* [27] employed a multiscale feature pyramid and hierarchical boosting-based network (FPHBN). Zhou *et al.* [28] considered the semantic differences between different feature layers to process a network for crack detection. Fan *et al.* [29] proposed a novel road crack detection algorithm based on deep learning and adaptive thresholding. T. Ahmad *et al.* [30] proposed a novel network based on YOLOv1by modifying loss function and adding spatial pyramid pooling layer. Zou *et al.* [31] proposed an end-to-end trainable method to automatically detect cracks by taking full use of the information of the encoder and decoder network. Li *et al.* [32] designed a network composed of skip-squeeze-excitation and the atrous spatial pyramid pooling to detect cracks. Kim *et al.* [33] applied the semantic segmentation technique to develop a hierarchical convolutional

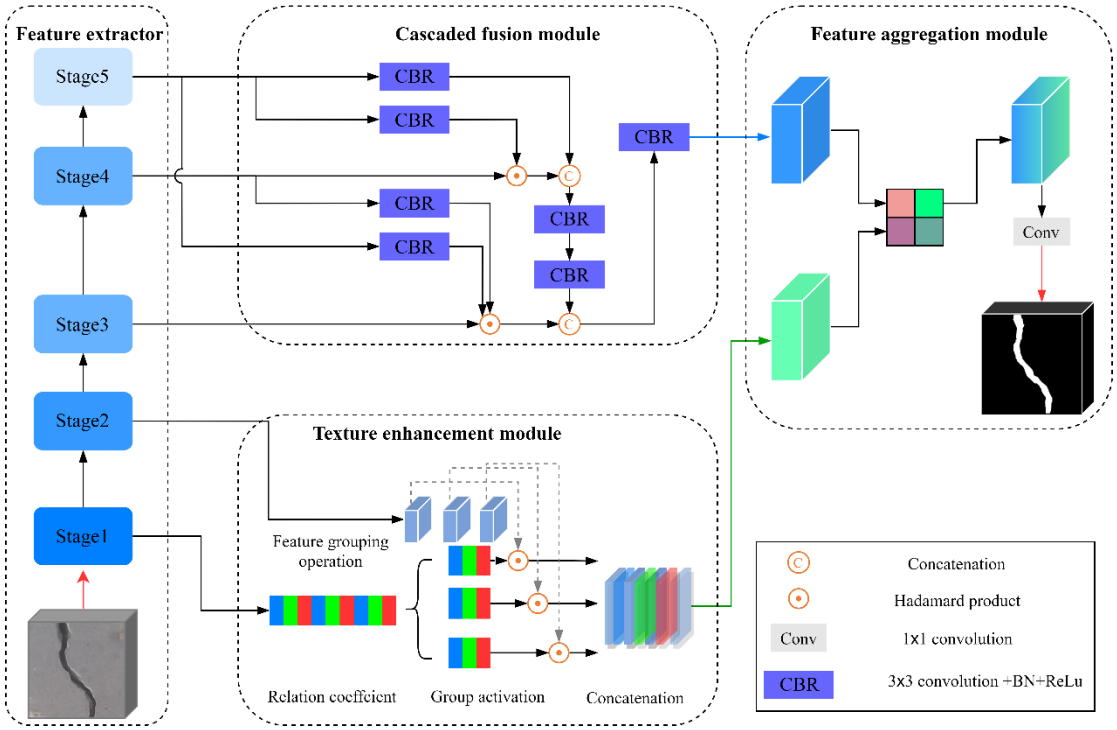


FIGURE 1. Structure of the proposed MSCNet, which consists of the feature extractor, TEM, CFM, and FAM.

neural network to improve the accuracy and rate of crack detection.

Despite their advances, the existing crack detection networks suffer from the following limitations. First, most CNN-based networks overlook the importance of low-level features, which often contain rich details and are beneficial for distinguishing cracks. Second, most networks have poor robustness, and the detection accuracy is easily disturbed by the environment. Third, most networks adopt a simple aggregation such as concatenation and addition for feature fusion method for multi-layer feature fusion without considering the different roles of low-level and high-level features in crack detection, which leads to a waste of computing resources and cannot capture valuable information that are strongly related to crack detection; thus, the detection methods inevitably have unsatisfactory performance. In order to optimize the limitations of detection accuracy and detection speed, we have designed a new and efficient bridge crack detection network termed MSCNet. MSCNet adopts the TEM to enhance the network's ability to extract the low-level detail information of cracks and strengthen the anti-interference ability of the network. MSCNet adopts the CFM to reduce the calculation resource consumption of the network for deep feature reasoning and speed up the reasoning speed of the network. Based on the above two modules, the network adopts the FAM to strengthen the relationship between low-level features and high-level features; it improves network detection accuracy.

III. OVERVIEW OF MSCNet

Fig. 1 demonstrates the overall crack detection framework of the proposed MSCNet. It mainly consists of four parts: a feature extractor, texture enhancement module (TEM), cascaded fusion module (CFM), and feature aggregation module (FAM). In particular, the feature extractor is used to extract features from input images. The TEM is applied to reduce noise and enhance low-level representation cues of cracks. The CFM collects the semantic and location information of cracks in high-level features by utilizing progressive aggregation. The FAM is designed to fuse low- and high-level features obtained from the TEM and CFM for identifying cracks accurately. As shown in Fig. 1, given an image $I \in \mathbb{R}^{H \times W \times 3}$, the feature extractor can obtain a set of feature maps x_i, whose resolution is ${}^{H}/_{2^i} \times {}^{W}/_{2^i}$ ($i \in \{1, 2, 3, 4, 5\}$). Then, x_3, x_4, and x_5 are fed to the CFM for fusion, generating a feature map T_2. Meanwhile, low-level features x_2 and x_1 are converted to feature map T_1 through the TEM. Subsequently, the feature maps T_1 and T_2 are aggregated by the FAM. Finally, the 1×1 convolution is adopted to adjust the dimensions to predict the detection result.

A. FEATURE EXTRACTOR

The input image consists of a relatively small proportion of crack pixels and different bridge crack scales. The cracks have diverse shape, making feature extraction of bridge cracks considerably difficult. To enhance the multiscale feature extraction ability of the network, Res2Net [34] mainly

uses multiple available receptive fields at a more fine-grained level, unlike most other networks that adopt features with different resolutions. Multiscale receptive fields in the human visual system are experimentally proven to be beneficial to focus on small objects [35]. This motivates us to consider Res2Net-50 (without the top three layers) as the feature extractor to highlight the regions of the cracks.

B. TEXTURE ENHANCEMENT MODULE

Crack pixels account for a small proportion of image pixels. Their significant features, such as texture and shape, are easily disturbed by environmental noise, dramatically affecting the detection accuracy. The original image and five-stage feature maps of the feature extractor are shown in Fig. 2. Among them, the high-level feature layers pay more attention to the semantic information of the crack, and the extracted features are abstract, which is not conducive to accurate crack detection. By contrast, low-level feature maps (stage1 and stage2) are rich in detail information, which plays an irreplaceable role in the regression of crack detection. However, the background noise in the low-level features drastically interferes with the extraction of crack texture information. Hence, we introduce a TEM to enhance the crack texture information captured from low-level features, resulting in improved and effective extraction of crucial semantic information.

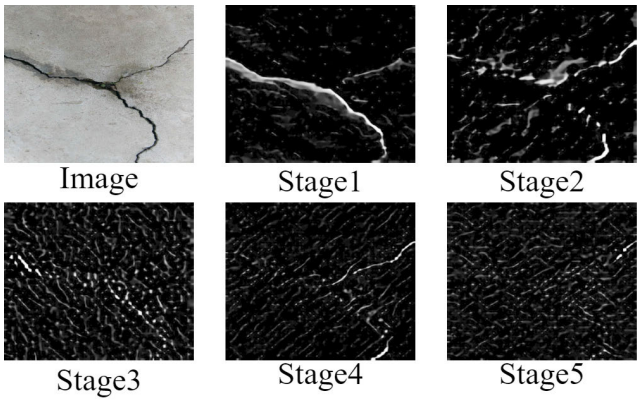


FIGURE 2. Original image and five-stage feature maps from Res2Net.

We adopt the grouping attention mechanism [36] to design a TEM. As shown in Fig. 1, the TEM has two inputs, the feature maps x_1 and x_2 from the feature extractor and the output of the feature map, T_1, which is enhanced by the TEM. Specifically, global average pooling is used to reduce the dimension of feature map x_2 and obtain parameter z_2 that only retains the dimension of feature channels; this process can be summarized as follows:

$$z_2 = \frac{1}{WH} \sum_{h=1}^{W} \sum_{w=1}^{H} x_2(h, w) \tag{1}$$

where z_2 represents the parameter of the feature channel dimension and $x_2(h, w)$ is the channel vector of feature map x_2 at height H and width W.

The relation coefficient y_1 of feature map x_1 can be calculated in terms of z_2 as shown below:

$$y_1 = A_1 \cdot z_2 \tag{2}$$

where A_1 is the parameter for calculating the characteristic relation coefficient of feature map x_1.

The overall TEM mainly consists of the following four stages:

1) Since features have different semantic concepts, grouping activation of the feature relation coefficients can effectively avoid inhibition between different semantic features and enhance similar semantic features. Feature map x_1 and the relation coefficient y_1 of feature map x_1 are divided into k groups to generate feature map x_1^l and relation coefficient $y_1^l; l \in [1, \cdots, k]$.

2) To avoid the disappearance of the original feature information as the relation coefficient tends to 0, the normalized relation coefficient y_1^l is added by 1.

3) Grouping activates the feature relation coefficient and enhances feature map x_1. The operation can be summarized as follows:

$$\hat{x}_1^l = x_1^l \cdot \left(1 + S\left(y_1^l\right)\right), \quad l \in [1, \ldots, k] \tag{3}$$

where $\hat{x}_1^l$ represents the l-th group enhanced features of feature map x_1, x_1^l is the feature of the l-th group of feature map x_1, y_1^l represents the relation coefficient of the l-th group of feature map x_1, and $S(\cdot)$ is the SoftMax activation function.

4) The obtained enhanced feature $\hat{x}_1^l$ is concatenated to generate T_1, as shown below:

$$T_1 = concat(\hat{x}_1^l), \quad l \in [1, \ldots, k] \tag{4}$$

C. CASCADED FUSION MODULE

Fig. 3 shows the performance of the output at different stages in the feature extractor. In Fig. 3, maxF represents the maximum F-measure of the five-stage outputs of the original Res2Net in the crack forest dataset [22], and we set the inference time of the backbone as 1 and indicate the inference time of the output of each stage. In order to highlight the performance of low-level features in the process of network depth feature aggregation, we arrange the abscissa from large to small in flashback when making charts. As the Fig. 3 shows, the feature extraction performance saturates rapidly as features are aggregated from the high-stage 5 to the low-stage 1. Moreover, the integration of low-level features with high-level features significantly increases computational complexity. As suggested in [35], high-level features with a low resolution represent semantic information, while low-level features with a high resolution represent spatial details. To improve the accuracy and reduce the consumption of computing resources, we design the CFM to aggregate the top three high-level feature maps.

As shown in Fig. 1, the CFM mainly adopts convolution, batch normalization, and ReLu (CBR) and a resize operation for fusion pretreatment. In addition, it employs concatenation and the Hadamard product to fuse different features.

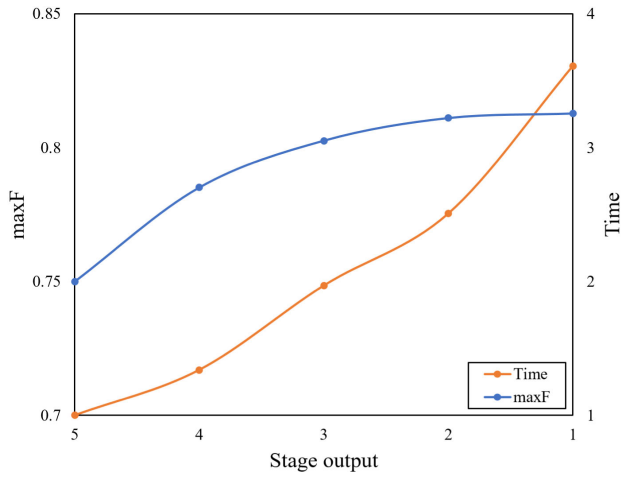


FIGURE 3. Performance of the feature extractor at different stages.

The process, as shown in Eqs. 5–7. First, feature map x_5 is resized and passed through two CBRs to obtain x_5^1 and x_5^2. Then, x_5^1 is concatenated with feature map x_4, and the result is passed through a CBR for smoothening it, yielding feature map x_{45}.

$$x_5^1 = c_1(x_5) \tag{5}$$
$$x_5^2 = c_2(x_5) \tag{6}$$
$$x_{45} = c\left(concat(x_5^1 \odot x_4, x_5^2)\right) \tag{7}$$

Using the same process, other feature maps are fused by resizing feature maps x_5, x_4, and x_{45} to the same size as x_3 and smoothing them with different CBRs. Then, the smoothed results of x_5 and x_4 are multiplied with x_3, and the smoothed result of x_{45} is concatenated. Finally, the concatenation result is fed to the CBR to reduce the dimension and obtain T_2. The process can be summarized as follows.

$$T_2 = c(concat(c(x_5) \odot c(x_4) \odot x_3, c(x_{45})) \tag{8}$$

D. FEATURE AGGREGATION MODULE

To explore the high-order relationships between the low-level local features of the TEM and the high-level cues of the CFM, we adopt the FAM to inject detailed appearance features into high-level semantic features. The structure of the FAM is shown in Fig. 4.

Self-attention is used to fuse feature map T_2 containing high-level semantic information and feature map T_1 containing rich textural details. For T_2, two 1×1 convolutions $C_A(\cdot)$ and $C_B(\cdot)$ are adopted for linear mapping to reduce the dimension and obtain feature maps Q and K, as given below.

$$Q = C_A(T_2) \tag{9}$$
$$K = C_B(T_2) \tag{10}$$

By contrast, a 1×1 convolution $C_C(\cdot)$ is used to reduce the channel dimension of T_1 and interpolate it to the same size as T_2. Then, the SoftMax function is applied to the channel dimension, and the second channel is selected as the attention map to obtain T_1', as shown in Eq. 11.

$$T_1' = S(C_C(T_1)) \tag{11}$$

To assign different weights to different pixels and increase the weight of edge pixels, the Hadamard product between K and T_1' is calculated. Subsequently, an adaptive pooling operation is adopted to limit the displacement of features, and a center crop operation is performed to obtain feature map V, as shown in Eq. 12.

$$V = AP\left(K \odot T_1'\right) \tag{12}$$

where $AP(\cdot)$ represents the adaptive pooling and crop operation.

Then, an inner product is used to establish connections between each pixel in V and K to obtain the correlation attention map g:

$$g = \sigma\left(V \otimes K^T\right) \tag{13}$$

where $\otimes$ is the inner product operation and K^T is the transpose of K.

Next, feature map Q multiplied with the correlation attention map g is fed to the graph convolutional network (GCN) layer [37]. The graph domain features are reconstructed into the original structural features by calculating the inner product between g and the output of the GCN layer:

$$D = g^T \otimes GCN(g \otimes Q) \tag{14}$$

To obtain the final result Z of the FAM, the dimensions of the reconstructed feature map D are updated and the reconstructed feature map D is combined with feature map T_2:

$$Z = T_2 + D \tag{15}$$

E. LOSS FUNCTION

The loss function for the proposed network is calculated between the final detection result P and the ground truth G, as follows:

$$L = L_{IOU}^w(P, G) + L_{BCE}^w(P, G) \tag{16}$$

where $L_{IOU}^w(\cdot)$ is the weighted intersection over union (IOU) loss [38] and $L_{BCE}^w(\cdot)$ is the weighted binary cross-entropy (BCE) loss [39]. Compared with the standard IOU and BCE losses, which treat all pixels equally, $L_{IOU}^w(\cdot)$ and $L_{BCE}^w(\cdot)$ consider the importance of the individual pixel and focus on hard pixels. Both weighted loss functions can restrict the prediction map in terms of the global structure (object level) and local detail (pixel level) perspectives.

IV. EXPERIMENTS AND RESULTS

This section first introduces the related content of the implementation (i.e., datasets, evaluation metrics and he network training of the proposed MSCNet). Then, the performance of MSCNet is evaluated through experiments and by comparing it with other baseline methods. Furthermore, the ablation

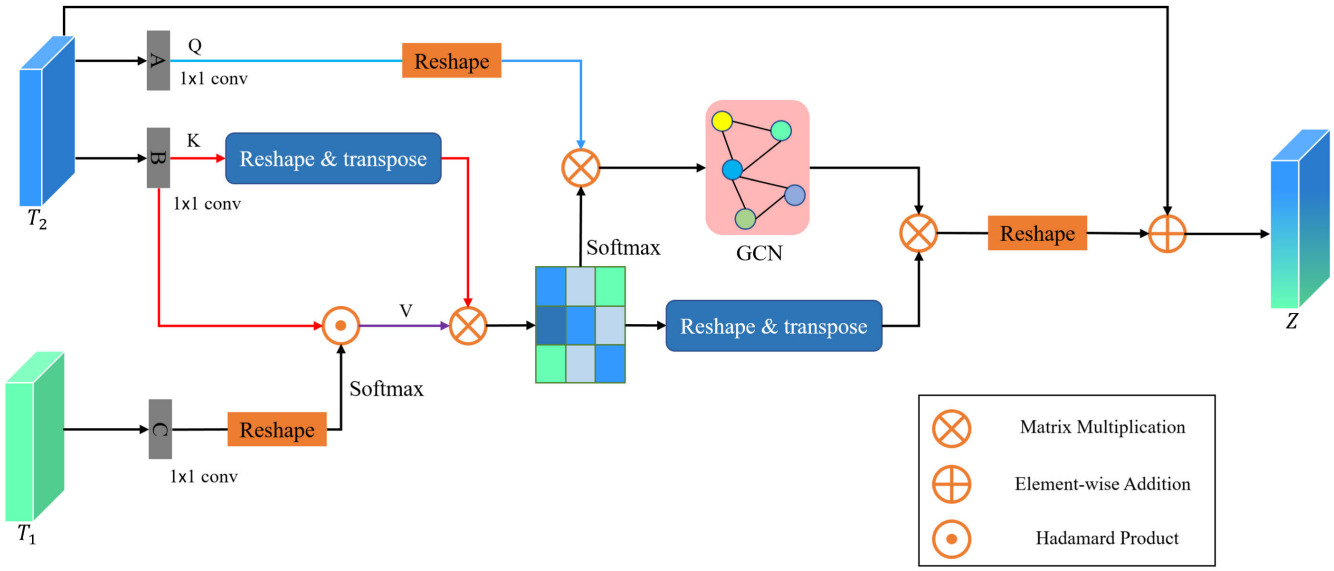


FIGURE 4. Structure of the feature aggregation module.

experiments conducted to determine the effectiveness of the three modules of MSCNet, namely the TEM, CFM, and FAM, are described. Finally, the experimental results are summarized and analyzed.

A. IMPLEMENTATION DETAILS

To make the network achieve excellent results in the actual detection task of bridge cracks, we study the experimental platform of network operation, the data sample size required for training, the parameter setting of the learning rate, and the gradient descent method.

1) DATASET AND COMPUTER ENVIRONMENT

The occurrence of cracks on bridge structures is random to a certain extent. The types and forms of cracks are diverse, leading to problems such as the original data samples having partial information and low credibility as well as lacking balance between different classes and data in the unique standard environment. In the object detection field, the amount of image data affects the detection accuracy of the network. According to the conclusion of [40], we adopted the method of homogeneous sample fusion to expand the data according to the visual semantic features of the color, shape, size, spatial location, and other aspects of the crack image.

In this study, two crack datasets were used as samples: the SDNET dataset [41] and CCIC dataset [42]. The SDNET dataset contained more than 56,000 images of walls, roads, and bridge surfaces, which were classified into cracked and non-cracked images. The CCIC dataset contained 40,000 crack and non-cracked images of crack and non-cracked images. The neural network training needs to invest many data samples. We analyze the data distribution characteristics and sample label types of SDNET dataset and CCIC dataset and combine the data samples of similar labels in the two datasets into one dataset. To better train the network, we adopted 240×240 fixed-size windows slide without overlap on the crack image and selected 16,000 images from the SDNET dataset and 14,000 images from the CCIC dataset, combining them into a large crack dataset called the WCD dataset. The samples in WCD data set are divided into training set, verification set and test set according to the ratio of 6:2:2. Specifically, 18,000 images in the training set were used for training the network, 6,000 images in the validation set were used to optimize the network weight parameters, and 6,000 images in the test set were used to evaluate the detection performance of the network.

Bridge crack detection is typically performed in the field environment, which is difficult to achieve through a large workstation in the laboratory. The algorithm must generally run on a small-scale computing platform and achieve a real-time processing effect for the algorithm to be applied effectively. Therefore, in this study, we selected a Dell laptop as the platform for executing the algorithm; the specific parameters of the platform are listed in Table 1.

TABLE 1. Specific index parameters of the platform.

Hardware/Software	Specification/Parameters/Version
CPU	Intel Core i5 8 Generation
GPU	NVIDIAGeForceGTX1060/6GB
RAM	4GB
Anaconda	3-5.1.0
Python	2.7.5
TensorFlow	1.10

2) NETWORK TRAINING

We compared three common gradient descent methods in object detection, namely the batch gradient descent (BGD),

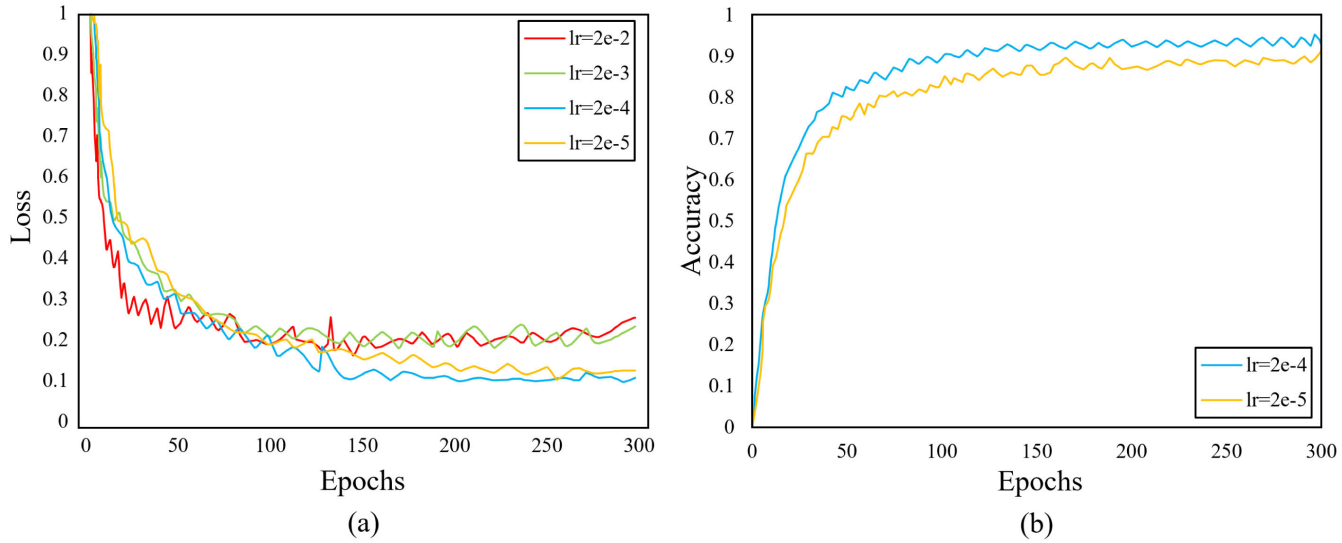


FIGURE 5. Curves of (a) training loss and (b) accuracy with different learning rates.

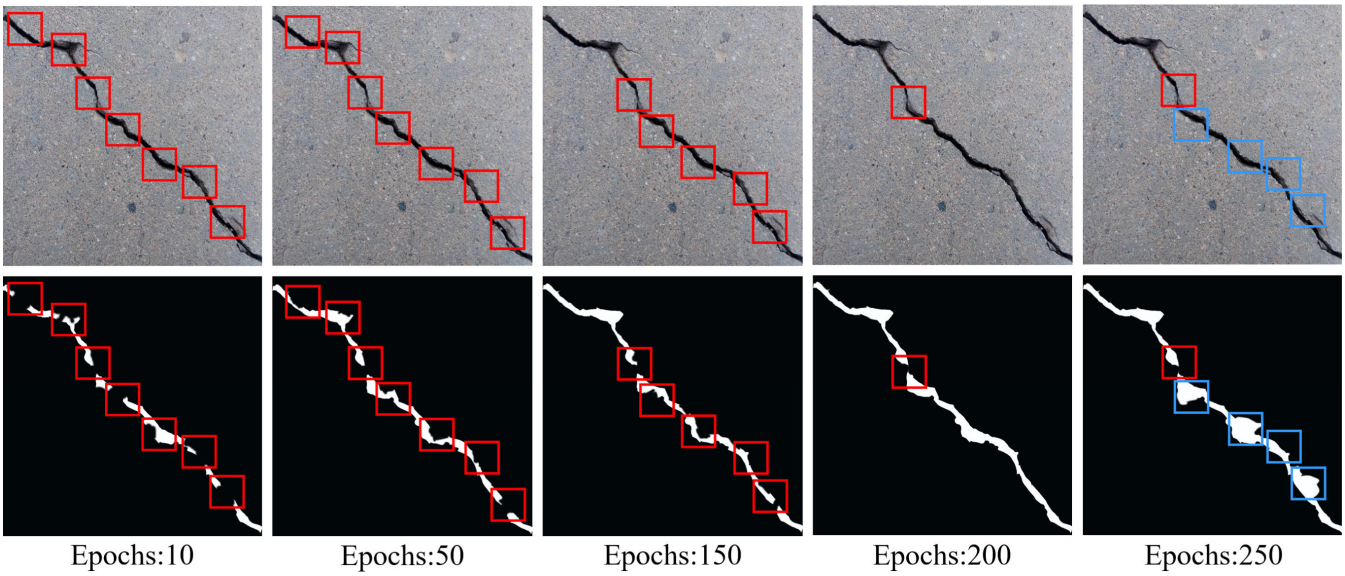


FIGURE 6. Comparison of output results at different epochs; the red boxes locate the missing detection parts of the detection results and the blue boxes locate the false detection parts of the detection results.

stochastic gradient descent (SGD), and mini-batch gradient descent (MBGD). In the BGD process, the training speed was too low to meet the timeliness requirement of the detection task. Meanwhile, although the SGD selects fewer samples in each iteration, which improves the update speed of each round of parameters. The gradient update direction cannot consider other samples, so the accuracy is low. Hence, we adopted the MBGD with a batch size of 16 and a weight decay of 0.0001 to balance the speed and accuracy.

The learning rate is one of the critical parameters in network training. An unreasonable learning rate will lead to gradient explosion or gradient disappearance of the network, resulting in incomplete training. We compared the training loss with different learning rates. As Fig. 5a shows, when the learning rate is $2e^{-4}$ or $2e^{-5}$, the curve decreases faster and the loss value in the steady state is lower. As Fig. 5b shows, when the learning rate is $2e^{-5}$, the training accuracy of the network reaches the stable state fastest and the accuracy value is higher. Therefore, we adopted $2e^{-5}$ as the initial learning rate. To intuitively understand the process of network training, we visually display the detection results of a sample in the training process.

During network training, we extracted the network and visually displayed the detection results of a sample under different epochs. An epoch means that all training samples are sent into the network and complete forwarding calculation

TABLE 2. Qualitative performance results of different networks on the WCD dataset.

Network	Evaluation metrics				
	Precision	Recall	F1-score	Accuracy	FPS
FPHBN [26]	0.849	0.833	0.841	0.842	12.8
DeepCrack [38]	0.851	0.855	0.853	0.847	11.3
CliqueNet [25]	0.854	0.861	0.857	0.849	32.4
SSENets [39]	0.873	0.868	0.871	0.867	17.6
LDCC-Net [40]	0.885	0.881	0.882	0.883	38.1
DFANet [24]	0.897	0.893	0.891	0.887	25.5
MSCNet (ours)	0.935	0.942	0.938	0.927	63.8

and back propagation. With the increase of the number of epochs, the number of weight update iterations increases, and the network's performance in the article changes from the initial non-fitting state to the optimal fitting state. Fig. 6 shows the detection effect of a data sample under different epochs during network training. The first row shows the original image in the data set, and the second row shows the detection results of different epochs. When the number of iterations is small (10 and 50 epochs), the network is in the state of underfitting and the detection effect is poor, both resulting in seven missing detection parts in the results. As the number of iterations increases to 150 epochs, the detection effect of the network gradually improves, the detection accuracy of the network is improved to a certain extent, and the missing detection parts are reduced to 5. When the network is in the state of under-fitting, the network's learning ability is insufficient, unable to learn the general law of target characteristics and accurately identify objects. Hence, there is a situation of missed detection. When the number of iterations reaches 200 epochs, the detection accuracy of the network is obviously enhanced. However, due to the small crack size, one part is still missing. When the number of iterations increases from 200 to 250 epochs, When the number of weight learning iterations is too many and the network is in the over-fitting state, the noise in the training sample and the unrepresentative features in the training sample are fitted. Currently, the network is straightforward to identify the non-target features as the target features mistakenly, so there is a false detection. This indicates that the network tends to over-fit, and the non-crack region is marked as the object region.

B. EVALUATION INDICATORS

While image classification only considers the accuracy and recall rate, object detection must classify and identify the object as well as accurately locate the object position. Therefore, the performance of the algorithm should be evaluated comprehensively considering many aspects. The following evaluation indexes were used to accurately assess the predicted results of the model: the precision rate (Pr), which measures how many of the samples that the model judges to be positive are positive samples; the recall rate (Rr), which represents the proportion of true positive samples that are predicted to be correct; accuracy (Acc), which predicts the exact ratio of positive and negative samples; F1-score (F1), which comprehensively considers the output results of the precision rate and recall rate; and IoU. These indexes are defined as follows:

$$\mathrm{Pr} = \frac{(TP)}{(TP + FP)} \tag{17}$$

$$\mathrm{Rr} = \frac{(TP)}{(TP + FN)} \tag{18}$$

$$\mathrm{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$\mathrm{F1} = 2\frac{(\mathrm{Pr} \times \mathrm{Rr})}{(\mathrm{Pr} + \mathrm{Rr})} \tag{20}$$

$$\mathrm{IoU} = \frac{area\left(B_{\mathrm{det}} \cap B_{gt}\right)}{area\left(B_{\mathrm{det}} \cup B_{gt}\right)} \tag{21}$$

where TP, TN, FN, and FP are the indicators in the confusion matrix; TP represents true positive (a positive sample predicted to be positive), TN indicates true negative (a negative sample predicted to be negative), FN represents false negative (a positive sample predicted to be negative), and FP indicates false positive (a negative sample predicted to be positive). B_{det} represents the size of the detection box, B_{gt} represents the size of the calibration box of the detection object, $area(B_{\mathrm{det}} \cap B_{gt})$ represents the overlapped area of the two boxes, and $\mathrm{area}(\mathrm{B}_{\mathrm{det}} \cup \mathrm{B}_{\mathrm{gt}})$ represents the total area of the combined two boxes.

The higher the correlation, the higher is the IoU [3]. In the process of model training, different thresholds of the IoU were set to measure the detection accuracy of the model. Fig. 7 shows the P-R curves under different IoU thresholds. By analyzing the results, we know that when the threshold of IoU is 0.5, the network's performance reaches the best after early training.

C. COMPARISON AND RESULTS

Considering the limited computing capacity of the bridge crack detection platform, six crack detection networks, namely FPHBN [27], DeepCrack [31], CliqueNet [26], SSENets [32], LDCC-Net [33], and DFANet [25], were selected for comparison. All the networks were trained and tested on the same experiment setup and dataset. The quantitative analysis results of each of the methods are shown in Table 2, and some results after visualization are shown in Figs. 8 and 9.

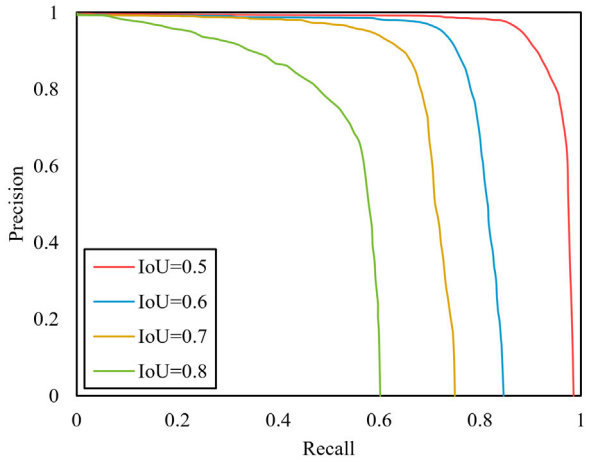


FIGURE 7. P-R Curves under different thresholds of IoU.

The five-evaluation metrics discussed in Section 4.1 were used to quantitatively compare the performance of the various networks. Table 2 indicates that FPHBN and DeepCrack were inferior to other algorithms in detection accuracy and detection speed. Although CliqueNet, SSENets, LDCC-Net, and DFANet performed better than FPHBN and DeepCrack, balancing the detection accuracy and detection speed was difficult for these algorithms. By contrast, the proposed MSC-Net achieved superior performance both in detection precision and detection speed, with 93.8% F1-score and 63.8 FPS, respectively.

For the in-depth analysis of the performance of the algorithm on difficult samples, we selected four types of crack samples under different backgrounds for depth visual feature analysis. Fig. 8 shows the detection results of the algorithm under five backgrounds: a rough cement pavement (row 1), a region with cluster interference (row 2), a stable background (row 3), a rough zebra crossing (row 4) and a rough lane line (row 5). The detection results of crack samples in the rough cement pavement background (row 1) indicate the weak performance of FPHBN and DeepCrack in noise reduction, which undoubtedly poses a challenge for later classification processing. Although other methods showed improved noise reduction ability to a certain extent, they are still not ideal. Only from the aspect of key information extraction, DFANet and MSCNet have obvious technical advantages in combating noise consistently. The detection results of the input images with obvious cluster interference regions (row 2) indicate an ineffective performance of our method in successfully eliminating the non-crack areas. However, compared with other methods, MSCNet showed obvious visual effect improvement in noise reduction, which implies that for object detection, achieving anti-interference and anti-noise capability is a serious challenge. In case of the input image with a relatively clear object in a stable background (row 3), although there were shadows of two cables on it, the numerical gradient of the interference in the image data was not large; therefore, each of the networks resulted in a good processing effect. In case of the input image with a rough zebra crossing (row 4), owing to a strong visual discrimination from the background, the feature optimization results were not perfect irrespective of the network. Different from the interference of rough zebra crossing (row 4), the area of lane crossing interference is more prominent (row 5), and the crack shape of the selected sample is more complex, which poses a challenge to the network. For the image with rough input lane lines (row 5), the ability of the network to extract crack features is reduced due to the influence of solid background interference, and the detection effect is not ideal. From the above-mentioned processing results of the five complex samples, the proposed network achieved relatively stable optimization results in strengthening the object and weakening the interference. However, in general, in the object detection under complex background, the interference information can only be weakened and cannot be eliminated. This implies that the visual distinguishability between the object and the background plays an important role in the detection results.

In fact, although the structural characteristics of bridge cracks cannot be identified as effectively as in case of object detection with edge information as key information, such as fingerprint recognition [43] and iris recognition [44], bridge cracks still have their structural characteristics such as poor smoothness and weak consistency of the crack scale. Because describing these visual features with randomness in a complete sense is difficult, we consider that these features comprise some consistency that cannot be described but can be well explored using deep learning theory. We extracted five common crack structures: the longitudinal crack, transverse crack, cross crack, network crack and cross network crack. We used depth vision feature analysis to determine the ability of the networks to extract texture information such as crack shape and scale as well as the ability to resist breakpoints in the extraction process. Fig. 9 visually shows the detection results of the five crack structures by different networks. The geometry of crack samples in the first and second columns in the Fig. 9 are relatively simple, and all networks could effectively extract continuous and complete texture information. By contrast, the structural complexity of the input images in the third, fourth and fifth rows is greater than that of the first two columns of samples, and all networks were affected to a certain extent. This phenomenon was particularly prominent when the networks detected mesh cracks. Comparison and analysis of the results of the five rows indicate that DFANet and MSCNet maintained good texture information extraction ability and strong anti-breakpoint ability in the process of crack feature extraction.

The results presented in Table 2, Fig. 8, and Fig. 9 confirm that DFANet and MSCNet have better processing results in terms of the visual expression of depth features. From the perspective of the processing mechanism

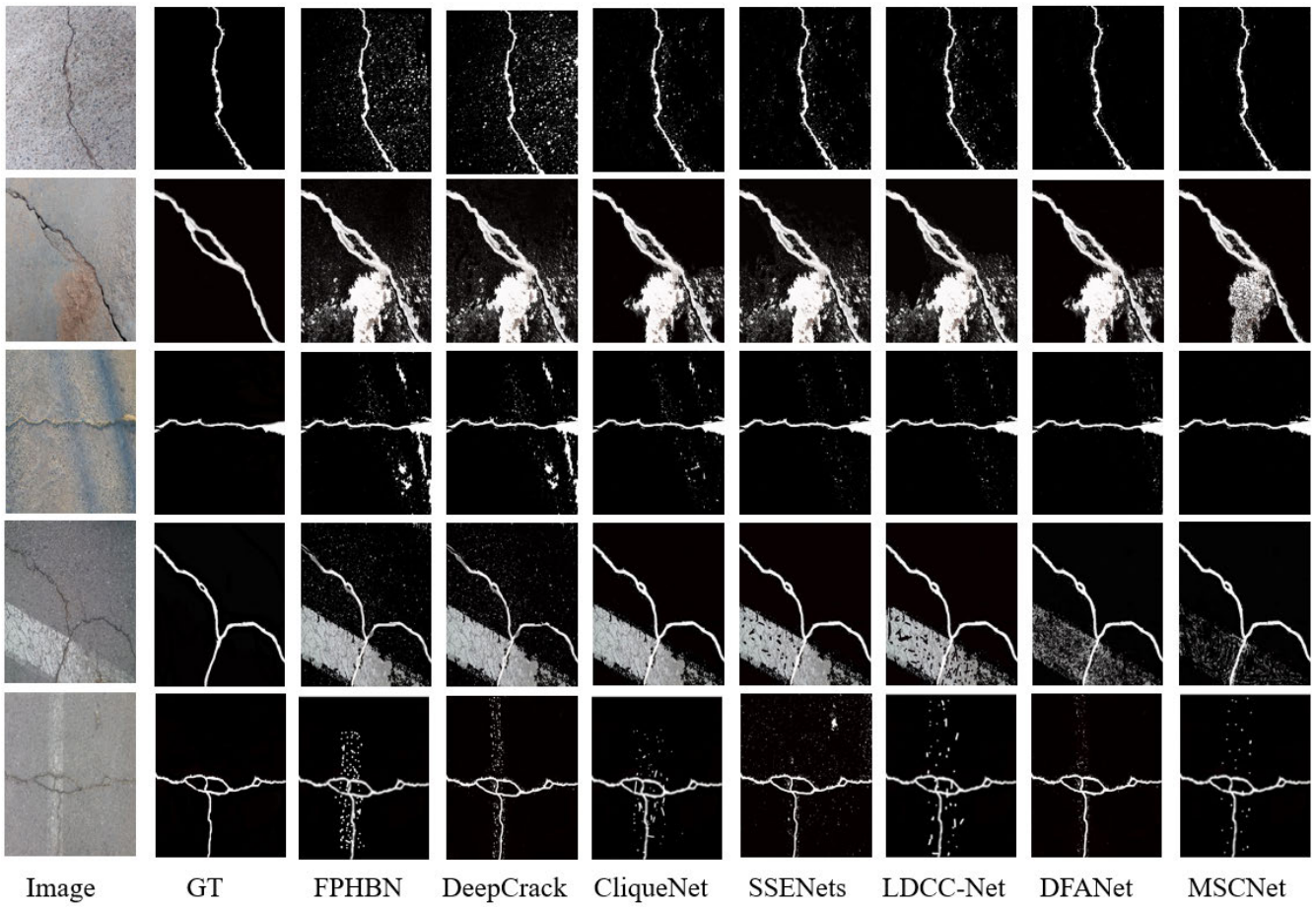


FIGURE 8. Comparison of the visualization of detection results of various networks under different environmental situations.

of the two object detection algorithms, these results confirm their better superior data support for classification processing, leading to higher accuracy of detection results. However, the reasoning speed of the proposed MSCNet is superior.

D. ABLATION EXPERIMENT

In our crack detection framework, the performance of the three optimization modules determines the final pixel-level crack detection. We further explored how each optimization module affects the performance of crack detection by removing or replacing each of them. Three experiments were designed to verify the effectiveness of the TEM, CFM, and FAM.

To prove the effectiveness of the TEM, an experiment was conducted on the test set, and the intuitive results are shown in Fig. 10. The confusion matrix [45] in Fig. 10a shows that MSCNet without the TEM misclassified 312 (out of 1500) crack regions of interest (ROIs) as background ROIs and 522 (out of 4500) background ROIs as crack ROIs. By contrast, as shown in Fig. 10b, using MSCNet with the TEM, the instances of misclassification sharply decreased, with only 35 (out of 1500) crack ROIs and 78 (out of 4500) background ROIs that were misclassified. These results demonstrate that the introduction of the TEM enhances the network's extraction efficiency of low-level features and ability to resist environmental noise.

To demonstrate the effectiveness of the CFM, we conducted experiments to compare the performance of different networks, including MSCNet without and with the CFM, in terms of mAP and FPS on the experiment platform. As shown in Fig. 11, MSCNet with the CFM achieved the fastest and highest accuracy results, achieving 63.8 FPS and 93.5%, respectively. By contrast, the performance of MSCNet without the CFM sharply decreased both in terms of speed and accuracy, with the values being 38.1 FPS and 88.6%, respectively. These results show that CFM fusion method avoids the direct fusion of low-level features and high-level features, and uses cascade fusion to refine the weak semantic information of high-level features. The CFM reduces the network's consumption of computing resources. Achieving a balance between accuracy and computing resource consumption is beneficial to improve the accuracy and speed of the network.

To examine the effectiveness of the FAM, we conducted experiments by replacing it from the standard MSCNet with

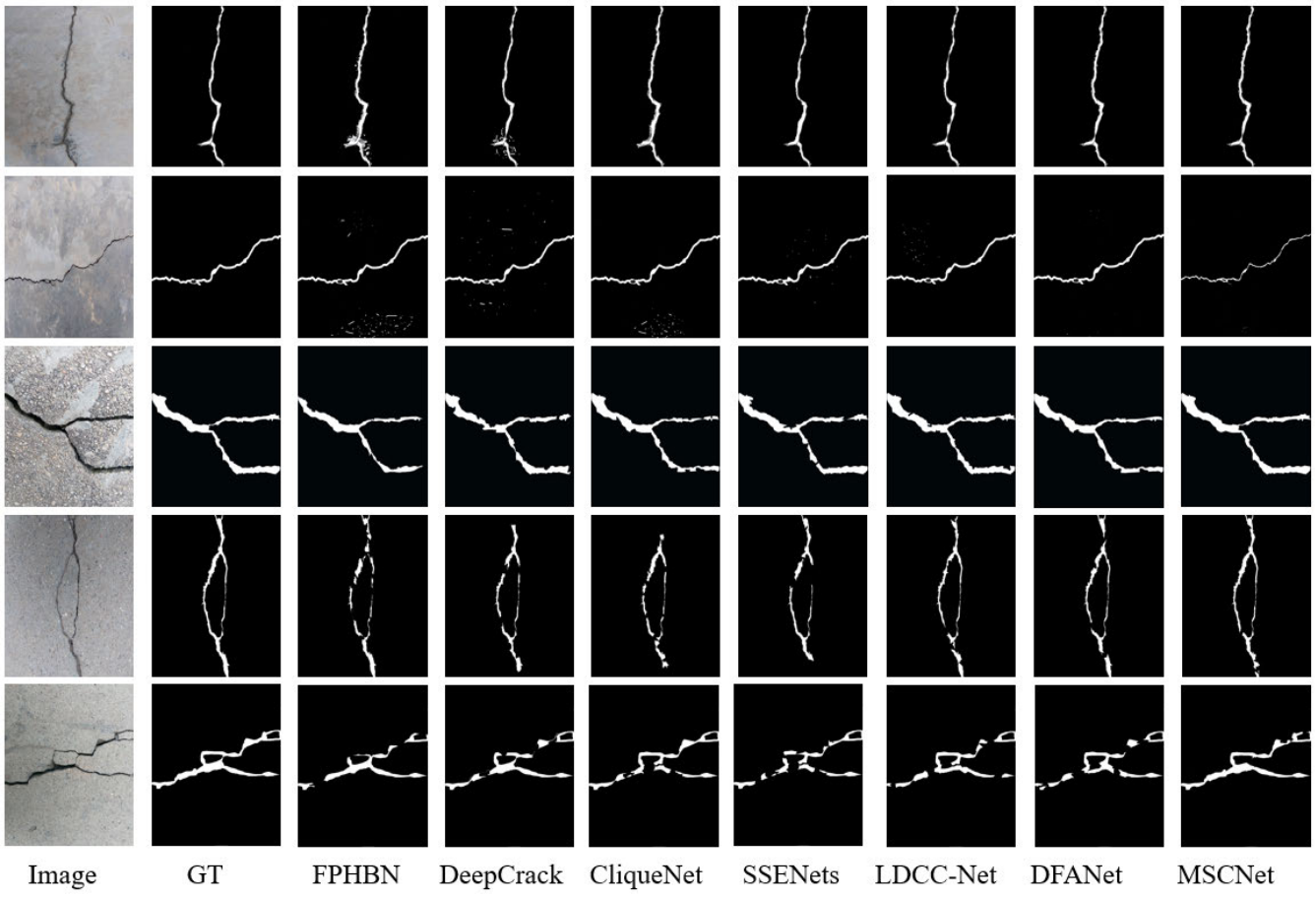


FIGURE 9. Comparison of the visualization of detection results of various networks for different types of cracks.

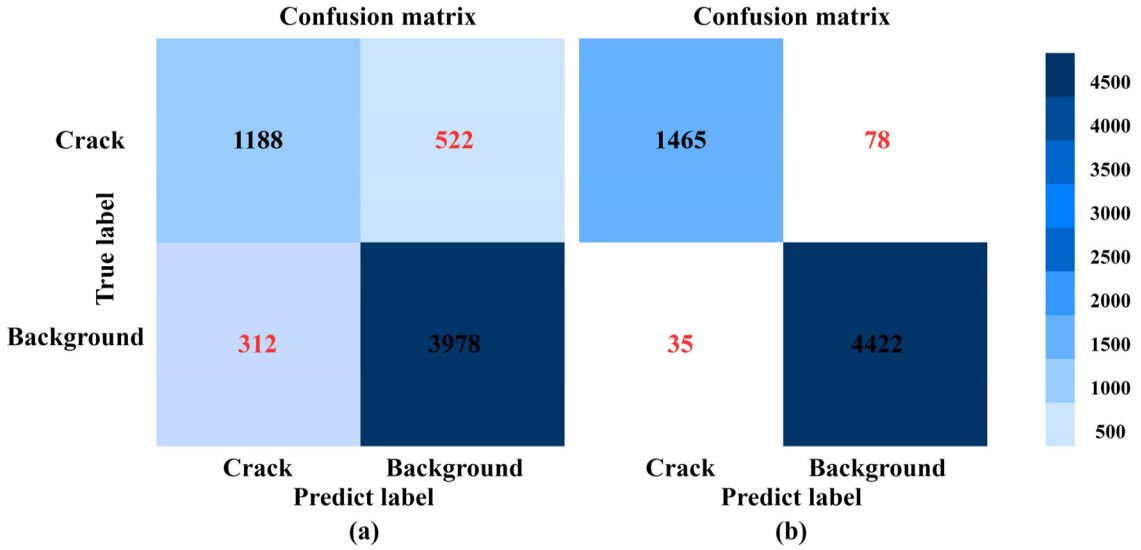


FIGURE 10. Confusion matrix constructed based on the results of (a) the MSCNet without the TEM and (b) the MSCNet with the TEM.

an element-wise addition operation, which is denoted as MSCNet (w/o FAM). The results shown in Fig. 12 indicate that the FAM, which is equipped with non-local and graph convolution layers to mine local pixels and global semantic cues from crack areas, incorporates detailed appearance features of global attention into high-level semantic features

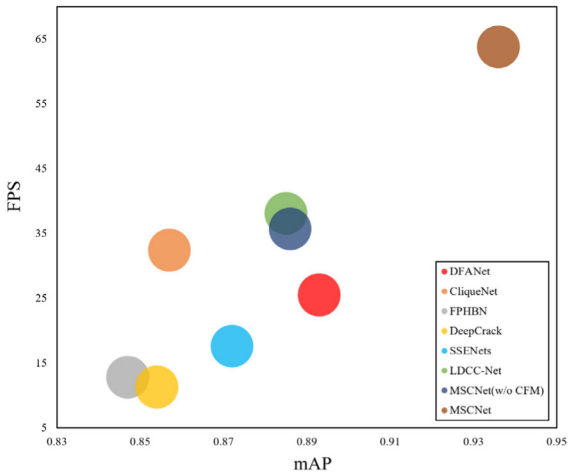


FIGURE 11. Comparison results of detection accuracy and reasoning speed of different networks.

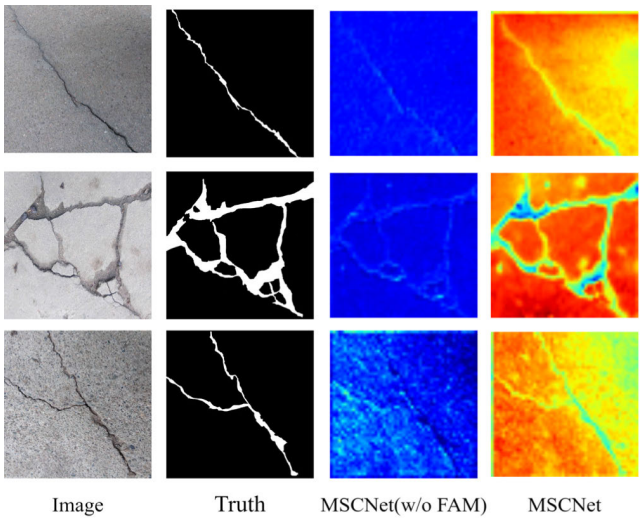


FIGURE 12. Comparison of visualization of the results of different networks to examine the effectiveness of the FAM.

and thus effectively aggregates high-level and low-level features.

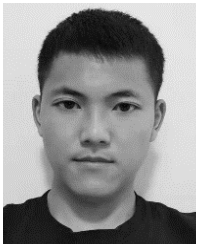
V. CONCLUSION

In this study, we proposed a novel and effective crack detection framework, called MSCNet, which uses a Res2Net backbone to extract features. The proposed network incorporates three modules, namely the TEM, CFM, and FAM, to effectively extract low-level features, reasonably balance accuracy and computational resource consumption, and fully integrate high- and low-level features to obtain the final output. A series of ablation experiments verified the effectiveness of these modules. Furthermore, extensive experiments demonstrated that MSCNet outperformed other state-of-the-art crack detection networks considered for comparison on the same dataset. Specifically, MSCNet achieved 93.8% F1-score and 63.8 FPS, which exceed the accuracy and speed of other networks significantly. Although the proposed MSCNet method can obtain more satisfactory performance than other methods, compared with the computing power of current portable computing terminals for bridge crack detection, the complexity of neural network structure and the redundancy of feature mapping is unbearable. We will focus on these issues in future research.

REFERENCES

[1] T. Gavrilov and G. Kolesnikov, "Conditions modelling of low-temperature cracks existence in the road upper layer," in *Proc. J. Phys., Conf.*, 2020, vol. 1614, Art. no. 012100.
[2] M. Zhou, W. Lu, J. Song, and G. C. Lee, "Application of ultra-high performance concrete in bridge engineering," *Construct. Building Mater.*, vol. 186, pp. 1256–1267, Oct. 2018.
[3] C. Zhang, C. C. Chang, and M. Jamshidi, "Concrete bridge surface damage detection using a single-stage detector," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 4, pp. 389–409, 2020.
[4] M. Koziarski 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, 2017.
[5] L. Zhang, J. Shen, and B. Zhu, "A research on an improved Unet-based concrete crack detection algorithm," *Struct. Health Monit.*, vol. 20, no. 4, pp. 1864–1879, 2012.
[6] Y. Ren, J. Huang, Z. Hong, W. Lu, J. Yin, L. Zou, and X. Shen, "Image-based concrete crack detection in tunnels using deep fully convolutional networks," *Construct. Building Mater.*, vol. 234, Feb. 2020, Art. no. 117367.
[7] M. Sharma, W. Anotaipaiboon, and K. Chaiyasarn, "Concrete crack detection using the integration of convolutional neural network and support vector machine," *Sci. Technol. Asia*, vol. 23, no. 2, pp. 19–28, 2018.
[8] A. M. Talab, Z. Huang, F. Xi, and H. Liu, "Detection crack in image using Otsu method and multiple filtering in image processing techniques," *Optik*, vol. 127, no. 3, pp. 1030–1033, 2016.
[9] B. O. Santos, J. Valença, and E. Júlio, "Detection of cracks on concrete surfaces by hyperspectral image processing," *Proc. SPIE*, vol. 10334, Jun. 2017, Art. no. 1033407.
[10] P. Prasanna, K. J. Dana, and N. Gucunski, "Automated crack detection on concrete bridges," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 591–599, Apr. 2016.
[11] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[12] D. T. Nguyen, T. N. Nguyen, H. Kim, and H. J. Lee, "A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1861–1873, Aug. 2019.
[13] F. Fang, L. Li, H. Zhu, and J. H. Lim, "Combining faster R-CNN and model-driven clustering for elongated object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 2052–2065, 2019.
[14] J. Li, H. C. Wong, S.-L. Lo, and Y. Xin, "Multiple object detection by a deformable part-based model and an R-CNN," *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 288–292, Feb. 2018.
[15] S. Jiang and J. Zhang, "Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 6, pp. 549–564, Jun. 2020.
[16] 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.
[17] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "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, 2018.
[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[19] 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.

[20] S. Chanda, G. Bu, H. Guan, J. Jo, U. Pal, and Y. C. Loo, "Automatic bridge crack detection-a texture analysis-based approach," in *Proc. IAPR Workshop Artif. Neural Netw. Pattern Recognit. (ANNPR)*, Montreal, QC, Canada, 2014, pp. 193–203.
[21] 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.
[22] 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, pp. 3434–3445, Dec. 2016.
[23] H. Zhang, J. Tan, L. Liu, Q. M. J. Wu, Y. Wang, and L. Jie, "Automatic crack inspection for concrete bridge bottom surfaces based on machine vision," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 4938–4943.
[24] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes, "Deeper networks for pavement crack detection," in *Proc. 34th Int. Symp. Autom. Robot. Construct. (IAARC)*, Taipei, Taiwan, Jul. 2017, pp. 479–485.
[25] W. Qiao, Q. Liu, X. Wu, B. Ma, and G. Li, "Automatic pixel-level pavement crack recognition using a deep feature aggregation segmentation network with a scSE attention mechanism module," *Sensors*, vol. 21, no. 9, p. 2902, Apr. 2021.
[26] G. Li, B. Ma, S. He, X. Ren, and Q. Liu, "Automatic tunnel crack detection based on U-Net and a convolutional neural network with alternately updated clique," *Sensors*, vol. 20, no. 3, p. 717, Jan. 2020.
[27] F. Yang, L. Zhang, S. Yu, D. V. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1525–1535, Apr. 2020.
[28] Q. Zhou, Z. Qu, and C. Cao, "Mixed pooling and richer attention feature fusion for crack detection," *Pattern Recognit. Lett.*, vol. 145, pp. 96–102, May 2021.
[29] R. Fan, M. J. Bocus, Y. Zhu, J. Jiao, L. Wang, F. Ma, S. Cheng, and M. Liu, "Road crack detection using deep convolutional neural network and adaptive thresholding," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 474–479, doi: 10.1109/IVS.2019.8814000.
[30] T. Ahmad, Y. Ma, M. Yahya, B. Ahmad, S. Nazir, and A. U. Haq, "Object detection through modified Yolo neural network," *Sci. Program.*, vol. 2020, pp. 1–10, Jun. 2020, doi: 10.1155/2020/8403262.
[31] 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.
[32] H. Li, H. Xu, X. Tian, Y. Wang, H. Cai, K. Cui, and X. Chen, "Bridge crack detection based on SSENets," *Appl. Sci.*, vol. 10, no. 12, p. 4230, Jun. 2020.
[33] J. Kim, S. Shim, Y. Cha, and G.-C. Cho, "Lightweight pixel-wise segmentation for efficient concrete crack detection using hierarchical convolutional neural network," *Smart Mater. Struct.*, vol. 30, no. 4, Apr. 2021, Art. no. 045023.
[34] S. H. Gao, M. M. Cheng, and K. Zhao, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.
[35] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3907–3916.
[36] X. Liu, L. Li, F. Liu, B. Hou, S. Yang, and L. Jiao, "GAFnet: Group attention fusion network for PAN and MS image high-resolution classification," *IEEE Trans. Cybern.*, early access, Mar. 30, 2021, doi: 10.1109/TCYB.2021.3064571.
[37] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6861–6871.
[38] E. Jörgensen, C. Zach, and F. Kahl, "Monocular 3D object detection and box fitting trained end-to-end using intersection-over-union loss," 2019, *arXiv:1906.08070*.
[39] R. LaLonde and U. Bagci, "Capsules for object segmentation," 2018, *arXiv:1804.04241*.
[40] Y. Zhang, D. Sidibé, O. Morel, and F. Mériaudeau, "Deep multimodal fusion for semantic image segmentation: A survey," *Image Vis. Comput.*, vol. 105, Jan. 2021, Art. no. 104042.
[41] S. Dorafshan, R. J. Thomas, and M. Maguire, "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks," *Data brief*, vol. 21, pp. 1664–1668, Dec. 2018.
[42] F. O. Çağlar and R. Özgenel, "Concrete crack images for classification," *Mendeley Data*, vol. 2, 2019.
[43] J. Priesnitz, R. Huesmann, C. Rathgeb, N. Buchmann, and C. Busch, "Mobile contactless fingerprint recognition: Implementation, performance and usability aspects," *Sensors*, vol. 22, no. 3, p. 792, Jan. 2022, doi: 10.3390/s22030792.
[44] C. Wang, J. Muhammad, Y. Wang, Z. He, and Z. Sun, "Towards complete and accurate iris segmentation using deep multi-task attention network for non-cooperative iris recognition," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2944–2959, 2020, doi: 10.1109/TIFS.2020.2980791.
[45] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, Oct. 2019.

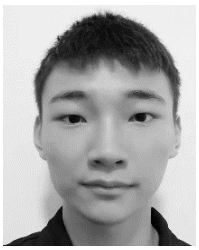


GUANLIN LU is currently pursuing the master's degree with the College of Field Engineering, Army Engineering University of PLA. His research interest includes deep learning.

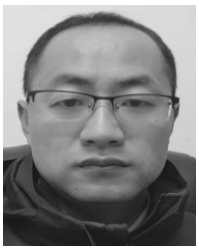
XIAOHUI HE was born in 1975. He received the Ph.D. degree from the Army Engineering University of PLA, China. He is currently a Professor at the Army Engineering University of PLA. His research interests include mechatronics, deep learning, and computer vision.

QIANG WANG was born in 1964. He received the Ph.D. degree from the Army Engineering University of PLA, China. He is currently a Professor at the Army Engineering University of PLA. His research interests include mechatronics, deep learning, and computer vision.

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



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



XIAOKANG ZHAO is currently pursuing the master's degree with the College of Field Engineering, Army Engineering University of PLA. His research interests include military operations research and deep learning.