DOI: 10.1111/mice.13128

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Encoder–decoder with pyramid region attention for pixel-level pavement crack recognition

Hui Yao[1] | Yanhao Liu[1] | Haotian Lv[2] | Ju Huyan[3,4] | Zhanping You[5] | Yue Hou[6]

[1]Beijing Key Laboratory of Traffic Engineering, College of Metropolitan Transportation, Faculty of Architecture, Civil and Transportation Engineering, Beijing University of Technology, Beijing, China

[2]School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin, China

[3]College of Future Transportation, Chang'an University, Xi'an, China

[4]School of Transportation, Southeast University, Nanjing, China

[5]Department of Civil and Environmental Engineering, Michigan Technological University, Houghton, Michigan, USA

[6]Department of Civil Engineering, Swansea University, Swansea, UK

Correspondence
Yue Hou, Department of Civil Engineering, Swansea University, Swansea, SA2 8PP, UK.
Email: yue.hou@swansea.ac.uk

Funding information
Hunan Expressway Group Co. Ltd and the Hunan Department of Transportation, Grant/Award Number: 202152; National Natural Science Foundation of China, Grant/Award Numbers: 51778038, 52108403; China National Key R&D Program, Grant/Award Numbers: 2021YFB2600600, 2021YFB2600604, 2018YFB1600202

Abstract

Timely and accurate extraction of pavement crack information is crucial to maintain service conditions and structural safety for infrastructures and reduce further road maintenance costs. Currently, deep learning techniques for automated pavement crack detection are far superior to traditional manual approaches in both speed and accuracy. However, existing deep learning models may easily lose crack details when processing images containing complex background textures or other noises. Although many studies have alleviated this challenge by introducing attention mechanisms, especially the non-local (NL) block, which has the ability to efficiently capture long-range dependencies to facilitate crack pixel capture, the huge computational cost of NL makes the inference time of the model too long, which is not conducive to practical implementation. In this study, a new module, namely, the pyramid region attention module (PRAM), was developed by combining the pyramid pooling module in the pyramid scene parsing network and optimized NL, which can achieve global multi-scale context integration and long-range dependencies capture at a relatively lower computational cost. By applying PRAM to deep skip connections in the modified U-Net, an effective crack segmentation model called CrackResU-Net was developed. The test results on the existing CrackForest dataset showed that CrackResU-Net not only achieved an F1 score of 0.9580 but also took only 25.89 ms to process an image with a resolution of 480 × 320, which had advantages in accuracy and speed, compared with several other state-of-the-art crack segmentation approaches. It was fully demonstrated that this approach could realize automatic fast and high-precision recognition of pavement cracks for engineering purposes.

1 | INTRODUCTION

Asphalt pavements inevitably deteriorate during service due to a variety of internal and external factors, such as repeated traffic loading and severe weather conditions. Cracks can significantly affect road smoothness and driving comfort. If not repaired in time, the road structure may be infiltrated by water through the initial cracks, causing a serious reduction in the service life of the road and resulting in a significant increase in road maintenance costs. Therefore, timely detection of cracks plays an important role for civil engineers. Traditional manual visual inspection relies almost entirely on manual operation, which has not only a high human cost but also the problem of subjective evaluation results and potential traffic safety concerns. As a result, efficient and accurate automatic crack detection methods are urgently needed to enhance the quality of road monitoring.

Methods based on traditional image processing such as threshold-based methods (Q. Li & Liu, 2008; Oliveira & Correia, 2009) and edge detection (Ayenu-Prah & Attoh-Okine, 2008; Huili Zhao et al., 2010) have been utilized for detecting pavement cracks in previous studies. Although these methods are easy to implement, the performance is greatly affected by the external environment, such as unsatisfactory detection results for low-quality pavement images containing shadows, rain, or other noise. As a result, machine learning methods, which have had great success in the field of computer vision, were subsequently introduced for pavement crack recognition, such as support vector machines (N. Li et al., 2009) and artificial neural networks (Cheng et al., 2001; Xu et al., 2008). Compared with image processing methods, machine learning-based methods are more accurate and detect less noise. However, the complex feature construction steps limited its applicability. Moreover, due to the limited computing power of computers at that time, the fewer neural network layers were insufficient to fully express the complex pavement features, which severely limited the improvement in detection accuracy.

As artificial intelligence is rapidly developing, the construction and application of deep learning models have been preliminarily realized, which means more neural network layers could be used for feature extraction and integration. Deep learning is a powerful technology that not only has been developed for collision avoidance perception of mobile robots (Macias-Garcia et al., 2021), classification of motor imagery electroencephalography (EEG) signals (Hassanpour et al., 2019), and EEG-based emotion recognition (Olamat et al., 2022) but also has good applications in solving civil engineering problems, such as forecasting earthquakes (Rafiei & Adeli, 2017), estimating the sale prices of real estate units (Rafiei & Adeli, 2016), estimating concrete properties (Rafiei et al., 2017), and evaluating construction costs (Rafiei & Adeli, 2018). In 2012, the great success of AlexNet (Krizhevsky et al., 2012) in image classification attracted the interest of researchers engaged in automated pavement crack detection research. Among the various methods, image-level classification methods (Hou et al., 2021; Pauly et al., 2017) and object detection methods (Maeda et al., 2018; Yao et al., 2022) can obtain only the rough location information of pavement cracks, which could not meet the needs of pavement condition assessment. Therefore, pixel-level classification methods that can provide accurate geometric features of cracks have become the mainstream direction. Most of the current networks designed for pixel segmentation were based on the encoder–decoder model, in which the encoder is for feature extraction and the decoder up-samples the encoder output to the original image size and outputs the segmentation results. Bang et al. (2019) developed a deep convolutional encoder–decoder network and successfully achieved pixel-level segmentation of road cracks in black box images. Jenkins et al. (2018) built a U-Net-based (Ronneberger et al., 2015) pavement crack segmentation network, the advantage of which is that its skip connections could allow information fusion between its encoder and decoder, thereby alleviating the problem of insufficient spatial information in the decoder. However, the model accuracy remains to be improved for pavement images that normally contain various noises and complex background textures.

Providing models with sufficient multi-scale context information is a way to improve accuracy. Zou et al. (2018) and Yang et al. (2019) improved the model's ability to distinguish cracks from the background by fusing feature maps from multiple scales. However, the computational complexity may be high when dealing with large-size images. Comparatively, atrous spatial pyramid pooling (ASPP; L. -C. Chen et al., 2018), which uses atrous convolutions with different dilated rates to capture and fuse features from different receptive fields, is more lightweight. Song et al. (2020) established a pavement crack segmentation network named CrackSeg, which utilized a multi-scale dilated convolution module to learn rich semantic information and fused it with shallow spatial high-resolution features to obtain crack feature details. Ye et al. (2023) adopted a structure similar to ASPP at the top of their network to integrate multi-scale information, which is beneficial for extracting abstract features and improving segmentation accuracy. Another lightweight module for obtaining multi-scale context information is the pyramid pooling module (PPM; Hengshuang Zhao et al., 2017), which performs multi-scale pooling operations on the input feature map to extract the multi-scale context information. In pavement crack detection studies, modules similar to PPM were commonly applied to the deep layers of the network to help understand the semantic structure of the whole scene and improve segmentation performance (Xiang et al., 2020; Zhou et al., 2021).

The attention mechanism is another technique to enhance model precision. In 2018, the squeeze and excitation (SE) module (Hu et al., 2018) was proposed as a tool for enhancing model performance. This module enabled the network model to learn the importance of different feature channels by generating an attention map and reweighting each feature channel of the feature maps. Test results showed that applying the SE module to classification models can improve its performance without significantly increasing the computational cost. Inspired by SE, more spatial or feature channel attention modules have been developed and introduced into pavement crack detection studies. Based on U-Net, Augustauskas and Lipnickas (2020) utilized attention gate (AG) (Oktay et al., 2018) to enhance crack detection performance. Song et al. (2019) captured deep crack details by introducing a feature channel attention module. Xiang et al. (2020) used bottleneck attention module (BAM; Park et al., 2018) in their encoder–decoder model to accurately capture cracks and suppress irrelevant information. Qiao et al. (2021) applied the concurrent spatial and channel squeeze and channel excitation block (Roy et al., 2018) to their model and achieved better crack segmentation precision. However, the feature channel and spatial attention modules described above can capture only local information and cannot model long-range dependencies. As a result, the self-attention mechanism (Vaswani et al., 2017) has attracted the attention of researchers recently. By introducing the idea of self-attention, the non-local (NL) block was proposed by X. Wang et al. (2018) and achieved excellent performance in several computer vision tasks due to its effective capture of long-range dependencies. Inspired by NL, Wan et al. (2021) designed CrackResAttentionNet for pavement crack detection based on the encoder–decoder network, where two self-attention-based attention modules were added after different encoder layers to aggregate long-range context information. Ong et al. (2023) used self-attention to refine each feature pyramid network (FPN) layer so that the deep and shallow layers of the FPN could enhance crack information and reduce noise impact, respectively. However, the large computational cost of the self-attention mechanism severely limits the detection speed.

In this paper, an innovative pyramid region attention module (PRAM) is proposed to capture long-range dependencies and integrate multi-scale context information with a relatively lower computational cost. By applying the PRAM, BAM, auxiliary branch, and spatial attention module (SA) to the modified U-Net embedded with ResNet-34 (He et al., 2016), an effective crack segmentation model named CrackResU-Net was designed. The proposed model was tested on the CrackForest dataset (CFD; Shi et al., 2016), Cracktree200 (Zou et al., 2012), and Crack500 (Yang et al., 2019), and results were discussed to evaluate its effectiveness and robustness. The main purpose of this research was to build an efficient crack segmentation model for on-site testing by civil engineers that can achieve rapid detection while ensuring high accuracy.

2 | METHODOLOGY

2.1 | Network architecture of CrackResU-Net

As a powerful semantic segmentation model, U-Net has achieved success in medicine (Ronneberger et al., 2015). Recently, with the emergence and development of attention mechanisms, the combination of the two has significantly improved its performance (Augustauskas & Lipnickas, 2020). Thus, the proposed CrackResU-Net was based on the modified U-Net embedded with ResNet-34. Figure 1 shows its network architecture. The input image with three channels is input into CrackResU-Net, and the feature extraction is performed by ResNet-34 combined with BAM. As down-sampling proceeds, the image size is reduced to, at most, 1/32. Subsequent up-sampling by the Upsample Block is performed to reduce the feature channel while scaling up the feature map size, which facilitates the fusion with the corresponding encoder feature map. The Upsample Block contains an up-sampling layer and a 3 × 3 kernel size convolution layer. After the fusion operation, an Upconv Block, which consists of two 3 × 3 kernel size convolution layers, is used for feature integration. Then, the up-sampling process is repeated, and finally the Upconvlast Block, containing an up-sampling layer and a 1 × 1 kernel size convolution layer, is passed through to obtain pixel-level binary detection results (crack and non-crack). In particular, the PRAM is applied to the two deep skip connections of CrackResU-Net for capturing long-range dependencies and aggregating multi-scale context information. Meanwhile, the SA is introduced in the two shallow skip connections of CrackResU-Net to enhance the spatial crack edge information for more accurate segmentation results. More details about the important components of CrackResU-Net are introduced in the subsequent sections.

2.1.1 | SA

With down-sampling in the encoder, some spatial information is gradually lost, which leads to insufficient information obtained by the decoder, thus resulting in poor

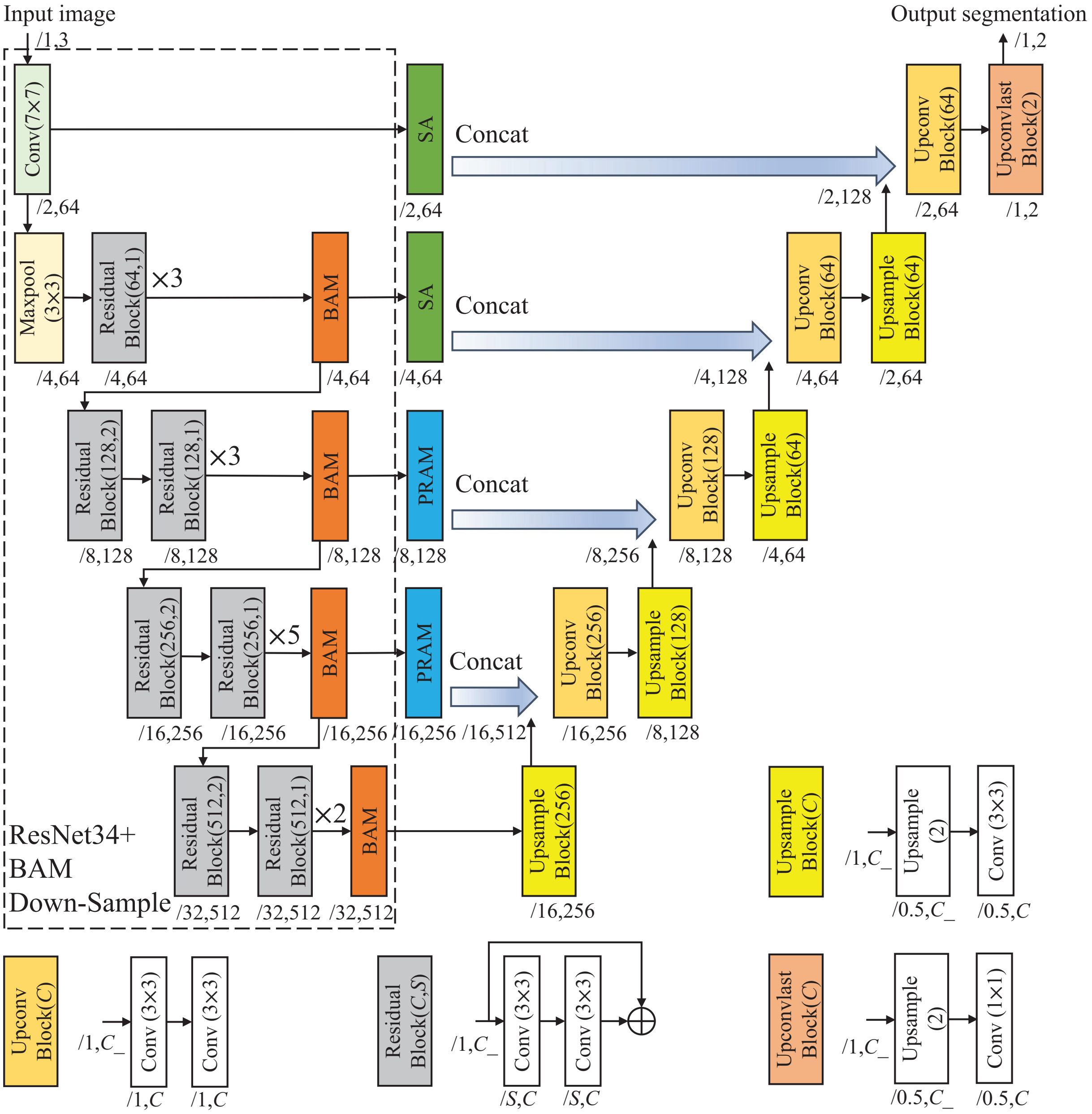

FIGURE 1 Network architecture of CrackResU-Net.

detection of tiny targets or edges. U-Net has designed the skip connection to alleviate this problem. Through skip connections, features at different levels in the encoder could be integrated with the corresponding features in the decoder to complement the missing spatial details. However, simple concatenation does not allow the network to capture important information related to the detection target to guide network training, so several studies have used attention modules to improve skip connections, such as AG (J. Chen & He, 2022) and channel attention block (Hsieh & Tsai, 2021). In this paper, SA was introduced to CrackResU-Net's skip connections to refine the spatial information of cracks. Without attention modules applied, the output feature maps for different encoder layers of CrackResU-Net were visualized using Grad-Cam (Selvaraju et al., 2017) and are shown in Figure 2. It can be seen that features of deep networks have less spatial information than semantic information, so the application of SA for reinforcement may not be significant. In CrackResU-Net, SA was applied to the two shallow skip connections. The architecture of SA is shown in Figure 3a. Inspired by the convolutional block attention module (CBAM; Woo et al., 2018), SA performs average and maximum pooling at each spatial location and concatenates the pooled features in the feature channel direction. Finally, a convolution operation with a 7 × 7 kernel size is performed to generate the attention map for reweighting the input of SA, which could help the network

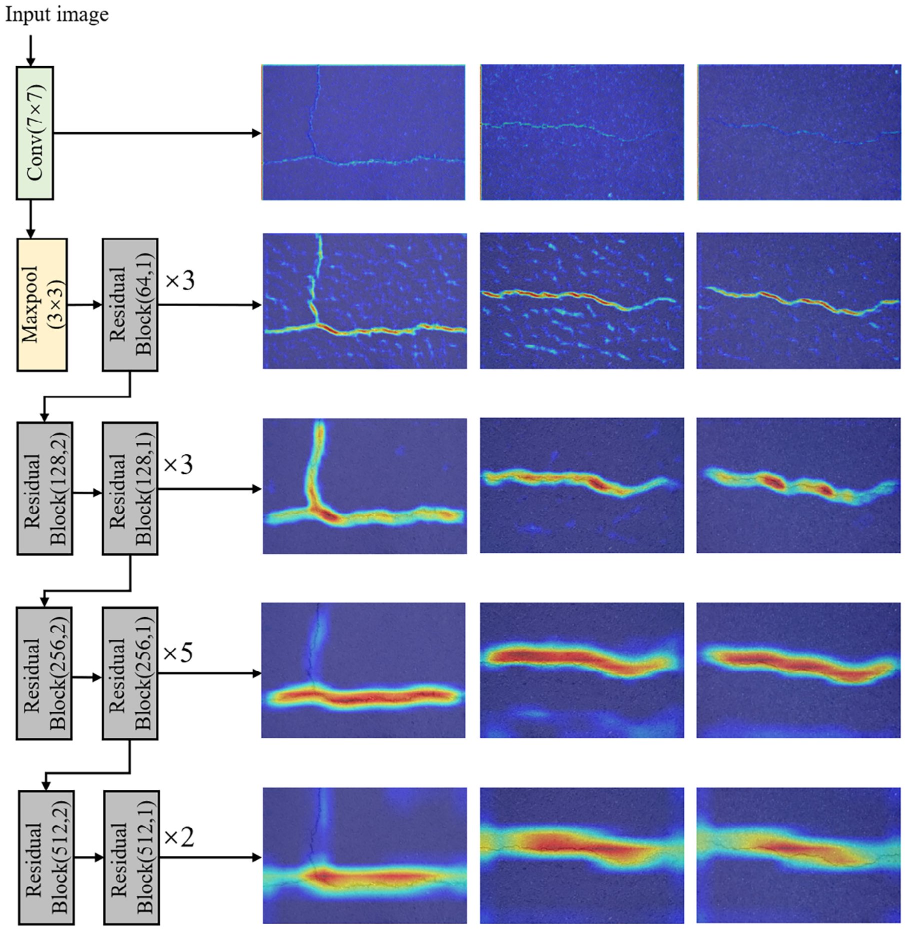

FIGURE 2 Visualization of the output feature maps for CrackResU-Net's different encoder layers without attention modules applied.

capture important pixel information and suppress useless information.

2.1.2 | PRAM

Long-range dependencies refer to connections between pixels or regions in an image that are far away from each other. The consideration of long-range dependencies allows the model to capture a wider range of context information and better understand the relationships between different regions in an image, which is critical for tasks such as image classification, object detection, and segmentation. While convolutional neural networks rely on convolution operators to model long-range dependencies, this approach has its limitations. Simply repeating convolution operations is computationally inefficient and challenging to optimize (X. Wang et al., 2018). This makes it difficult for the network to transfer information between long-range locations, which leads to ineffective modeling of long-range dependencies. The NL solves this problem better by introducing the idea of self-attention. As shown in Figure 3b, NL first performs dimensionality reduction on the input and generates three feature maps by three 1×1 kernel size convolution layers. Then, matrix multiplication is performed on two of the feature maps to obtain correlation coefficients between each feature spatial position and all other positions. After passing through the Softmax function, the correlation coefficient matrix is multiplied with the remaining feature map for weighted reinforcement. Finally, the output is summed with the original input after passing through a 1 × 1 kernel size convolution layer. Although such an architecture could enable the network to capture dependencies between related objects that are far away, the large computational cost of NL limits the inference speed, which is not conducive to practical applications.

To solve the above problem, the NL was first optimized, and SimNL was proposed. Figure 3c shows its architecture. Its design adopts the idea of the spatial attention branch of CBAM, which obtains the attention map by fusing the maximum and average data at each spatial location of the feature map. By performing maximum and average pooling in CPC, the maximum and average information at each spatial position in the two feature maps used for calculating the attention map in NL are extracted and integrated,

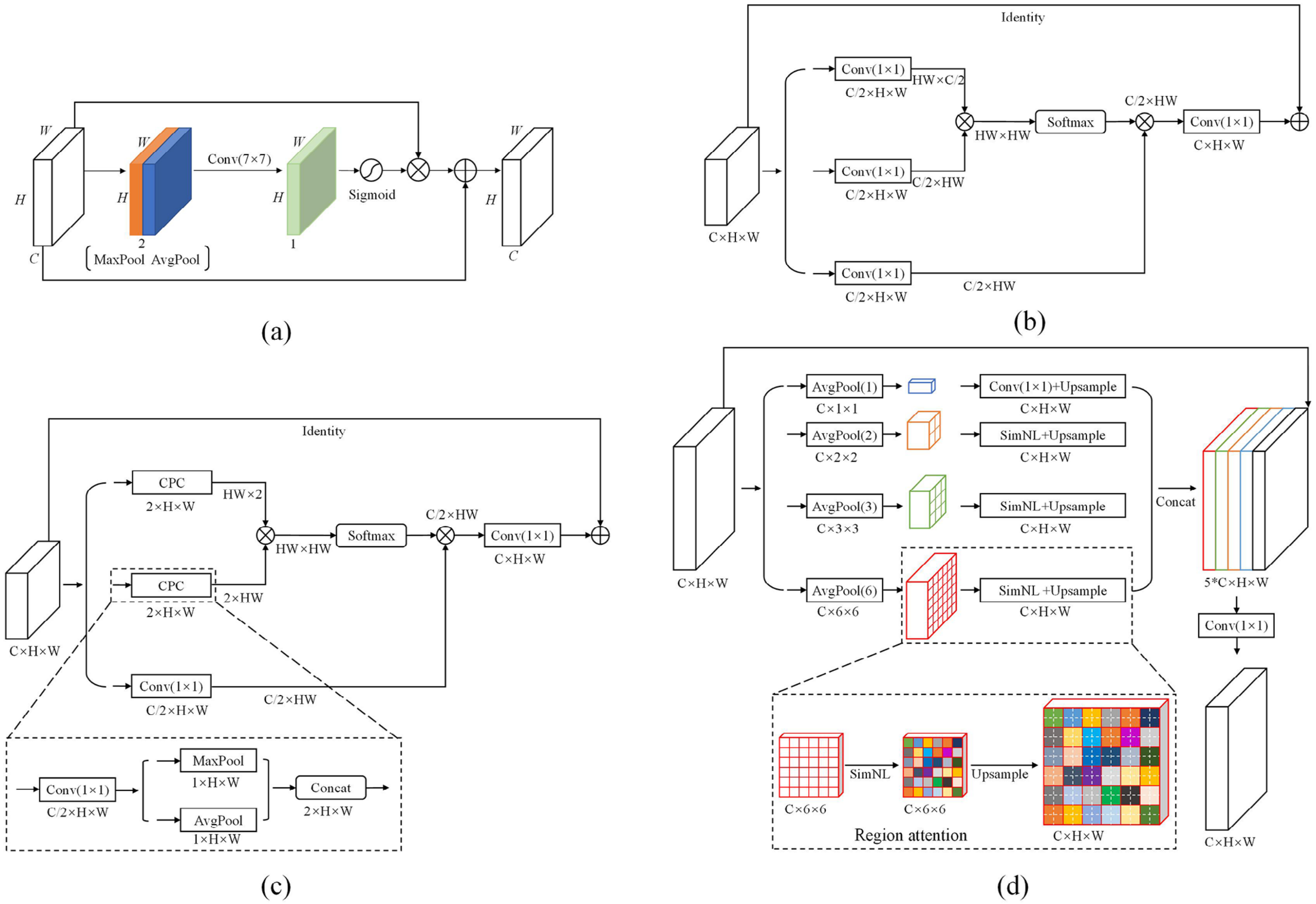

FIGURE 3 Network architectures: (a) spatial attention module (SA), (b) non-local (NL; X. Wang et al., 2018), (c) SimNL, and (d) pyramid region attention module (PRAM).

which leads to better attention map generation. Then, the PRAM was proposed by combining the PPM in a pyramid scene parsing network (PSPNet; Hengshuang Zhao et al., 2017) with SimNL. The architecture of PRAM is shown in Figure 3d. Four feature maps of sizes 1 × 1, 2 × 2, 3 × 3, and 6 × 6 are obtained by performing average pooling at four scales on each feature channel of the input. Then, except for the globally pooled feature map that passes through a 1 × 1 kernel size convolution layer, the remaining feature maps are processed by SimNL for long-range dependencies capture and spatial information enhancement of different sub-regions, and thus the proposed module is called the PRAM. It is worth noting that the residual connection of SimNL has been canceled in PRAM. Finally, all four sizes of feature maps are up-sampled to the input size and then concatenated with the input in the feature channel direction and output through a 1 × 1 kernel size convolution layer. Such an architecture not only allows the integration of context information at different scales, which is beneficial for the model's understanding of the global scene but also reduces the input feature map size of SimNL to a fixed value, which significantly decreases the computational cost. As shown in Figure 4, in this paper, PPM, and NL were simply connected to form PPM+NL, and its computational cost was compared with the shallow PRAM. When the input of the model is (320, 320, 3), the input feature map of PRAM and PPM+NL is (40, 40, 128), and the

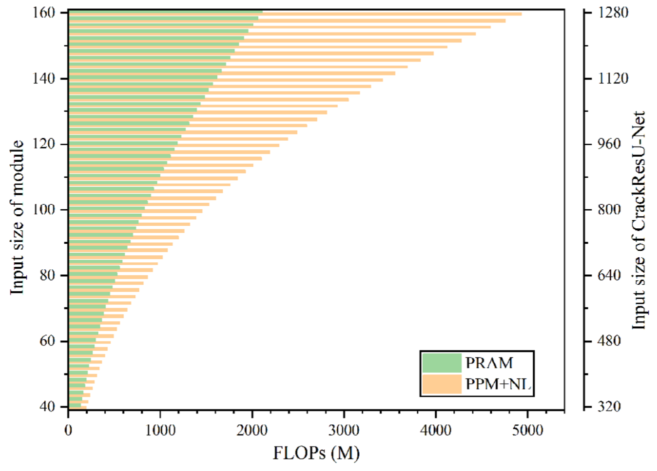

FIGURE 4 Computational cost of PRAM and pyramid pooling module (PPM)+NL.

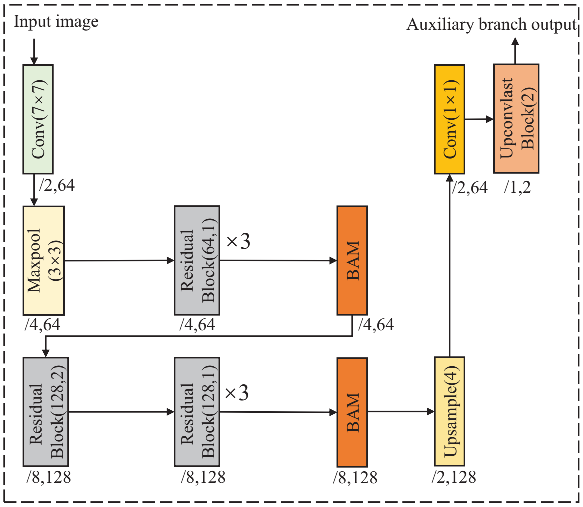

FIGURE 5 Network architecture of the auxiliary branch.

computational cost of PRAM is slightly less than that of PPM+NL. As the size of the detected image increases, the difference in computational cost between the two becomes more and more obvious. When the input of the model is (1280, 1280, 3), the computational cost of PRAM is less than half of that of PPM+NL, which indicates that multiscale region attention has a great advantage in detecting large-size images. The PRAM was applied to CrackResU-Net's deep skip connections to provide rich integration information for the decoder.

2.1.3 | Auxiliary branch

Inspired by the auxiliary branch in PSPNet, an auxiliary branch was added to CrackResU-Net's encoder to assist in encoder training. The auxiliary branch can provide additional gradient signals, allowing gradients to propagate better to shallow network parts, which helps alleviate gradient vanishing or exploding problems and improves the training stability of the encoder. Due to the different distribution of spatial and semantic information that different levels of network layers contain, for guiding the training more effectively, the auxiliary branch was applied to the middle position of CrackResU-Net's encoder, where the spatial and semantic information was more balanced. Figure 5 shows its architecture, which consists of a four times up-sampling layer, a 1 × 1 kernel size convolution layer, and an Upconvlast Block.

2.2 | Loss function

A Softmax function is used to process the output of CrackResU-Net to generate the final predicted class probabilities for each pixel point. The loss function of CrackResU-Net in this paper consists of a main loss L_{main} and an auxiliary loss L_{aux} as shown in Equation (1).

$$L = L_{\text{main}} + L_{\text{aux}} \tag{1}$$

As a commonly used loss function, cross-entropy loss can better deal with the problem of uneven distribution of categories in data (Yang et al., 2019). Dice coefficient loss is a loss function based on set similarity measurement, which considers the spatial relationship between predicted pixels and true labels (Augustauskas & Lipnickas, 2020). As a result, the main loss consists of the cross-entropy loss and the dice coefficient loss, which is shown in Equation (2).

$$L_{\text{main}} = \frac{1}{n}\sum_{\text{i}=1}^{n}\sum_{\text{c}=1}^{k} -p_{\text{ic}} \cdot \log\left(q_{\text{ic}}\right) + \frac{1}{k}\sum_{\text{c}=1}^{k}\left(1 - \frac{2 \cdot \sum_{\text{i}=1}^{n} p_{\text{ic}} \cdot q_{\text{ic}}}{\sum_{\text{i}=1}^{n} p_{\text{ic}} + \sum_{\text{i}=1}^{n} q_{\text{ic}}}\right) \tag{2}$$

where p_{ic} and q_{ic} are the true label and main branch prediction for the ith pixel that belongs to category c, respectively, n represents the total pixels in the image, and k is 2 in this paper.

The auxiliary loss consists of the dice coefficient loss, which is calculated by Equation (3).

$$L_{\text{aux}} = \frac{1}{k}\sum_{\text{c}=1}^{k}\left(1 - \frac{2 \cdot \sum_{\text{i}=1}^{n} p_{\text{ic}} \cdot z_{\text{ic}}}{\sum_{\text{i}=1}^{n} p_{\text{ic}} + \sum_{\text{i}=1}^{n} z_{\text{ic}}}\right) \tag{3}$$

where p_{ic} and z_{ic} are the true label and auxiliary branch prediction for the ith pixel that belongs to category c, respectively, n represents the total pixels in the image, and k is 2 in this paper.

2.3 | Optimization

In this paper, the Adam optimizer was selected to optimize CrackResU-Net. This involved calculating the gradient of a small batch of samples during each iteration, allowing for weight updates. Adam is similar to root mean squared propagation (RMSProp) but with the addition of momentum terms. It dynamically adjusts the learning rate of the network parameters using first-moment and second-raw-moment estimates of the gradient. One of its advantages is that the learning rate has a well-defined range for each iteration after bias correction, which contributes to increased parameter stability.

3 | EXPERIMENTS AND ANALYSIS

For validating CrackResU-Net's effectiveness, comparative tests were conducted on existing public datasets. The network training, data processing, and evaluation were conducted on a workstation with a Linux operating system (Intel Core i5-9400F @ 2.90 GHz CPU, and NVIDIA Geforce RTX 2060 GPU).

3.1 | Dataset establishment

In this paper, CFD was mainly utilized for CrackResU-Net's training and testing, and the additional experimental validation was conducted on Cracktree200 and Crack500. These three datasets and their processing are described as follows:

1. CFD: CFD is a commonly used dataset that contains 118 images taken by iPhone5 with 480 × 320 pixels in resolution. The images basically reflect the road conditions in Beijing, China. For expanding the training sample, after dividing these images into training, validation, and testing sets by a ratio of 6:2:2, cropping was performed on each image in steps of 20 pixels in horizontal and vertical directions to extract images with 320 × 320 pixels in resolution. Then, five data augmentation methods, including rotations of 90°, 180°, and 270° and flipping 180° horizontally and vertically, were applied to gain 3834 training images, 1296 validation images, and 1242 test images.
2. Cracktree200: Cracktree200 is a crack segmentation dataset proposed by Zou et al. (2012) and includes 206 crack images with 800 × 600 pixels in resolution. The images have challenges, such as low contrast and poor occlusion. In this paper, these images were divided into training and testing sets by a ratio of 6:4.
3. Crack500: Crack500 was presented by Yang et al. (2019) and contains 500 images of pavement cracks with 2000 × 1500 pixels in resolution. To fit the model input, each original image was cropped to 640 × 360 size, and thus 1896 training images, 348 validation images, and 1124 testing images were used in this paper.

3.2 | Evaluation metrics

For crack segmentation, pixels correctly predicted as cracks are true positive (TP), pixels incorrectly predicted as cracks are false positive (FP), pixels correctly predicted as background are true negative (TN), and pixels incorrectly predicted as background are false negative (FN). In this paper, precision (Pr), recall (Re), and F1 score are used to evaluate CrackResU-Net and are calculated as Equation (4), Equation (5), and Equation (6), respectively.

$$\mathrm{Pr} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

$$\mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

$$\text{F1 score} = \frac{2 \cdot \mathrm{Pr} \cdot \mathrm{Re}}{\mathrm{Pr} + \mathrm{Re}} \tag{6}$$

Since manual annotation often results in a transition region between non-cracked and cracked pixels, the predicted crack pixels within two pixels of the crack label can be regarded as TP, per the strategy used by Liu et al. (2020).

3.3 | Hyperparameters

For model training, the batch size used for calculating the gradient is set to 8, and L2 regularization with a weight decay factor of 0.00001 is applied. Considering that the decaying learning rate helps network training, this research adopted a dynamic learning rate modification strategy based on model performance. The dice coefficient is calculated on the validation set once every 1/8 epoch of training, according to Equation (7). When no better dice coefficient is obtained three times in a row, the learning rate decreases by 90%. In this paper, an initial learning rate of 0.0001 is adopted.

$$\text{Dice coefficient} = \frac{2 \cdot \mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TP} + \mathrm{FN}} \tag{7}$$

3.4 | Network training and model optimization on CFD

The strategy of Gaussian noise addition, random color augmentation, and random image translation was used during network training to increase the robustness of CrackResU-Net. Meanwhile, images with short sides of less than 320 pixels were scaled before passing through the model in order to ensure a sufficient amount of information contained in the pooling grid at the smallest scale in PRAM. The model was basically fitted after training six epochs, and the weights of the model that achieved the best dice coefficient on the validation set were saved for the next tests. In addition, Figure 6 visualizes the output feature maps of CrackResU-Net for different training periods with and without an auxiliary branch, and it can be seen that the addition of an auxiliary branch can significantly improve the model fitting speed, which is important for the optimization of the model parameters.

3.4.1 | The experiment of weight coefficient for the auxiliary loss

To explore the optimal percentage of auxiliary loss in the total loss, four auxiliary loss weight coefficients were

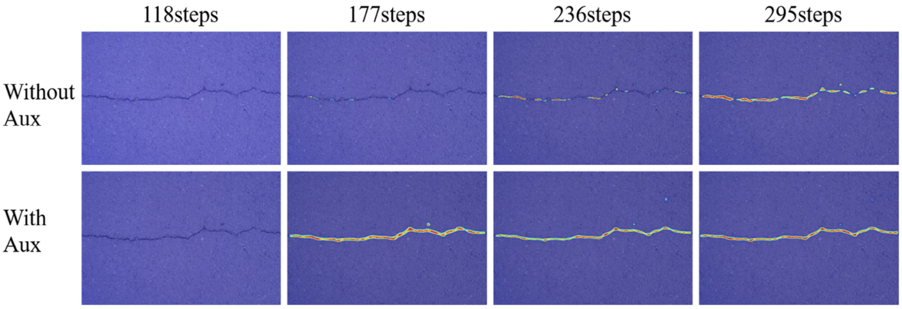

FIGURE 6 Visualization of CrackResU-Net with and without an auxiliary branch.

TABLE 1 Comparison of different weight coefficients for auxiliary loss.

Weight coefficients	Precision (Pr)	Recall (Re)	F1
0.3	**0.9678**	0.9405	0.9539
0.5	0.9554	**0.9520**	0.9537
0.7	0.9644	0.9489	0.9566
1.0	0.9659	0.9502	**0.9580**

The bolded numbers mean the best options in these coefficients.

experimentally verified. As shown in Table 1, the performance of CrackResU-Net tends to improve with the increase of weight coefficient for auxiliary loss and reaches the optimum at 1.0. Thus, the weight coefficient of auxiliary loss during training was selected as 1.0.

3.4.2 | An experiment of branch selection of PRAM

PRAM has four different scales of pooling branches (in the following, they are referred to by their average pooling size, i.e., 1, 2, 3, and 6). The more grids generated by the average pooling, the smaller the range of spatial information represented by each grid, and the more similar this branch is to the original NL. In order to find the optimal combining methods of branches, five combining methods were selected for experimental validation. As shown in Table 2, the model performance, computational cost, and parameters all gradually increase as the number of pooling branches used increases, and the best performance is reached when 1, 2, 3, and 6 branches are used at the same time. In addition, the model can reach the same level of performance as the simultaneous use of 1, 2, and 3 branches when the 6 branch is used alone, which indicates that the small-scale pooling branch seems to be more beneficial to the model in capturing the crack details, compared with the large-scale pooling branch. For better crack detection, CrackResU-Net used 1, 2, 3, and 6 branches simultaneously to implement PRAM.

TABLE 2 Comparison of different pooling branches used in pyramid region attention module (PRAM).

Pooling branches	Pr	Re	F1	Parameters	Floating point operations (FLOPs)
1	0.9541	0.9525	0.9533	**25.65 M**	**22,592.26 M**
12	**0.9704**	0.9400	0.9550	25.89 M	22,671.57 M
123	0.9679	0.9452	0.9564	26.14 M	22,751.70 M
6	0.9584	**0.9542**	0.9563	25.73 M	22,598.14 M
1236	0.9659	0.9502	**0.9580**	26.39 M	22,836.31 M

Note: FLOPs of all models were measured with the input of size 480 × 320. The bolded numbers mean the best options in these coefficients.

3.5 | Test results and analysis

3.5.1 | Test results on CFD

To verify the crack detection performance of CrackResU-Net, this research conducted comparative experiments and visualized some detection results using CrackResU-Net, FCN8s (Long et al., 2015), SegNet (Badrinarayanan et al., 2017), PSPNet, U-Net, and DeepLabv3+ (L. -C. Chen et al., 2018) on CFD. As shown in Figure 7, it is indicated that the detection results of FCN8s, SegNet, U-Net, and DeepLabv3+ all show FP points far from the true labels, while PSPNet and CrackResU-Net have no such problem. In addition, because CrackResU-Net makes full use of global multi-scale information and introduces SimNL to capture long-range dependencies, its detection results are smoother and more detailed, compared with other methods. Compared with DeepLabv3+, CrackResU-Net is slightly lacking in the recall of alligator cracks, and it has more advantages in accuracy. Overall, CrackResU-Net is able to accurately extract the majority of crack pixels regardless of lighting conditions, shadows, pavement markings, and other noise.

A quantitative comparison of the above models on CFD is shown in Table 3. CrackResU-Net has the highest Pr and F1 score among these methods, which fully demonstrates the effectiveness of CrackResU-Net in pavement crack

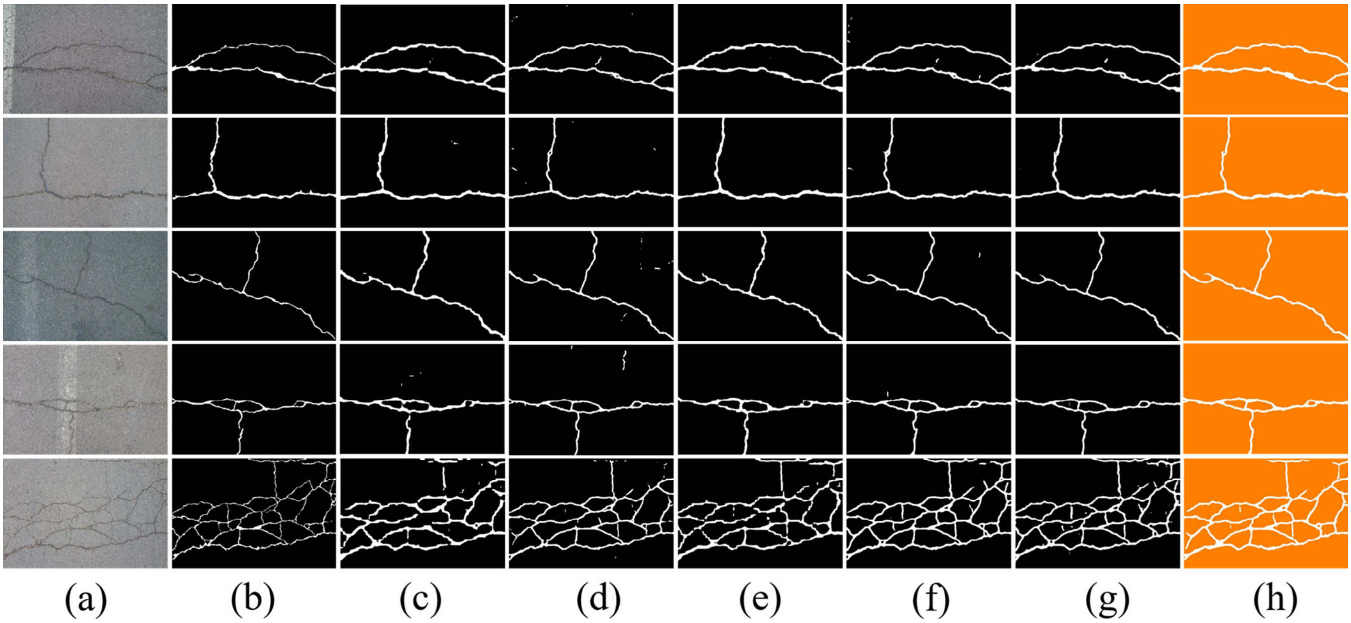

FIGURE 7 Detection results of CrackResU-Net and various semantic segmentation algorithms on CrackForest dataset: (a) original image, (b) true label, (c) FCN8s, (d) SegNet, (e) pyramid scene parsing network (PSPNet), (f) U-Net, (g) DeepLabv3+, and (h) CrackResU-Net.

TABLE 3 Performance comparison of CrackResU-Net and various semantic segmentation algorithms on CrackForest dataset (CFD).

Methods	Pr	Re	F1	Inference speed ms/image
FCN8s	0.9116	0.9500	0.9304	**24.97**
SegNet	0.9308	0.9186	0.9247	40.04
PSPNet	0.9123	0.9492	0.9304	48.95
DeepLabv3+	0.9494	**0.9526**	0.9510	34.73
U-Net	0.9392	0.9520	0.9455	58.39
CrackResU-Net	**0.9659**	0.9502	**0.9580**	25.89

Note: The inference speed was measured at a batch size of 1.
The bolded numbers mean the best options in these coefficients.

segmentation. Although DeepLabv3+ performs better in Re, its Pr is much lower than CrackResU-Net. In addition, in terms of inference speed, CrackResU-Net processes an image with a resolution of 480 × 320 in only 25.89 ms, which is 8.84 ms less than DeepLabv3+, and thus it is more beneficial in terms of practicality.

In addition, a quantitative comparison of CrackResU-Net with other crack segmentation algorithms on CFD is presented in Table 4, where it can be seen that CrackResU-Net has a higher F1 score, compared with other crack detection algorithms with the same tolerance margin.

3.5.2 | Test results on Crack500

Figure 8 shows some crack detection results of CrackResU-Net, FCN8s, SegNet, PSPNet, U-Net, and DeepLabv3+ on Crack500. It is shown that DeepLabv3+ performs the best among the other five semantic segmentation models, detecting the most crack pixel points, but it is inferior to CrackResU-Net in crack refinement and continuity. Especially from the detection results of the fifth crack image, compared with other models, CrackResU-Net can accurately extract crack pixels in pavement images containing complex textures and has better robustness.

TABLE 4 Performance comparison of CrackResU-Net and various crack segmentation algorithms on CFD.

Methods	Tolerance margin	F1
Otsu (1979)	2	0.1250
Canny (1986)	2	0.3268
Ai et al. (2018)	2	0.8700
Fan et al. (2018)	2	0.9244
U-HDN (Fan, Li, Chen, Wei, et al., 2020)	2	0.9390
ResU-Net (Lau et al., 2020)	2	0.9555
Liu et al. (2020)	2	0.9575
Ensemble network (Fan, Li, Chen, Di Mascio, et al., 2020)	2	0.9533
ResU-Net + ASPP (Augustauskas & Lipnickas, 2020)	2	0.9570
Parallel ResNet (Fan et al., 2022)	2	0.9563
CrackResU-Net	2	**0.9580**

Abbreviation: ASPP, atrous spatial pyramid pooling.
The bolded numbers mean the best options in these coefficients.

Table 5 quantitatively compares CrackResU-Net and five semantic segmentation models on Crack500. Corresponding to the visualization results, DeepLabv3+

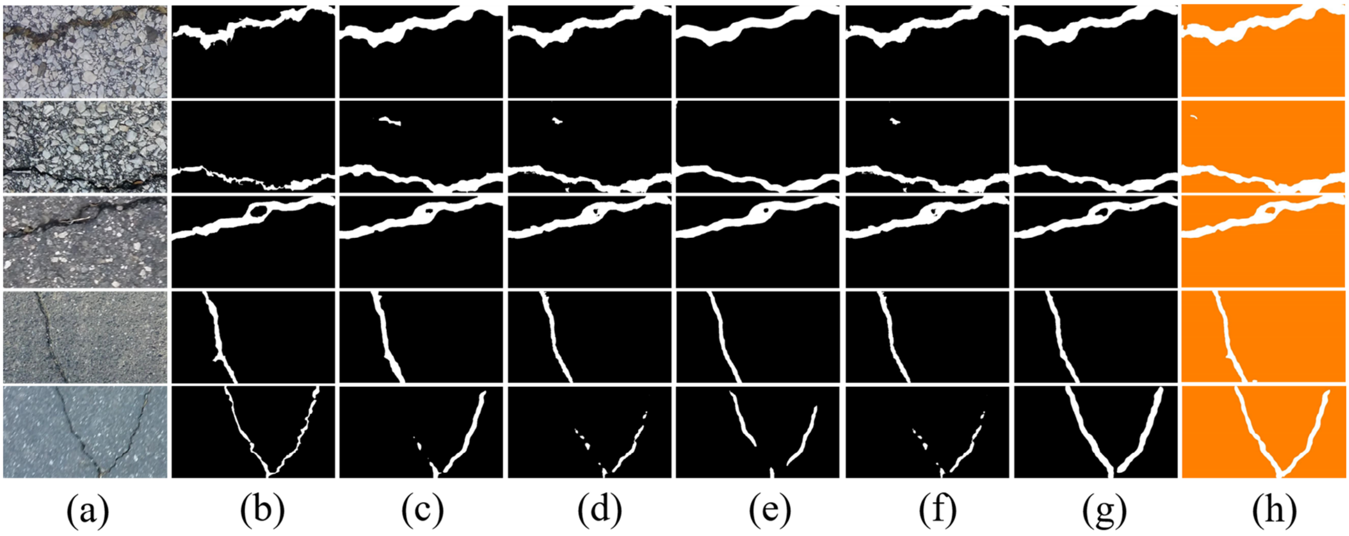

FIGURE 8 Detection results of CrackResU-Net and various semantic segmentation algorithms on Crack500: (a) original image, (b) true label, (c) FCN8s, (d) SegNet, (e) PSPNet, (f) U-Net, (g) DeepLabv3+, and (h) CrackResU-Net.

TABLE 5 Performance comparison of CrackResU-Net and various semantic segmentation algorithms on Crack500.

Methods	Pr	Re	F1	Inference speed ms/image
FCN8s	0.7037	0.8244	0.7593	26.42
SegNet	0.7373	0.7611	0.7490	47.87
PSPNet	0.7336	0.8204	0.7746	57.88
DeepLabv3+	0.7301	**0.8354**	0.7793	34.21
U-Net	**0.7465**	0.7868	0.7661	73.20
CrackResU-Net	0.7370	0.8320	**0.7816**	**26.14**

Note: The inference speed was measured at a batch size of 1.
The bolded numbers mean the best options in these coefficients.

TABLE 6 Performance comparison of CrackResU-Net and various crack segmentation algorithms on Crack500.

Methods	Tolerance margin	F1
W. Wang and Su (2020)	2	0.7681
ResU-Net (Lau et al., 2020)	2	0.7327
CrackResU-Net	2	**0.7816**

The bolded numbers mean the best options in these coefficients.

performs best among the other five segmentation algorithms and achieves the highest Re. CrackResU-Net achieves the best F1 score and inference speed.

In addition, a quantitative comparison of CrackResU-Net with other crack segmentation algorithms on Crack500 is presented in Table 6. It shows that CrackResU-Net performs better with the same tolerance margin.

TABLE 7 Performance comparison of CrackResU-Net and various semantic segmentation algorithms on Cracktree200.

Methods	Pr	Re	F1	Inference speed ms/image
FCN8s	0.9129	0.9412	0.9268	52.06
SegNet	**0.9692**	0.9031	0.9350	97.74
PSPNet	0.9057	**0.9576**	0.9309	124.27
DeepLabv3+	0.9654	0.9572	**0.9613**	54.90
U-Net	0.9529	0.9570	0.9550	146.18
CrackResU-Net	0.9660	0.9551	0.9605	**41.12**

Note: The inference speed was measured at a batch size of 1.
The bolded numbers mean the best options in these coefficients.

3.5.3 | Test results on Cracktree200

The cracks in the Cracktree200 are labeled finer and thus have fewer crack pixels, which results in difficult learning of the crack segmentation model during the initial training phase. Therefore, the model weights trained on CFD were used when performing the model training. Figure 9 visualizes some detection results of CrackResU-Net, FCN8s, SegNet, PSPNet, U-Net, and DeepLabv3+ on Cracktree200. Compared with other methods, DeepLabv3+ and CrackResU-Net are more accurate for capturing tiny cracks and are not affected by the lighting conditions and shadows in the images. In addition, DeepLabv3+ is stronger in recalling crack pixels, while CrackResU-Net is more accurate in detecting cracks and has smoother and more continuous detection results.

Table 7 quantitatively compares the CrackResU-Net and five segmentation models on Cracktree200. Unlike the

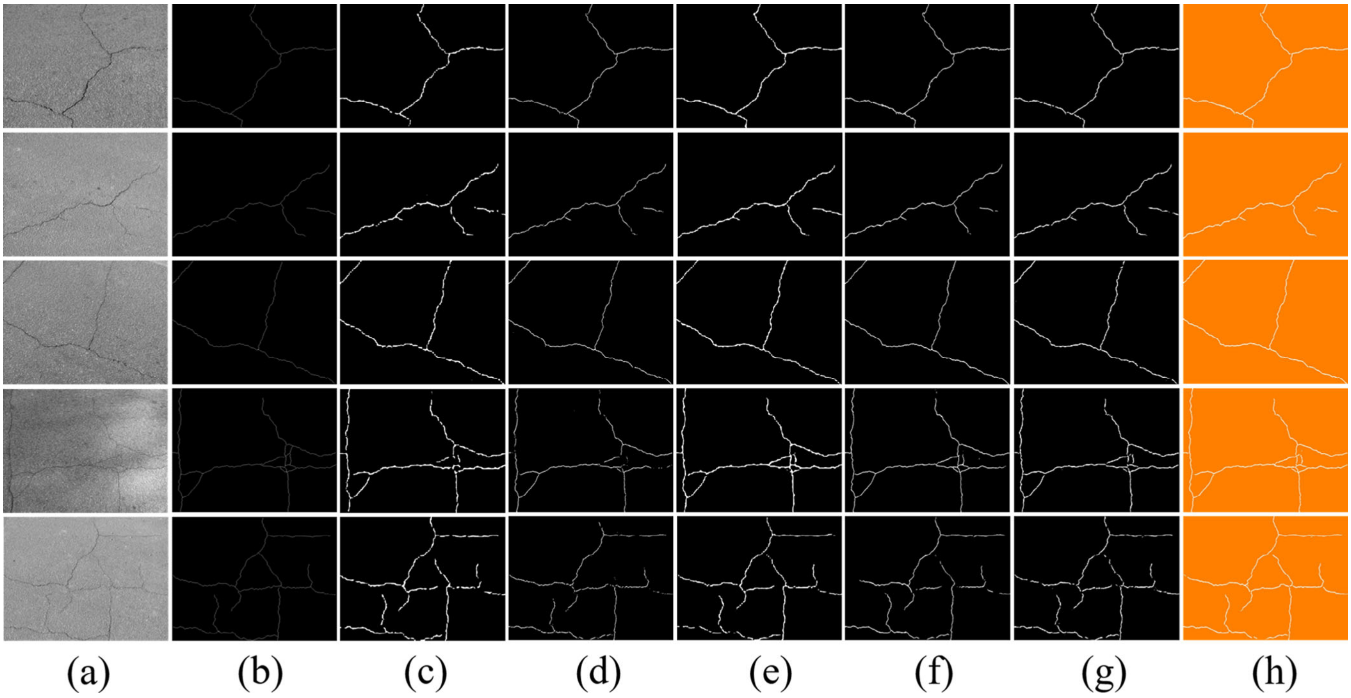

FIGURE 9 Detection results of CrackResU-Net and various semantic segmentation algorithms on Cracktree200: (a) original image, (b) true label, (c) FCN8s, (d) SegNet, (e) PSPNet, (f) U-Net, (g) DeepLabv3+, and (h) CrackResU-Net.

TABLE 8 Performance comparison of CrackResU-Net and various crack segmentation algorithms on Cracktree200.

Methods	Tolerance margin	F1
CrackTree (Zou et al., 2012)	2	0.8500
Parallel ResNet (Fan et al., 2022)	2	0.9308
CrackResU-Net	2	**0.9605**

The bolded numbers mean the best options in these coefficients.

TABLE 9 Ablation experiments on CFD.

Models	F1	FLOPs
U-Net	0.9455	153,577.88 M
ResU-Net+BAM	0.9458	**22,428.35 M**
CrackResU-Net without SA	0.9564	22,831.60 M
CrackResU-Net without Aux	0.9516	22,836.31 M
CrackResU-Net without PRAM	0.9504	22,433.05 M
CrackResU-Net (PRAM replaced by PPM)	0.9533	22,832.28 M
CrackResU-Net	**0.9580**	22,836.31 M

Note: Aux indicates the application of an auxiliary branch; ResU-Net+BAM indicates U-Net embedded with ResNet-34 and BAM; FLOPs of all models were measured with the input of size 480 × 320.

Abbreviations: BAM, bottleneck attention module; PPM, pyramid pooling module; SA, spatial attention module.

The bolded numbers mean the best options in these coefficients.

results in the two previous datasets, DeepLabv3+ achieves the highest F1 score, which may be due to the fact that the true labels of the cracks on Cracktree200 are too fine to match the pooling branches in PRAM. For example, except for the original feature map without scaling, the grids generated by other pooling branches contain too large a range of information, resulting in difficulties in capturing position information of tiny cracks. Using smaller-scale pooling branches in PRAM or weighting different branches may alleviate this problem in the future. Compared with DeepLabv3+, CrackResU-Net sacrifices an F1 score of 0.0008 and reduces the processing time by 13.78 ms for an image with a resolution of 800 × 600, which is more advantageous.

In addition, a quantitative comparison of CrackResU-Net with other crack segmentation algorithms on Cracktree200 is presented in Table 8, and CrackResU-Net achieves a better F1 score with the same tolerance margin.

4 | RESULT ANALYSIS AND DISCUSSION

4.1 | Ablation experiment

To understand the behavior of ResNet-34 with BAM, PRAM, SA, and the auxiliary branch, a comprehensive ablation experiment of CrackResU-Net on CFD was conducted. As in Table 9, ResNet-34 with BAM is a good encoder, which not only reduces a lot of computational costs, compared with U-Net, but also has a slight improvement in performance. On this basis, the combination of SA, auxiliary branch, and PRAM can provide a 1.22%

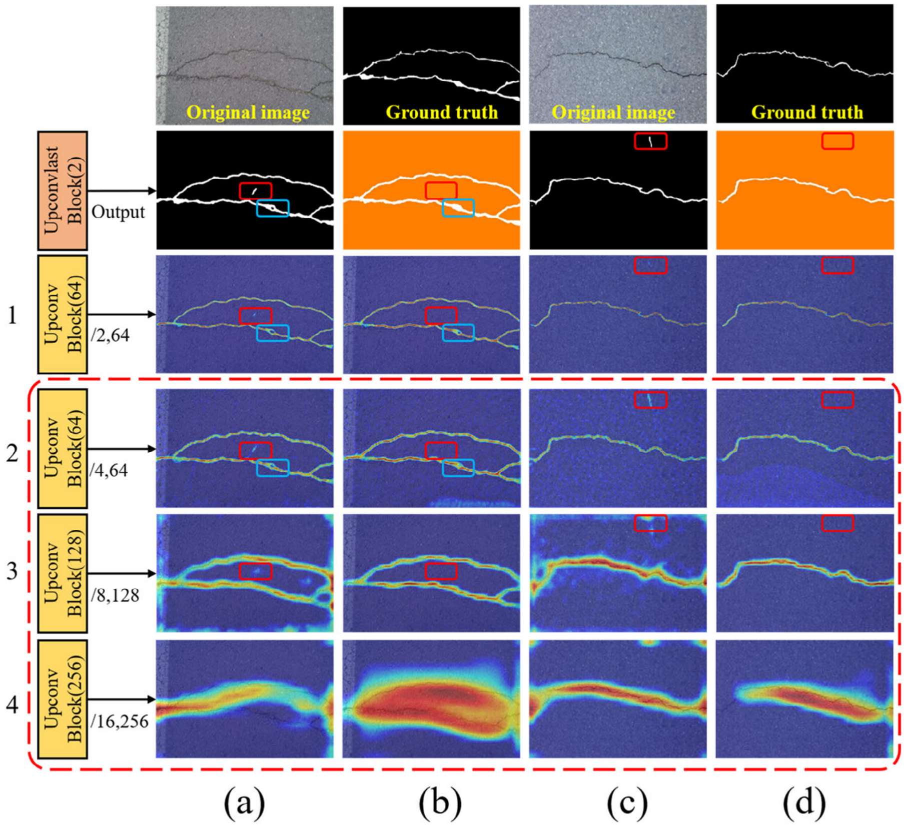

FIGURE 10 Visualization of the output feature maps for different decoder layers: (a) ResU-Net+bottleneck attention module (BAM), (b) CrackResU-Net, (c) ResU-Net+BAM, and (d) CrackResU-Net.

improvement in the F1 score. By observing the experimental results, it can be noticed that PRAM contributes the most to the performance improvement of CrackResU-Net, and its performance is better than PPM, which fully proves the effectiveness of long-range dependencies capture. The addition of an auxiliary branch not only does not contribute to computational complexity but also significantly improves model performance. In addition, although SA improves the performance of CrackResU-Net by a small amount only, it still helps in model optimization.

4.2 | Visualization of the behaviors of PRAM and SA

Figure 10 visualizes the output feature maps for different decoder layers when processing pavement crack images using ResU-Net+BAM and CrackResU-Net. It can be seen that after integrating the rich semantic information provided by PRAM, the third Upconv Block of CrackResU-Net locates the cracks more accurately, focusing the scattered attention points around the cracks. Compared with ResU-Net+BAM, FP points in the red solid box are eliminated. The first and second Upconv blocks refine the previous feature maps and the addition of SA results in fewer FN points in the blue solid box. Although the improvement of SA is not significant in the figure, it cannot be denied that the results of the ablation experiment demonstrate its optimization on the model. Through further observation, the predicted results are more similar to the visualization of the second Upconv block, indicating that the parts within the red dashed box are more important to the model. Therefore, optimizing this part and pruning other parts of the decoder may be future research directions.

4.3 | Comparison with other skip connection methods

The PRAM and SA proposed in this paper were applied to skip connections to improve detection performance. In order to further verify the effectiveness of this modified skip connection method, the comparison of other improved skip connection methods used for crack detection was conducted on CFD. As shown in Table 10, the proposed method has significant advantages over other algorithms in terms of Re and F1 scores and does not increase a lot of computational costs.

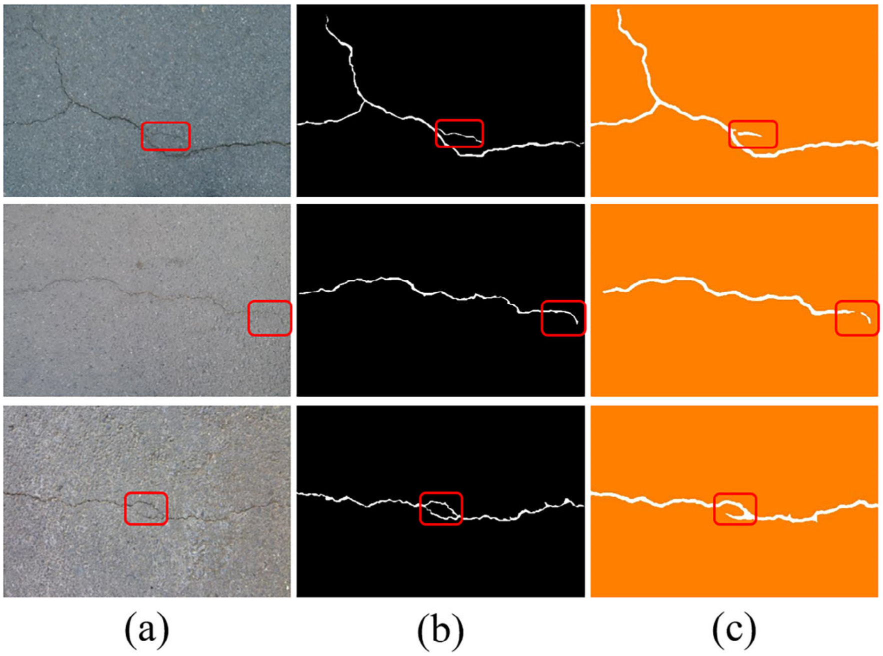

FIGURE 11 Several detection results with false negative points: (a) original image, (b) true label, and (c) CrackResU-Net.

TABLE 10 Comparison of the proposed method and other skip connection methods.

Methods	Pr	Re	F1	FLOPs
CrackResU-Net (AG) (L. -C. Chen & He, 2022)	**0.9669**	0.9349	0.9506	22,719.70 M
CrackResU-Net (CAB) (Hsieh & Tsai, 2021)	0.9632	0.9423	0.9526	**22,707.33 M**
CrackResU-Net	0.9659	**0.9502**	**0.9580**	22,836.31 M

Note: FLOPs of all models were measured with the input of size 480 × 320.
Abbreviations: AG, attention gate; CAB, channel attention block.
The bolded numbers mean the best options in these coefficients.

TABLE 11 Comparison of SimNL and non-local (NL) in PRAM.

Modules	Pr	Re	F1
NL	**0.9664**	0.9450	0.9556
SimNL	0.9659	**0.9502**	**0.9580**

The bolded numbers mean the best options in these coefficients.

4.4 | Comparison of SimNL and NL

To validate the effectiveness of SimNL, a comparison experiment between SimNL and NL in PRAM was conducted. As shown in Table 11, SimNL outperforms in terms of contributing F1 score and Re.

4.5 | FN problems for fine cracks

Although CrackResU-Net has shown good detection performance on CFD, Cracktree200, and Crack500, it has FN issues in detecting some fine cracks. As shown in Figure 11, the small cracks in the red boxes that are difficult to distinguish with the eyes are not detected. This corresponds to the test results on Cracktree200, possibly due to the poor matching of pooling branches in PRAM with these small crack points, and further improvement is needed in the future.

5 | CONCLUSION

An effective pavement crack segmentation approach based on the modified U-Net, namely, CrackResU-Net, was proposed in this paper. ResNet-34 with BAM was first embedded into U-Net for efficient encoder feature extraction, and an auxiliary branch was added to the model for better encoder training. Then a new module named PRAM was proposed by combining the PPM and modified NL, which could achieve multi-scale context information integration and long-range dependencies capture. Moreover, since the PRAM reduces the input feature map size of modified NL to a fixed value, the computational cost is significantly saved, which is more advantageous for detecting large-size images. PRAM was applied to CrackResU-Net's deep skip connections to provide rich integration information for the decoder, and the SA module was applied to CrackResU-Net's shallow skip connections to enhance crack spatial details.

CFD was used for the training, validation, and testing of CrackResU-Net. Test results showed that CrackResU-Net achieved Pr, Re, and F1 scores of 0.9659, 0.9502, and 0.9580, respectively. In addition, it took only 25.89 ms to process an image with a resolution of 480 × 320, which achieved a better balance between speed and accuracy, compared with several other state-of-the-art crack segmentation methods. To further evaluate its effectiveness and robustness, the test results with other crack detection models on Crack500 and Cracktree200 were also discussed, and CrackResU-Net still performed well. However, it had FN issues in detecting some fine cracks, and using smaller-scale pooling branches in PRAM or weighting different branches may alleviate this problem in the future.

Finally, comprehensive experiments were performed on CFD to discuss in detail the behavior of the important components of CrackResU-Net. For PRAM, it contributed the most to the performance improvement of CrackResU-Net and could help the decoder locate the cracks more accurately, which fully proves the effectiveness of long-range dependencies capture. For the auxiliary branch, when it was applied to the middle position of CrackResU-Net's encoder, it not only accelerated the model fitting but also had a significant improvement in model performance without increasing computational cost. As a result, it could be further improved and applied to future detection models. For SA, its improvement in model performance was little, but it still helped in model optimization. In addition, the improved skip connection method combining SA and PRAM exhibited better performance than other modified skip connection methods with attention, and this method may be extended to other image recognition tasks.

In the future, we will focus on detecting more pavement diseases, such as potholes and ruts. In addition, by combining ground-penetrating radar data, 3D reconstruction of the pavement structure will be realized, and more information on hidden distresses can be obtained, which is of great significance for both pavement design and maintenance.

ACKNOWLEDGMENTS

The authors appreciate the financial support from Hunan Expressway Group Co. Ltd and the Hunan Department of Transportation (No. 202152) in China. The authors also appreciate the funding support from the National Natural Science Foundation of China (No. 51778038), the China National Key R&D Program (2021YFB2600600, 2021YFB2600604, 2018YFB1600202), National Natural Science Foundation of China (grant number: 52108403), and the Beijing high-level overseas talents returned to China. Any views and findings presented in this paper are those of the authors and do not necessarily represent the opinion of any organization.

REFERENCES

Ai, D., Jiang, G., Kei, L. S., & Li, C. (2018). Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods. *IEEE Access*, *6*, 24452–24463.

Augustauskas, R., & Lipnickas, A. (2020). Improved pixel-level pavement-defect segmentation using a deep autoencoder. *Sensors*, *20*(9), 2557.

Ayenu-Prah, A., & Attoh-Okine, N. (2008). Evaluating pavement cracks with bidimensional empirical mode decomposition. *EURASIP Journal on Advances in Signal Processing*, *2008*, 861701.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495.

Bang, S., Park, S., Kim, H., & Kim, H. (2019). Encoder–decoder network for pixel-level road crack detection in black-box images. *Computer-Aided Civil and Infrastructure Engineering*, *34*(8), 713–727.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*(6), 679–698.

Chen, J., & He, Y. (2022). A novel U-shaped encoder–decoder network with attention mechanism for detection and evaluation of road cracks at pixel level. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1721–1736.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (pp. 801–818).

Cheng, H. D., Wang, J., Hu, Y. G., Glazier, C., Shi, X. J., & Chen, X. W. (2001). Novel approach to pavement cracking detection based on neural network. *Transportation Research Record*, *1764*(1), 119–127.

Fan, Z., Li, C., Chen, Y., Di Mascio, P., Chen, X., Zhu, G., & Loprencipe, G. (2020). Ensemble of deep convolutional neural networks for automatic pavement crack detection and measurement. *Coatings*, *10*(2), 152.

Fan, Z., Li, C., Chen, Y., Wei, J., Loprencipe, G., Chen, X., & Di Mascio, P. (2020). Automatic crack detection on road pavements using encoder-decoder architecture. *Materials*, *13*(13), 2960.

Fan, Z., Lin, H., Li, C., Su, J., Bruno, S., & Loprencipe, G. (2022). Use of parallel ResNet for high-performance pavement crack detection and measurement. *Sustainability*, *14*(3), 1825.

Fan, Z., Wu, Y., Lu, J., & Li, W. (2018). *Automatic pavement crack detection based on structured prediction with the convolutional neural network*. arXiv Preprint. arXiv:1802.02208.

Hassanpour, A., Moradikia, M., Adeli, H., Khayami, S. R., & Shamsinejadbabaki, P. (2019). A novel end-to-end deep learning scheme for classifying multi-class motor imagery electroencephalography signals. *Expert Systems*, *36*(6), e12494.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV (pp. 770–778).

Hou, Y., Li, Q., Han, Q., Peng, B., Wang, L., Gu, X., & Wang, D. (2021). MobileCrack: Object classification in asphalt pavements using an adaptive lightweight deep learning. *Journal of Transportation Engineering, Part B: Pavements*, *147*(1), 04020092.

Hsieh, Y.-A., & Tsai, Y.-C. J. (2021). DAU-Net: Dense attention U-Net for pavement crack segmentation. *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, IN (pp. 2251–2256).

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT (pp. 7132–7141).

Jenkins, M. D., Carr, T. A., Iglesias, M. I., Buggy, T., & Morison, G. (2018). A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks. *2018 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy (pp. 2120–2124).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105.

Lau, S. L., Chong, E. K., Yang, X., & Wang, X. (2020). Automated pavement crack segmentation using U-Net-based convolutional neural network. *IEEE Access*, *8*, 114892–114899.

Li, N., Hou, X., Yang, X., & Dong, Y. (2009). Automation recognition of pavement surface distress based on support vector machine. *2009 Second International Conference on Intelligent Networks and Intelligent Systems*, Tianjin, China (pp. 346–349).

Li, Q., & Liu, X. (2008). Novel approach to pavement image segmentation based on neighboring difference histogram method. *2008 Congress on Image and Signal Processing*, Hainan, China (Vol. *2*, pp. 792–796).

Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Cheng-Siong Lee, V., & Ding, L. (2020). Automated pavement crack detection and segmentation based on two-step convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *35*(11), 1291–1305.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA (pp. 3431–3440).

Macias-Garcia, E., Galeana-Perez, D., Medrano-Hermosillo, J., & Bayro-Corrochano, E. (2021). Multi-stage deep learning perception system for mobile robots. *Integrated Computer-Aided Engineering*, *28*(2), 191–205.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1127–1141.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). *Attention U-Net: Learning where to look for the pancreas. ar*Xiv Preprint. arXiv:1804.03999.

Olamat, A., Ozel, P., & Atasever, S. (2022). Deep learning methods for multi-channel EEG-based emotion recognition. *International Journal of Neural Systems*, *32*(5), 2250021.

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

Ong, J. C., Lau, S. L., Ismadi, M.-Z., & Wang, X. (2023). Feature pyramid network with self-guided attention refinement module for crack segmentation. *Structural Health Monitoring*, *22*(1), 672–688.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66.

Park, J., Woo, S., Lee, J.-Y., & Kweon, I. S. (2018). *BAM: Bottleneck attention module*. arXiv Preprint. arXiv:1807.06514.

Pauly, L., Hogg, D., Fuentes, R., & Peel, H. (2017). Deeper networks for pavement crack detection. *Proceedings of the 34th ISARC, IAARC*, Taipei, Taiwan (pp. 479–485).

Qiao, W., Liu, Q., Wu, X., Ma, B., & Li, G. (2021). Automatic pixel-level pavement crack recognition using a deep feature aggregation segmentation network with a scse attention mechanism module. *Sensors*, *21*(9), 2902.

Rafiei, M. H., & Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, *142*(2), 04015066.

Rafiei, M. H., & Adeli, H. (2017). NEEWS: A novel earthquake early warning model using neural dynamic classification and neural dynamic optimization. *Soil Dynamics and Earthquake Engineering*, *100*, 417–427.

Rafiei, M. H., & Adeli, H. (2018). Novel machine-learning model for estimating construction costs considering economic variables and indexes. *Journal of Construction Engineering and Management*, *144*(12), 04018106.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2), 237.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference*, Munich, Germany (pp. 234–241).

Roy, A. G., Navab, N., & Wachinger, C. (2018). Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks. *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference*, Granada, Spain (pp. 421–429).

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy (pp. 618–626).

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, *17*(12), 3434–3445.

Song, W., Jia, G., Jia, D., & Zhu, H. (2019). Automatic pavement crack detection and classification using multiscale feature attention network. *IEEE Access*, *7*, 171001–171012.

Song, W., Jia, G., Zhu, H., Jia, D., & Gao, L. (2020). Automated pavement crack damage detection using deep multiscale convolutional features. *Journal of Advanced Transportation*, *2020*, 6412562.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. NIPS'17: Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems *30*, Long Beach, CA (pp. 6000–6010).

Wan, H., Gao, L., Su, M., Sun, Q., & Huang, L. (2021). Attention-based convolutional neural network for pavement crack detection. *Advances in Materials Science and Engineering*, *2021*, 5520515.

Wang, W., & Su, C. (2020). Convolutional neural network-based pavement crack segmentation using pyramid attention network. *IEEE Access*, *8*, 206548–206558.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT (pp. 7794–7803).

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (pp. 3–19).

Xiang, X., Zhang, Y., & El Saddik, A. (2020). Pavement crack detection network based on pyramid structure and attention mechanism. *IET Image Processing*, *14*(8), 1580–1586.

Xu, G., Ma, J., Liu, F., & Niu, X. (2008). Automatic recognition of pavement surface crack based on BP neural network. *2008 International Conference on Computer and Electrical Engineering*, Phuket, Thailand (pp. 19–22).

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2019). Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *21*(4), 1525–1535.

Yao, H., Liu, Y., Li, X., You, Z., Feng, Y., & Lu, W. (2022). A detection method for pavement cracks combining object detection and attention mechanism. *IEEE Transactions on Intelligent Transportation Systems*, *23*(11), 22179–22189.

Ye, W., Ren, J., Zhang, A. A., & Lu, C. (2023). Automatic pixel-level crack detection with multi-scale feature fusion for slab tracks. *Computer-Aided Civil and Infrastructure Engineering*, *38*(18), 2648–2665.

Zhao, H., Qin, G., & Wang, X. (2010). Improvement of canny algorithm based on pavement edge detection. *2010 3rd International Congress on Image and Signal Processing*, Yantai, China (Vol. *2*, pp. 964–967).

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI (pp. 2881–2890).

Zhou, Q., Qu, Z., & Cao, C. (2021). Mixed pooling and richer attention feature fusion for crack detection. *Pattern Recognition Letters*, *145*, 96–102.

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, *33*(3), 227–238.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & Wang, S. (2018). DeepCrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, *28*(3), 1498–1512.

How to cite this article: Yao, H., Liu, Y., Lv, H., Huyan, J., You, Z., & Hou, Y. (2023). Encoder–decoder with pyramid region attention for pixel-level pavement crack recognition. *Computer-Aided Civil and Infrastructure Engineering*, 1–17. https://doi.org/10.1111/mice.13128