DOI: 10.1111/mice.13103

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

A lightweight encoder–decoder network for automatic pavement crack detection

Guijie Zhu[1,2] | **Jiacheng Liu**[1,2] | **Zhun Fan**[1,2] | **Duan Yuan**[1,2] | **Peili Ma**[1,2] | **Meihua Wang**[3] | **Weihua Sheng**[4] | **Kelvin C. P. Wang**[5]

[1]College of Engineering, Shantou University, Shantou, China

[2]International Cooperation Base of Evolutionary Intelligence and Robotics of Guangdong Province, Shantou University, Shantou, China

[3]College of Mathematics & Informatics, South China Agricultural University, Guangzhou, China

[4]School of Electrical & Computer Engineering, Oklahoma State University, Oklahoma, Stillwater, USA

[5]School of Civil & Environmental Engineering, Oklahoma State University, Oklahoma, Stillwater, USA

Correspondence
Zhun Fan, College of Engineering, Shantou University, Shantou 515063, China.
Email: zfan@stu.edu.cn

Meihua Wang, College of Mathematics & Informatics, South China Agricultural University, Guangzhou 501642, China.
Email: wangmeihua@scau.edu.cn

Funding information
National Key Research and Development Program of China, Grant/Award Numbers: 2021ZD0111501, 2021ZD0111502; National Natural Science Foundation of China, Grant/Award Number: 62176147; Science and Technology Planning Project of Guangdong Province of China, Grant/Award Numbers: 2019A050520001, 2021A0505030072; Science and Technology Special Funds Project of Guangdong Province of China, Grant/Award Numbers: STKJ2021216, STKJ2021019

Abstract

Cracks are the most common damage type on the pavement surface. Usually, pavement cracks, especially small cracks, are difficult to be accurately identified due to background interference. Accurate and fast automatic road crack detection play a vital role in assessing pavement conditions. Thus, this paper proposes an efficient lightweight encoder–decoder network for automatically detecting pavement cracks at the pixel level. Taking advantage of a novel encoder–decoder architecture integrating a new type of hybrid attention blocks and residual blocks (RBs), the proposed network can achieve an extremely lightweight model with more accurate detection of pavement crack pixels. An image dataset consisting of 789 images of pavement cracks acquired by a self-designed mobile robot is built and utilized to train and evaluate the proposed network. Comprehensive experiments demonstrate that the proposed network performs better than the state-of-the-art methods on the self-built dataset as well as three other public datasets (CamCrack789, Crack500, CFD, and DeepCrack237), achieving F1 scores of 94.94%, 82.95%, 95.74%, and 92.51%, respectively. Additionally, ablation studies validate the effectiveness of integrating the RBs and the proposed hybrid attention mechanisms. By introducing depth-wise separable convolutions, an even more lightweight version of the proposed network is created, which has a comparable performance and achieves the fastest inference speed with a model parameter size of only 0.57 M. The developed mobile robot system can effectively detect pavement cracks in real scenarios at a speed of 25 frames per second.

1 | INTRODUCTION

In civil engineering structures, cracking is the most critical type of distress, existing in substantial civil infrastructures such as bridges, roads, buildings, dams, tunnels, and others. Detecting cracks in time is of great significance to maintain the safety and serviceability of roads. The traditional manual-based inspection of pavement cracks is time-consuming and costly in most cases. In the past few decades, several image-processing technologies, such as edge detection (Zhao et al., 2010), thresholding (Peng et al., 2020), and mathematical morphology (Tanaka & Uematsu, 1998), have been utilized to detect pavement cracks. However, the performances of these algorithms are highly susceptible to background noises such as water stains, oil stains, leaves, branches, pavement markings, shadows, and maintenance hole covers, partly because these methods use only relatively simple features. Consequently, accurately and efficiently inspecting cracks on the surface of pavement is still a challenging task.

Recently, vision-based crack detection methods have garnered considerable interest from both academia and industry owing to their advantages of safety, cost, efficiency, and objectivity (Jang et al., 2021; F. Yang et al., 2020). Deep learning techniques have been successfully employed in object detection and image classification tasks with superior experimental results. For example, sliding window methods based on convolutional neural networks (CNNs; Cha et al., 2017; I. Zhang et al., 2016) that can classify each sliding window as a crack region or non-crack region have been developed. However, it is difficult to select an appropriate window size to achieve superior detection performance. To address this problem, region-based CNNs (Deng et al., 2020; Kim & Cho, 2019) and "You Only Look Once"-based (J. Liu et al., 2020; C. Zhang et al., 2020) methods have been proposed for generating flexible proposal regions. However, these methods can only identify approximate crack locations. When the fine details of cracks need to be known, the cracks must be detected at the pixel level.

Recently, CNNs have become widely used in the field of computer vision, and an increasing number of CNNs have been successfully applied to damage detection in civil engineering at the pixel level (X. Pan & Yang, 2020; N. Wang et al., 2018). For example, Fan et al. (2018) proposed a CNN-based method for learning the different structures of cracks from images. In addition, Fan, Li, Chen, Di Mascio, et al. (2020) constructed an ensemble CNN consisting of three visual geometry group (VGG) nets that vote on crack detection results. Furthermore, CNN-based approaches that detect cracks at the pixel level, including CrackNet (A. Zhang et al., 2017), CrackNet II (A. Zhang et al., 2019), and CrackNet V (Fei et al., 2020), have been proposed. The latter two methods achieved higher accuracy and speed than the original CrackNet. In addition, Choi and Cha (2019) utilized a CNN-based method to detect concrete cracks in images. They implemented both separable and dilated convolution to design a lightweight crack segmentation network, which significantly reduced the detection time.

In 2015, fully convolutional network (FCN), a special kind of CNN, was developed by Long et al. (2015) for image segmentation at the semantic level. Unlike traditional CNNs, which have fully connected layers at the output, FCNs employ fully convolutional layers so that the network has no limit on the input size, which greatly promotes the development of segmentation tasks and pixel-level target detection tasks.

Inspired by probabilistic autoencoders, a deep convolutional encoder–decoder network was developed from the FCN (Ranzato et al., 2007). In recent years, U-Net (Ronneberger et al., 2015) and its variants based on the encoder–decoder structure have become extensively utilized for crack detection tasks at the pixel level (T. Chen et al., 2020; W. Wang & Su, 2021; A. A. Zhang et al., 2022). In the research of Cheng et al. (2018), the skip connections in U-Net can effectively fuse low-level features from the encoder part and high-level features from the decoder part, helping the network generate accurate pixel-level semantic segmentation of crack images. A U-shaped network called CrackU-Net was proposed by Ju et al. (2020), which performed better than U-Net in pixel-wise pavement crack detection. Zou et al. (2018) built DeepCrack, which fuses multi-scale convolutional features from hierarchical convolutional stages. Fan, Li, Chen, Wei, et al. (2020) proposed a modified U-Net called U-HDN for detecting pavement cracks, in which dilated convolution was integrated into hierarchical feature learning to improve the model's performance. F. Yang et al. (2020) created a crack detection network called FPHBN, which utilized a feature pyramid and hierarchical boosting module to integrate contextual information between the low- and high-level layers. Unlike networks that classify pixels in crack images using only the last layer of feature maps, DeepCrack, U-HDN, and FPHBN integrate feature maps at each scale to perform classifications and significantly prevent information loss. Although these specially designed encoder–decoder networks obtain satisfactory crack detection performance, they still have some drawbacks, such as a time-consuming training process and high computational cost, which hinder their application potential.

In recent years, attention mechanisms have become prominent in deep learning communities (Y. Pan & Zhang, 2022; Qu, Wang, et al., 2022). Hu et al. (2020) proposed a squeeze-and-excitation network (SENet) to fully utilize the information between feature channels, enhancing the

accuracy of image classification. Woo et al. (2018) presented a convolutional block attention module (CBAM) that combines spatial attention and channel attention mechanisms to improve the performance of the network. However, the attention mechanism proposed in the abovementioned studies was only used to process a single feature map as the input. Oktay et al. (2018) proposed an Attention U-Net model that can effectively integrate information from low- and high-level feature maps. They also embedded attention gate blocks into the U-Net architecture to improve the representation of features in regions of interest. Based on these studies, some researchers have incorporated attention mechanisms into the encoder–decoder architecture to enhance the performance of the network for crack detection (Sun et al., 2022; Kang & Cha, 2022). By integrating multi-feature fusion and attention mechanisms into an encoder–decoder backbone, Qu, Chen, et al. (2021) developed a novel variant of U-Net to detect cracks in pavement images. A global context block was used as the attention module to capture the global context information of the crack texture. The fact that the attention module was embedded only in the encoder introduced challenges to the fusion of multi-scale features and deep semantic information for identifying detailed information associated with cracks. Lin et al. (2021) proposed a full-attention U-Net for improving the segmentation accuracy of crack edges. By integrating the attention gate into each output of the backbone in the architecture to reduce noise at the crack edges, the network can improve the edge detection of steel cracks in large-scale images. To improve the detection accuracy of road cracks, J. Chen and He (2022) proposed a novel U-like encoding–decoding network with an attention gate embedded between the encoder and decoder. In short, the attention mechanism can facilitate the fusion of feature information from multiple layers to improve the crack detection accuracy of the network. Accordingly, in this work, a novel hybrid attention block (HAB) that merges channel and spatial attention was designed to more effectively fuse information from low- and high-level feature maps while reducing the parameter size.

Object detection has been widely investigated as one of the most important computer vision tasks. In the last decade, several computer vision-based approaches have been extensively applied to damage detection in civil structures such as crack detection (Celik et al., 2022; Kong et al., 2021; Chao Liu & Xu, 2022), distresses detection (Giglioni et al., 2023; E. Yang et al., 2022), skid resistance measurement (Chenglong Liu et al., 2023), and corrosion detection (Xu et al., 2020), and so forth. Among them, the detection of crack damages has received much attention from researchers. For example, Choi and Cha (2019) designed a semantic damage detection network for segmenting concrete cracks in real time. Based on deep learning, a network designed to detect cracks in underwater metallic surfaces was developed by F. C. Chen and Jahanshahi (2018). Chao Liu and Xu (2022) proposed a CNN based on VGGNet for detecting night pavement cracks. Y. Liu et al. (2019) combined high- and low-level feature maps to improve the network performance.

While many achievements have been made in proposing new models with improved accuracy of detecting cracks in pavements, the real-world deployment of these models in physical systems such as mobile robot platforms (J. Chen & He, 2022; Liao et al., 2022) or drones (Chu et al., 2022; Jang et al., 2021) is also very important. This calls for a lightweight model with sufficiently high accuracy to be designed and implemented. Although several studies have proposed lightweight models (Meng et al., 2023; Shim et al., 2021; Zhu et al., 2023), their performances generally suffer a deterioration, compared with their counterparts, or they cannot outperform other state-of-the-art models. Many other researchers have proposed numerous deep learning-based models for crack detection. Most of them contain a large number of model parameters, and their deployments heavily rely on expensive and high-end computing devices like graphics processing units (GPUs). However, the stability and reliability of these computing devices are only guaranteed in clean and controllable servers and data centers instead of wide and outdoor scenes, which are typical for crack detection. Therefore, although these models have achieved accurate segmentation performance, they are merely applied in practical on-site applications. To aid practical applications, edge computing devices (such as NVIDIA Jetson TX2, Jetson Nano, and Jetson Xavier NX) are widely used and possess favorable characteristics such as their small size, low power consumption, and portability. They can be mounted on unmanned aerial vehicles (UAVs) or robot platforms to detect cracks in various infrastructures such as high-rise buildings, bridges, tunnels, and roads. Recently, several models are proposed specifically for edge devices deployment. For example, J. Chen et al. (2023) designed a lightweight model called MCLD with a parameter size of 0.43 M. Jiang and Zhang (2020) proposed a method that used a wall-climbing UAV system to acquire crack images and then used a wireless data transmission method to meet real-time detection requirements. However, even though these models have been specifically designed for practical crack detection scenarios, they can only achieve the detection speed of 9.7 and 6 frames per second (FPS), respectively, which cannot meet the requirement for real-time applications.

In general, there are two main difficulties for practical crack detection applications. First, in the case of outdoor deployment scenarios, the crack detection machine

is usually deployed in wild and outdoor scenes with unpredictable interference, such as high environmental temperature and/or humidity, smoke, rain, or fog, which would significantly reduce the power and other operating performance of computing devices. Therefore, high-end devices like GPUs are not suitable, and thus researchers have started to focus on edge computing devices. Second, in the case of limited computing capabilities and resources on edge devices, existing lightweight crack detection models cannot achieve real-time performance on common edge devices, such as Jetson Xavier NX. Therefore, in order to successfully deploy models for real-time crack detection in extreme environments or in new applications such as robot platforms and UAVs with resource-constrained embedded devices, it is crucial and may become a trend for the research community to design models with smaller parameter size and faster inference speed while maintaining high detection accuracy. In this study, a novel model with a computational cost lower than the published models and a detection performance better than other state-of-the-art ones is proposed for automatic pavement crack detection at the pixel level.

In summary, this work makes three primary contributions.

1. This study proposes a novel crack detection model called RHACrackNet, which has the lowest computational cost that can be deployed in a robot platform for crack detection and a detection performance that is superior to other state-of-the-art models. Besides, RHACrackNet is specifically designed for deployment in physical systems like robot platforms and UAVs, which are merely considered in existing works.
2. A novel HAB is proposed for effective feature extraction on crack areas, which enables RHACrackNet to remove redundant feature channels for smaller model parameters while maintaining high detection accuracy. In addition, by adopting depth-wise separable convolutions (DS_Convs), a lightweight version of the proposed network, RHACrackNet* was created with a comparable performance.
3. A robot system with an omnidirectional mobile platform and a six-degree-of-freedom manipulator is designed and developed for crack detection. A UAV can be installed on the mobile platform to enable more advanced applications in the future. The camera installed at the end effector of the manipulator can achieve real-time detection of pavement cracks at a speed of 25 FPS by utilizing the lightweight version of the proposed RHACrackNet model.

The rest of this paper is organized as follows. In Section 2, the architecture and configuration of the proposed network are explained. In Section 3, details of the experiments are provided. The experimental results are provided and analyzed in Section 4. Finally, the contribution and the limitations of this study are presented in Section 5.

2 | METHODOLOGY

In the down-sampling stage of encoder–decoder structure, in order to extract more abstract global features, the image size will be continuously reduced under the effect of the convolution layer. To prevent the network from losing excessive feature information during this process, the encoder–decoder network generally sets a large number of channels. For example, the number of channels in U-Net is as high as 512, which leads to complex network calculations and large parameter sizes; thus, it is not suitable for deployment on small computing devices. To significantly reduce the parameter size of the network, the proposed method reduces the number of channels in each layer, and the maximum number of channels is only a quarter of U-Net. However, this generates two problems: (1) more detailed information may be lost, and (2) the feature extraction ability of the network decreases. To remedy this, the number of network layers can be increased, and the extraction ability can be improved through additional feature extractions and fusions. However, this will result in more parameters and operations. Therefore, a novel network structure is proposed by adding residual blocks (RBs) to the deepest layer of the encoder and using the learnable characteristics of the residual structure to avoid the operation expansion by simply increasing the number of invalid network layers. As a result, our final parameters are only 1/20 of the number of attention U-Net and even less than 1/40 after using depth-wise convolution. At the same time, the crack detection performance has reached the state of the art in both public datasets and our collected dataset.

2.1 | Network architecture

The structure of RHACrackNet is depicted in Figure 1, which consists primarily of an encoder network, a RBs, five HABs, a decoder network, and a pixel-wise classification layer for generating prediction results. The encoder network contains an initial convolution block and four encoder blocks. The input original image is first processed by the initial convolution block containing a 3×3 convolution operation and a rectified linear unit (ReLU) and then goes through four encode blocks.

Each encoder block is an operation combination of a down-sampling procedure, two 3×3 convolutions, and a ReLU in sequence, which will reduce the size of its input

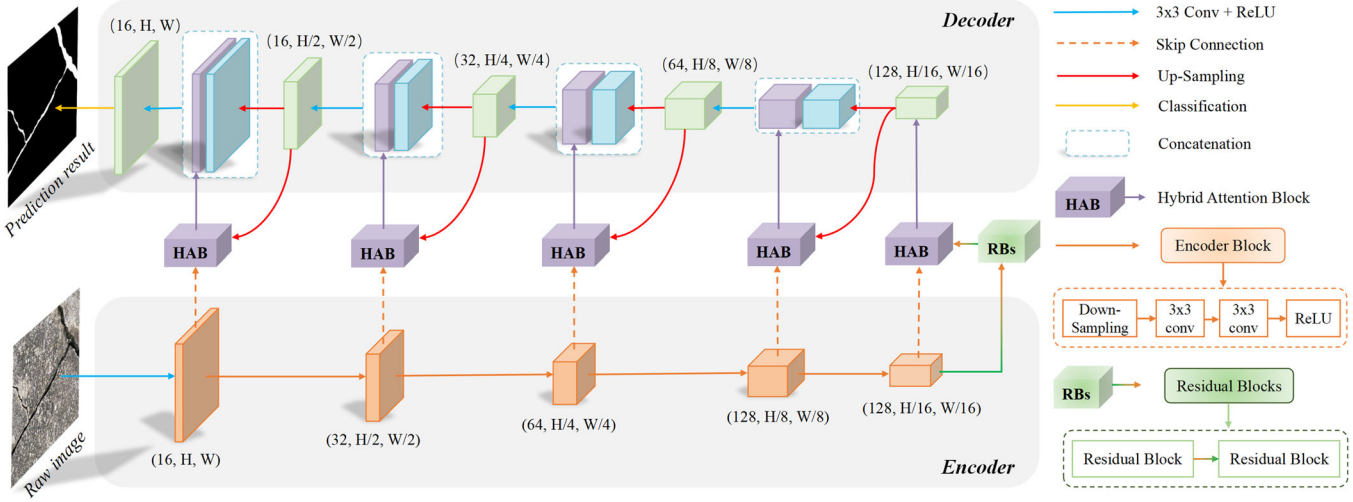

FIGURE 1 Architecture of the proposed RHACrackNet.

feature map by a quarter and double the channels of the feature map. In the deepest layer of the RHACrackNet, the RBs are used to deepen the network, that is, to augment the number of network layers and avoid network degradation. The decoder network is designed according to the encoder network. Each decoder block consists of a combination of corresponding operations of a bilinear up-sampling and a 3 × 3 convolution operation. Combining the output results of the decoder and the corresponding encoder can effectively enhance the ability of the network to characterize image details. To achieve a better combination, a HAB is carefully designed by integrating channel and spatial attention. The HABs are embedded into the encoder–decoder architecture such that the outputs from the corresponding pairs of the encoder and decoder are combined in an optimal way. The result of the last decoder block is input to a soft-max classifier to obtain the prediction result.

2.2 | HAB

Since the encoder–decoder architecture produces hierarchical features (i.e., low-level features $\boldsymbol{F_l}$ and high-level features $\boldsymbol{F_h}$), HABs are designed by fusing $\boldsymbol{F_l}$ (from encoder blocks), $\boldsymbol{F_h}$ (from RBs or up-sampling operations), and channel and spatial attention shown in Figure 2.

In general, the information presented in the low-level features focuses more on the location details of the cracks, and the high-level features encompass more semantic information. Inspired by CBAM, the HABs incorporating channel and spatial attention mechanisms are used to merge low- and high-level features. Channel attention tells us which channels' outputs should be more emphasized, while spatial attention specifies which positions' outputs should be more valued. Specifically, channel attention deals with 'what' is significant for the input image, and spatial attention copes with "where" is substantial for the input image.

On the channel part, inspired by SENet, adaptive average pooling, full connection layer, and activation function are utilized to gain the channel attention graph that enables the network to present the importance of different channels. In Figure 2, since $\boldsymbol{F_l}$ and $\boldsymbol{F_h}$ have the same size, their feature maps can be added pixel-by-pixel to obtain an overlapping map. Thereafter, an adaptive average pooling operation is utilized to squeeze the overlapping map into a map of the channel weight and apply 1 × 1 convolution to extract the relationship among different channels. This operation converts every two-dimensional feature map of each channel into a weight number with a global receptive field, which means much channel internal information is lost, and this map of the channel weights only has preliminary channel attention information. Since the low- and high-level features have different channel attention information, they are extracted and fused to enhance the representation of channel information in the preliminary channel attention map $\boldsymbol{M_p}$. The low-level features generated by the encoder block pass through fewer convolution operations, which implies that the low-level feature map $\boldsymbol{M_l}$ squeezed by adaptive average pooling and 1 × 1 convolution operation has less semantic information. In contrast, the high-level feature map $\boldsymbol{M_h}$ obtained by adaptive average pooling and 1 × 1 convolution on $\boldsymbol{F_h}$ contains more semantic information. Therefore, $\boldsymbol{M_p}$ is multiplied with $\boldsymbol{M_l}$ to downplay insignificant channels and highlight the important ones, then the result is fused with $\boldsymbol{M_h}$ by element-wise summation. The above operations enable the channel information of $\boldsymbol{F_l}$ and $\boldsymbol{F_h}$ to be more fully utilized. Subsequently, the channel attention map $\boldsymbol{M_c}$ is obtained after average pooling, 1 × 1 convolution, and softmax

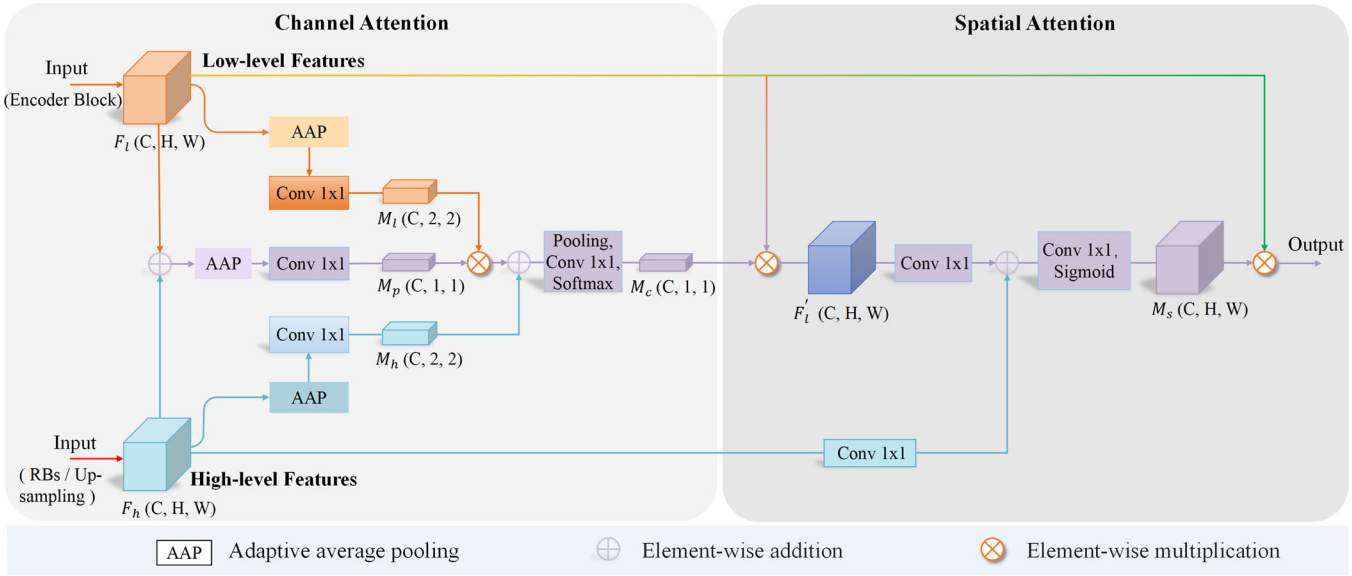

FIGURE 2 Schematic of the proposed hybrid attention block (HAB).

operations in sequence. Briefly, $\boldsymbol{M_c}$ is computed as follows:

$$\begin{aligned}\boldsymbol{M_c} &= f_c(\boldsymbol{F_l}, \boldsymbol{F_h}) = Soft\max(\sigma(Avg\,Pool(\sigma(AAP(\boldsymbol{F_h} \oplus \boldsymbol{F_l}))\\ &\otimes \sigma(AAP(\boldsymbol{F_l})) \oplus \sigma(AAP(\boldsymbol{F_h}))))\\ &= Soft\max(Avg\,Pool(\boldsymbol{M_p} \otimes \boldsymbol{M_l} \oplus \boldsymbol{M_h})) \end{aligned} \quad (1)$$

where $f_c(\boldsymbol{F_l}, \boldsymbol{F_h})$ is the calculation function of the channel attention map, σ represents a 1×1 convolution operation and a ReLU, *AAP* represents an adaptive average pooling operation, *AvgPool* represents an average pooling operation, *Softmax* represents a softmax operation, $\oplus$ and $\otimes$ are the element-wise summation and multiplication, respectively. The final output of the channel part $\boldsymbol{F_l'}$ is obtained by multiplying the channel attention map $\boldsymbol{M_c}$ with the low-level features $\boldsymbol{F_l}$.

On the spatial part, extracting the region of interest from the feature map facilitates filtering out the influence of the background and improves the spatial representational ability of the network. Element-wise summation is applied to fuse the $\boldsymbol{F_l'}$ and $\boldsymbol{F_h}$, then a 1×1 convolution operation and sigmoid function are used to extract the spatial area of interest and generate the corresponding spatial attention map $\boldsymbol{M_s}$. In short, $\boldsymbol{M_s}$ is computed as follows:

$$\boldsymbol{M_s} = fs(\boldsymbol{F_l'}, \boldsymbol{F_h}) = Sigmoid(\sigma(\boldsymbol{F_l'}) \oplus \sigma(\boldsymbol{F_h})) \quad (2)$$

where $f_s(\boldsymbol{F_l'}, \boldsymbol{F_h})$ is the calculation function of the spatial attention map, *Sigmoid* represents a sigmoid function. Finally, the output of the HAB fusing $\boldsymbol{F_l}$ and $\boldsymbol{F_h}$ is obtained by multiplying $\boldsymbol{M_s}$ and $\boldsymbol{F_l}$, which is calculated as follows:

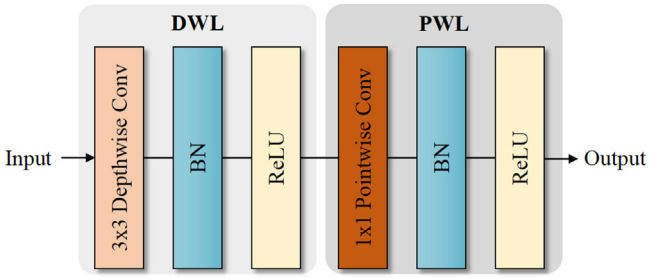

FIGURE 3 Depth-wise separable convolution operation. DWL and PWL denote the depth-wise convolution layer and point-wise convolution layer, respectively.

$$f_{hybrid}(\boldsymbol{F_l}, \boldsymbol{F_h}) = f_s(\boldsymbol{F_l} \otimes f_c(\boldsymbol{F_l}, \boldsymbol{F_h}), \boldsymbol{F_h}) \otimes \boldsymbol{F_l} \quad (3)$$

2.3 | The lightweight version of RHACrackNet

In this study, the following points need to be considered when designing a lightweight crack detection model. First, the lightweight model needs to have a very high inference speed. Second, it needs to have a high detection accuracy. Hence, by reducing the number of model channels and fusing the RBs and the hybrid attention modules, we attempt to enhance the inference speed of the model while improving the detection accuracy of the model. Additionally, to facilitate the deployment of the proposed network on the embedded platform, all conventional convolutions are replaced with DS_Convs, which can further improve the running speed of the network by reducing its parameters. Figure 3 illustrates the configuration of DS_Conv

operation, which contains a depth-wise and point-wise convolution layer (i.e., DWL and PWL). Specifically, the DWL is composed of a 3×3 depth-wise convolution, batch normalization (BN), and ReLU in sequence. The PWL contains a 1×1 point-wise convolution, BN, and ReLU in sequence. In conventional convolution, each convolution kernel has to convolve all channels, while depth-wise convolution only allows each convolution kernel to convolve one channel. This implies that the amount of computation and parameters of the convolution operations are significantly reduced.

However, if only depth-wise convolution is used, the information between channels is not utilized. The PWL fuses the feature maps of all channels with a 1×1 convolution kernel.

2.4 | Loss function

Pixel-wise detection of pavement cracks is a binary classification task that aims to separate cracked pixels from non-cracked ones in crack images. Typically, cracked pixels represent a smaller proportion of pixels in a crack image dataset, compared with non-cracked ones, meaning that the dataset is significantly unbalanced. To solve this problem, during the training phase, the weighted binary cross-entropy loss function is employed to train the model with the following equation:

$$L_{wbce} = -\sum_{i=1}^{N} \left(w_p y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right) \tag{4}$$

$$w_p = \frac{(P_c + P_n)}{\alpha * P_c} \tag{5}$$

where N, y, $\hat{y}$, and w_p denote the total number of samples, model prediction, ground truth, and balance factor between crack and non-crack samples, respectively. P_c represents the number of pixels occupied by the crack area in each image in the dataset, P_n represents the number of pixels occupied by the background area in each image in the dataset, and α is a hyperparameter, representing the magnification of the loss weight of the positive sample used to alleviate the imbalance of the samples, which in the experiment is set to 3.

2.5 | EXPERIMENTS

A self-built crack dataset (CamCrack789) and three other public datasets are first presented in this section. Then the experimental settings and evaluation metrics are explained.

2.5.1 | Datasets

In this paper, four datasets, namely, CamCrack789, Crack500 (F. Yang et al., 2020), Crack Forest Dataset (CFD; Shi et al., 2016), and DeepCrack237, are used to verify the effectiveness of the proposed algorithm. As shown in Table 1, there are five major dimensions to differentiate them. According to the width of cracks, the crack images in the four datasets are classified into four types, that is, thin crack images, medium crack images, thick crack images, and extremely thick crack ones. Among the four datasets, it can be seen that CamCrack789 contains mainly thin and medium crack images; almost all images in CFD are thin crack images; Crack500 contains four types of crack images, but the proportion of thick and extremely thick crack images is larger. DeepCrack237 also contains four types of crack images, but their proportion is not significantly different. Figure 4 shows some samples and corresponding ground truth from the four datasets.

2.5.2 | CamCrack789

Figure 5b shows that the images of this dataset are taken from different locations on the campus of Shantou University. These images are taken by a Microsoft HD camera mounted on the end of the manipulator of a self-designed mobile robot (as shown in Figure 5a), capable of working in various weather situations, such as sunny, cloudy, and rainy conditions. The camera can record 2304×1728 pixels of video at 30 FPS. In total, 789 images of pavement cracks are selected to construct a pavement crack dataset named CamCrack789. They all have a resolution of 640×480. The cracks are classified into four types based on their topology: common, intersecting, block, and alligator cracks. The crack images are collected under different lighting conditions, such as dark and bright environments. Additionally, they contain various backgrounds, including water stains, shadows, leaves, pavement markings, oil stains, debris, and so on. Furthermore, some blurred crack images caused by the movement of the mobile robot or poor lighting conditions are included. In CamCrack789, every image has its ground truth labeled at the pixel level by hand. Figure 6 provides some examples of raw images and their ground truth. This dataset is randomly separated into 546 training images and 243 testing ones.

To enhance the generalizability of the model and prevent overfitting, the number and diversity of training images are increased by using data augmentation. The

TABLE 1 The major differences of the datasets.

Differences	CamCrack789	Crack500	CFD	DeepCrack237
Sample size	789 images	3368 images	118 images	237 images
Resolution	640 * 480	256 * 256	480 * 320	544 * 384
The way of use in this work	For training and evaluating the method	For training and evaluating the method	For training and evaluating the method	To prove the generalization of the methods, this dataset is only used for evaluating the method
Scenario and background interference	The images contain water stains, oil stains, leaves, branches, pavement markings, shadows, occlusion, soil, and other debris. (Some examples of this dataset are shown in Figure 6)	It contains some crack images with shadows, occlusions, and low contrast	The images have some complex circumstances, including shadows, water leakages, stains, and lane lines	It contains crack images with multiple textures and scenes
The crack width	Mostly thin and medium	Mostly thick and extremely thick	Mostly thin	More diverse including four major types: thin, medium, thick, and extremely thick

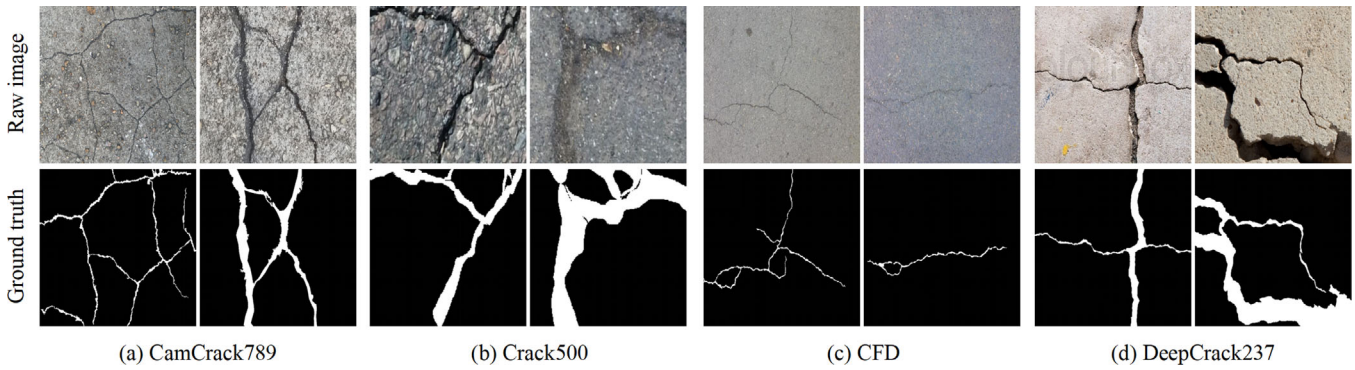

FIGURE 4 Crack samples and their ground truth from the four datasets. The resolution of these images is 512 × 512.

FIGURE 5 Robot and images of pavement cracks taken from different locations. (a) Robot for road crack image acquisition and detection and (b) concrete crack images collected from different locations.

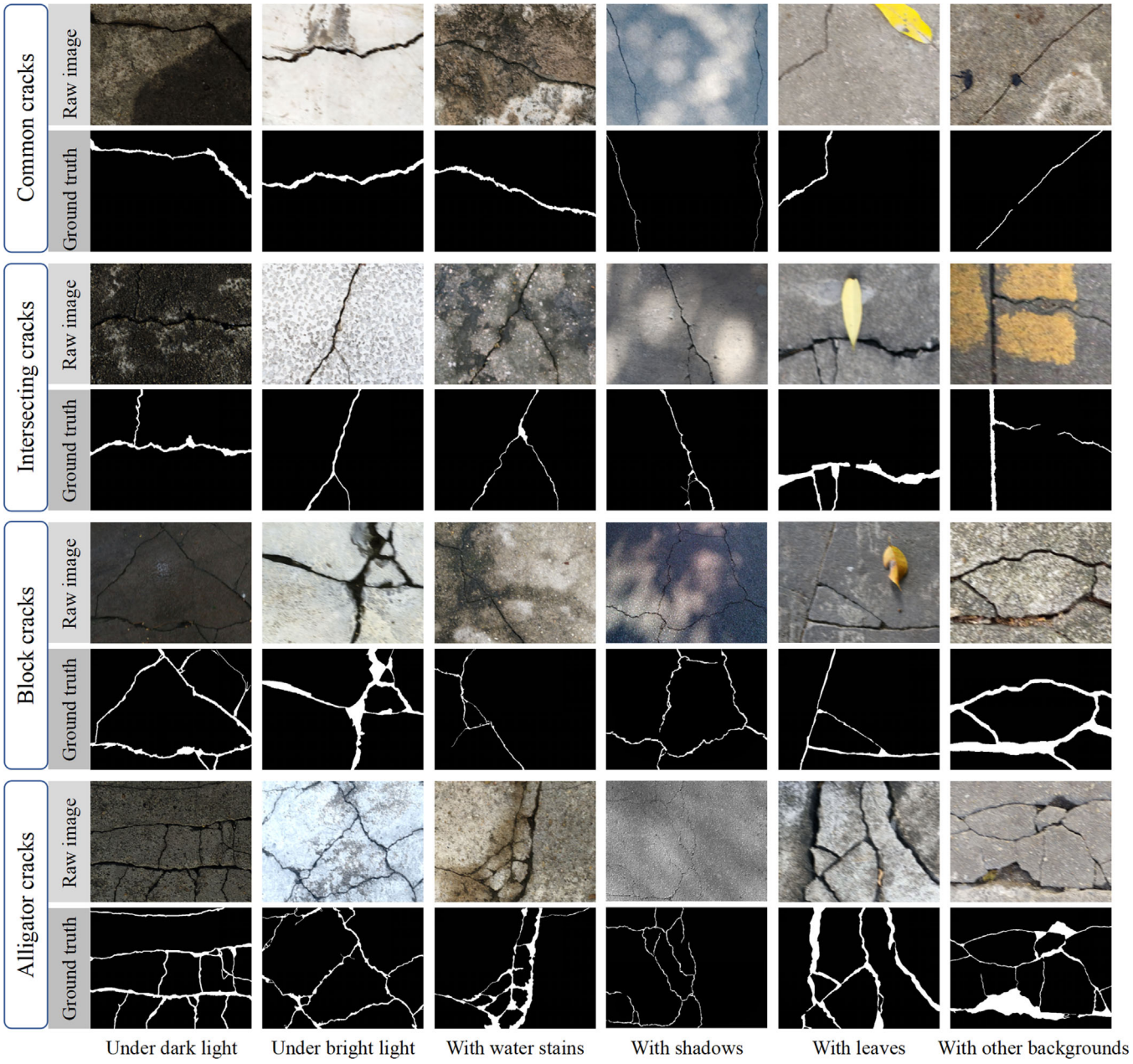

FIGURE 6 Examples of the CamCrack789 dataset.

crack images are randomly rotated by 180°, flipped vertically and horizontally, and have their brightness and contrast changed.

2.5.3 | Crack500

There are 3368 images in the Crack500 dataset from 500 original crack images with around 2000 × 1500 pixels captured by cell phones. This dataset contains some crack images with shadows, occlusions, and low contrast. Each original image is manually labeled at the pixel level. To prompt the image processing speed and reduce the consumption of computation resources, each image is resized to 256 × 256 pixels. In Crack500, the images are divided into three types: training images (1896), validation images (348), and testing images (1124). This dataset is utilized to train and evaluate the methods.

2.5.4 | CFD

There are 118 RGB images in the CFD with a fixed size of 480 × 320 pixels. These images were captured by an

iPhone5 in Beijing, China, under some complex circumstances, including shadows, water leakages, stains, and lane lines. In CFD, all ground truth images are annotated manually at the pixel level. These images are separated randomly into training images and test images in the ratio of 6:4. This dataset is used for training and evaluating the algorithms.

2.5.5 | DeepCrack237

This dataset is the testing set of the DeepCrack (Y. Liu et al., 2019) dataset, which contains 237 RGB color images manually annotated segmentations. The image resolution is 544 × 384 pixels. This dataset contains crack images with multiple textures, scenes, and scales making the detection task challenging. In this study, DeepCrack237 is used to evaluate the generalization ability and robustness of the model trained on the CamCrack789 dataset.

2.6 | Experimental settings

In this work, the PyTorch library is utilized to implement the proposed method. The specifications of the computing platform used to conduct training and test the network are as follows: Intel(R) Xeno(R) Gold 5115 2.40 GHz CPU, NVIDIA 11G GeForce RTX 2080 Ti. In the training phase, Adam (Kingma & Ba, 2014) is adopted as an optimizer assigned to a 1e-3 learning rate. The batch size is given to 8 for CamCrack789 and Crack500, 12 for CFD, and 4 for DeepCrack237, and the models are saved every 50 epochs for a total of 500 training epochs.

2.7 | Evaluation metrics

Detecting pavement cracks at the pixel level is a binary classification problem, which means that pixels in the crack image need to be classified as either cracks or non-cracks. The prediction output of the network is a classification probability map that assigns a probability for each pixel to belong to a crack. In this work, the following three metrics are adopted for evaluating the model: precision (*Pr*), recall (*Re*), and *F*1 score (*F*1). *F*1 is the harmonic average of *Pr* and *Re*, which is commonly used for evaluating a model's comprehensive performance. The evaluation metrics are given as follows:

$$Pr = TP/(TP + FP) \tag{6}$$

$$Re = TP/(TP + FN \tag{7}$$

$$F1 = 2 \times Pr \times Re/(Pr + Re) \tag{8}$$

where *TP*, *TN*, *FP*, and *FN* are the numbers of true positive, true negative, false positive, and false negative, respectively. For crack detection research, considering the crack width, a small distance (2 pixels in Amhaz et al., 2016; X. Zhang, et al., 2019) between the prediction outcome and the label is allowed in evaluation, which is adopted in this study as well.

3 | EXPERIMENTAL RESULTS AND DISCUSSIONS

3.1 | Comparative experiments and discussions

The proposed network was built on a U-like encoder–decoder architecture and compared to seven methods, including FCN, U-Net, Attention U-Net, DC_Zou, DC_Liu, DMA-Net (Sun et al., 2022), and AttentionCrackNet (J. Chen & He, 2022). In this work, DC_Zou and DC_Liu denote the models of DeepCrack (Zou et al., 2018) and DeepCrack (Y. Liu et al., 2019), respectively. For a fair comparison, all these networks are trained with the same hyperparameters. In the following subsections, the evaluation results of the proposed method on CamCrack789, Crack500, CFD, and DeepCrack237 are given and compared with those of existing methods.

3.1.1 | Results on CamCrack789

Table 2 shows that the proposed RHACrackNet gets the best results with *F*1 94.94% and *Re* 95.04%. Compared with U-Net, and DC_Liu, the performance improvement on *F*1 made in RHACrackNet are 1.19%, and 6.51%. Note that the lightweight version of RHACrackNet is only 0.14% lower than RHACrackNet on *F*1. From Figure 7a, it can be observed that RHACrackNet has the best precision and recall values. Furthermore, in visualization results shown in Figure 8, the detection accuracy of RHACrackNet is the best, which can more accurately identify four different geometrical characteristics of cracks (common, intersecting, block, and alligator cracks) and suppress the background noise. Notably, the lightweight version RHACrackNet* shows a very similar performance to RHACrackNet.

3.1.2 | Results on Crack500

Table 3 shows that RHACrackNet outperforms the other compared methods on Crack500. Compared with

TABLE 2 Experimental results on the CamCrack789 dataset.

Methods	Precision (*Pr*)	Recall (*Re*)	F1 score (*F*1)	Floating-point operations per second (FLOPs; G)	Time/image (s)
FCN [CVPR 2015]	0.9445	0.9440	0.9443	290.46	0.094
U-Net [MICCAI 2015]	0.9349	0.9401	0.9375	375.24	0.111
Attention U-Net [MIDL 2018]	0.9470	0.9495	0.9482	624.72	0.199
DC_Zou [TIP 2018]	0.9660	0.8455	0.9017	1283.64	0.361
DC_Liu [Neurocomputing 2019]	**0.9669**	0.8146	0.8843	188.56	0.058
DMA-Net [TITS 2022]	0.9377	0.9463	0.9420	212.12	0.068
AttentionCrackNet [CACAIE 2022]	0.9386	0.9450	0.9418	329.02	0.104
RHACrackNet	0.9483	**0.9504**	**0.9494**	21.60	0.033
RHACrackNet*	0.9494	0.9465	0.9480	**9.68**	**0.032**

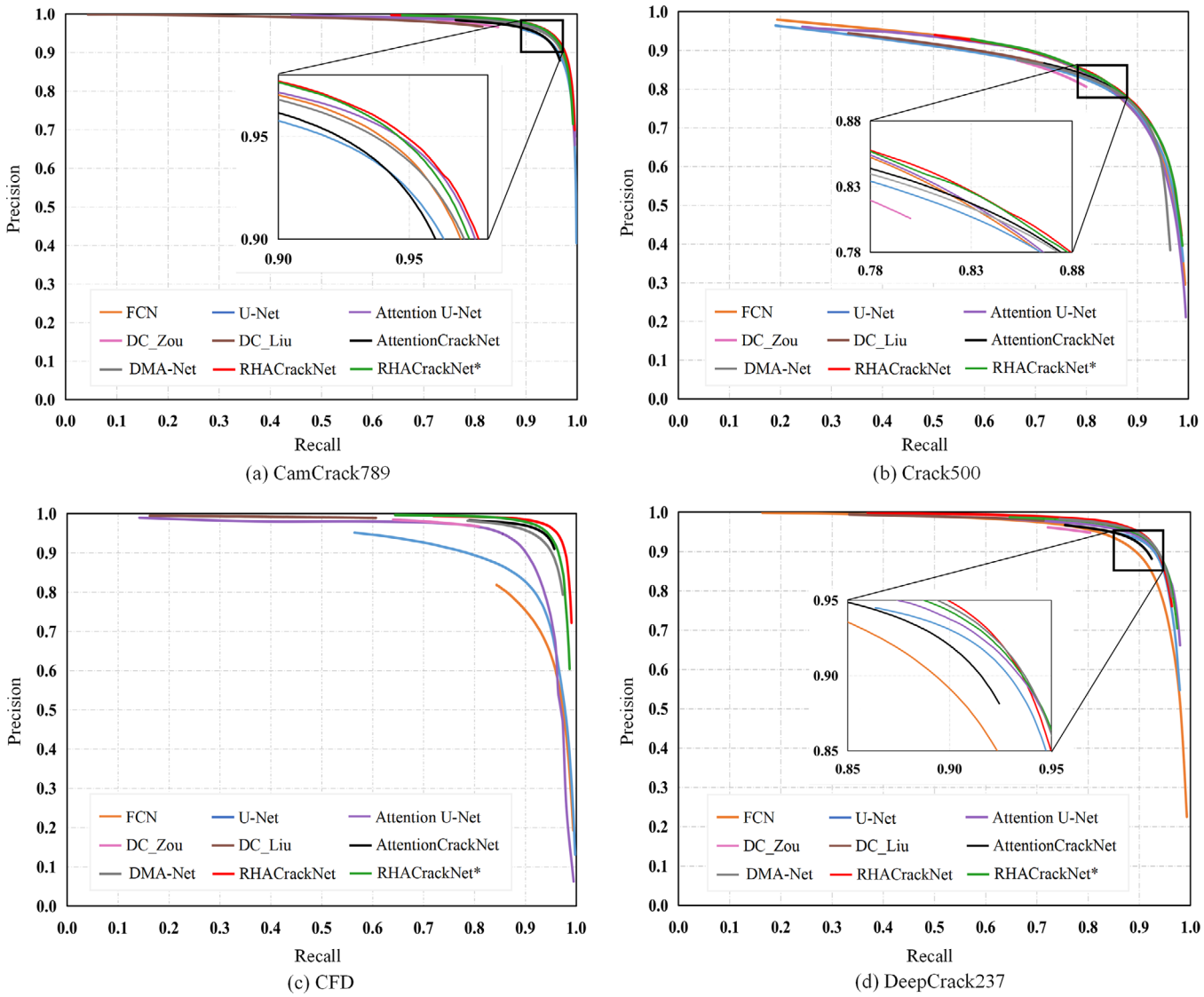

FIGURE 7 Precision-Recall curves of compared methods on CamCrack789, Crack500, CFD, and DeepCrack237 datasets.

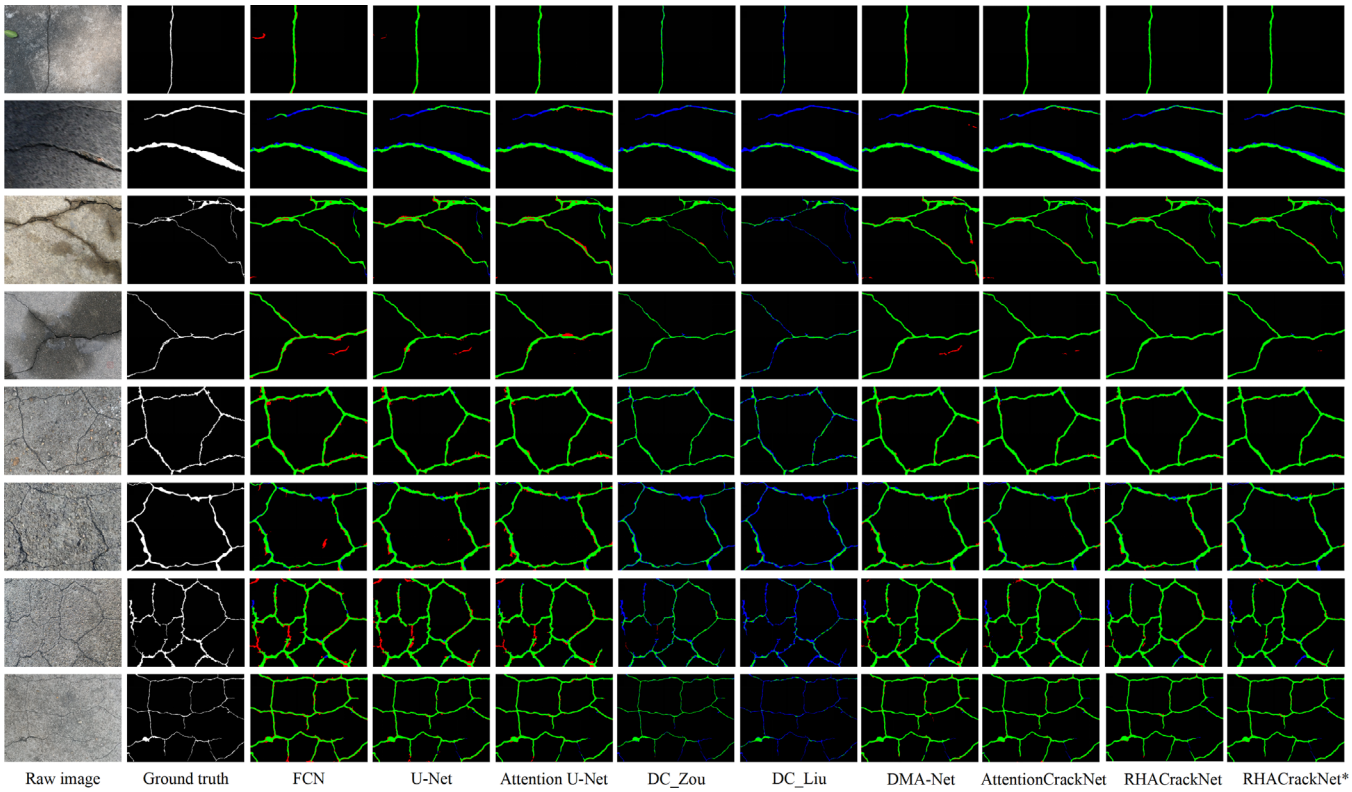

FIGURE 8 Visualization results of different models from the CamCrack789 dataset. The green, red, and blue pixels in the images represent true positives, false positives, and false negatives, respectively.

TABLE 3 Experimental results on Crack500.

Methods	*Pr*	*Re*	*F*1	FLOPs (G)	Time/image (s)
FCN [CVPR 2015]	0.8067	0.8385	0.8223	61.96	0.023
U-Net [MICCAI 2015]	0.7984	0.8464	0.8203	80.06	0.025
Attention U-Net [MIDL 2018]	0.8129	0.8345	0.8235	133.28	0.045
DC_Zou [TIP 2018]	0.8058	0.7999	0.8028	273.84	0.082
DC_Liu [Neurocomputing 2019]	**0.8542**	0.7478	0.7975	40.22	0.015
DMA-Net [TITS 2022]	0.7426	**0.9183**	0.8204	45.23	0.018
AttentionCrackNet [CACAIE 2022]	0.7794	0.8570	0.8006	70.2	0.024
RHACrackNet	0.8061	0.8542	**0.8295**	4.60	**0.011**
RHACrackNet*	0.8173	0.8404	0.8287	**2.06**	**0.011**

U-Net, and DC_Liu, it has 0.92%, and 3.20% performance improvement on *F*1. Moreover, from Figures 7b and 9a, it is observed that RHACrackNet* obtains a similar performance to that of RHACrackNet. In the first row of Figure 9a, although all methods fail to inspect the full shape of the cracks, RHACrackNet model and its lightweight version outperform the other methods. From the second row of Figure 9a, compared with RHACrackNet and RHACrackNet*, FCN and U-Net misidentify more background pixels at the crack edge as crack pixels, showing poorer detection results. DC_Zou fails to identify some cracks, while DC_Liu almost misses all of them.

3.1.3 | Results on CFD

As illustrated in Figure 7c, RHACrackNet outperforms other models on this dataset, and the lightweight version RHACrackNet* obtains a performance similar to that of RHACrackNet. Table 4 shows that RHACrackNet significantly outperforms the other comparative

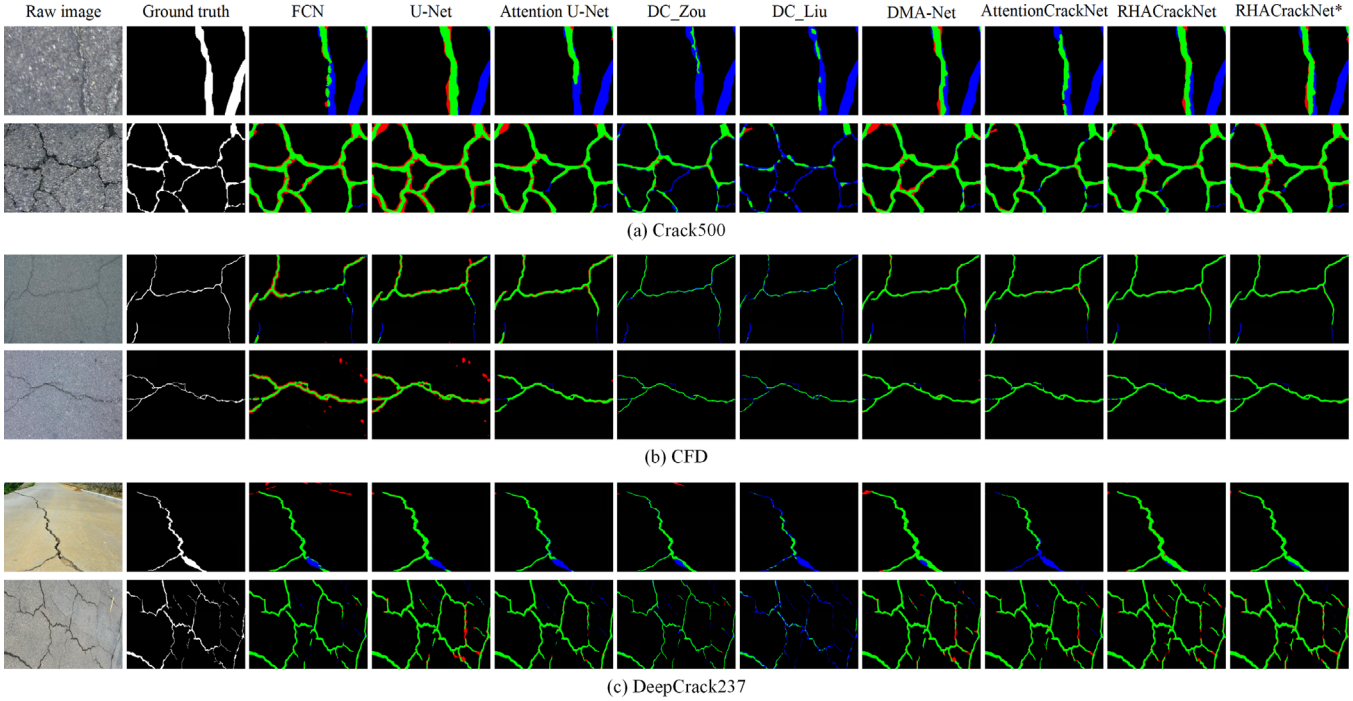

FIGURE 9 Visualization results of different models from Crack500, CFD, and DeepCrack237.

TABLE 4 Experimental results on CFD.

Methods	*Pr*	*Re*	*F*1	FLOPs (G)	Time/image (s)
FCN [CVPR 2015]	0.8118	0.8510	0.8309	145.22	0.064
U-Net [MICCAI 2015]	0.8463	0.8809	0.8633	187.62	0.068
Attention U-Net [MIDL 2018]	0.9253	0.8844	0.9044	312.36	0.117
DC_Zou [TIP 2018]	0.9674	0.8088	0.8810	641.82	0.200
DC_Liu [Neurocomputing 2019]	**0.9890**	0.6065	0.7519	94.28	0.047
DMA-Net [TITS 2022]	0.9171	0.9453	0.9310	106.06	0.053
AttentionCrackNet [CACAIE 2022]	0.9430	0.9408	0.9399	164.52	0.066
RHACrackNet	0.9629	**0.9520**	**0.9574**	10.80	**0.030**
RHACrackNet*	0.9595	0.9359	0.9476	**4.84**	**0.030**

models in *Re* and *F*1. The *F*1/*Re* values of FCN, U-Net, Attention U-Net, DC_Zou, DC_Liu, DMA-Net, and AttentionCrackNet are 12.65%/10.10%, 9.41%/7.11%, 5.30%/6.76%, 7.64%/14.32%, 20.55%/34.55%, 2.64%/0.67%, and 1.75%/1.12% lower than that of RHACrackNet, respectively. From Figure 9b, the visualization results depict that the prediction results of Attention U-Net, DMA-Net, AttentionCrackNet, RHACrackNet, and RHACrackNet* have a more complete shape than the other four methods and are closer to the ground truth. The main reason for this is probably due to the attention mechanism used by all five algorithms. In addition, some segmentation details of an exemplar image with tiny cracks are shown in Figure 10, in which images in the second and third rows are magnified views of the red and yellow rectangular boxes in the top row images, respectively. It can be seen that RHACrackNet has better performance than other models in detecting tiny cracks.

3.1.4 | Results on DeepCrack237

From Figure 7d and 9c, the proposed RHACrackNet outperforms the other models in comparison to the DeepCrack237 dataset. As shown in Figure 9, the cracks detected by RHACrackNet and RHACrackNet* have better continuity than the other methods. For complicated cracks like the second row in Figure 9c, RHACrackNet shows

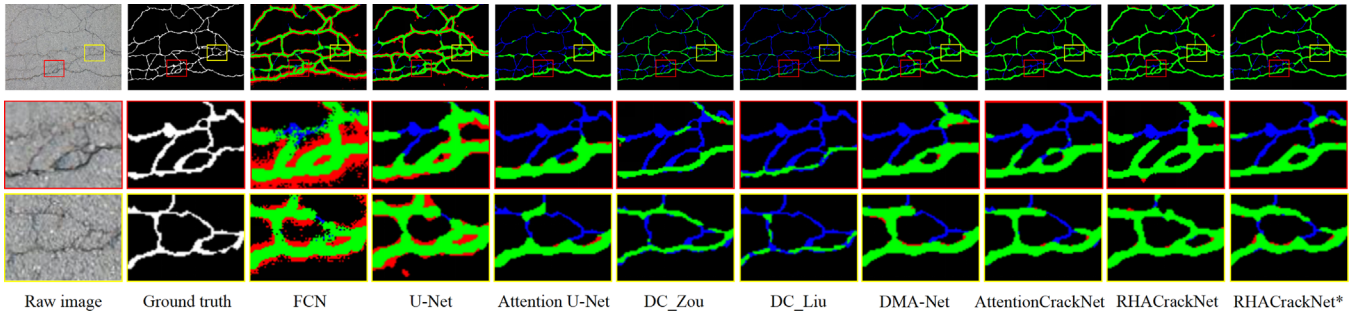

FIGURE 10 A detailed demonstration of the advantages of the proposed method in segmenting tiny cracks.

TABLE 5 Results of compared methods test on DeepCrack237 dataset.

Methods	*Pr*	*Re*	*F*1	FLOPs (G)	Time/image (s)
FCN [CVPR 2015]	0.9049	0.8886	0.8967	197.50	0.152
U-Net [MICCAI 2015]	0.9206	0.9130	0.9168	255.16	0.178
Attention U-Net [MIDL' 2018]	0.9284	0.9121	0.9202	424.80	0.317
DC_Zou [TIP 2018]	0.9481	0.8041	0.8702	872.88	0.585
DC_Liu [Neurocomputing 2019]	**0.9790**	0.7131	0.8252	128.22	0.107
DMA-Net [TITS 2022]	0.9421	0.9072	0.9241	144.24	0.127
AttentionCrackNet [CACAIE 2022]	0.9079	0.9052	0.8899	223.74	0.164
RHACrackNet	0.9364	**0.9141**	**0.9251**	14.68	**0.083**
RHACrackNet*	0.9312	0.9139	0.9226	**6.58**	**0.080**

the best performance in detecting thin cracks, and the performance of RHACrackNet* is comparable to that of RHACrackNet. From Table 5, it can be seen that both RHACrackNet and RHACrackNet* are better than Attention U-Net in all metrics. Compared to FCN, U-Net, Attention U-Net, DC_Zou, DC_Liu, DMA-Net, and AttentionCrackNet, there are 2.84%, 0.83%, 0.49%, 5.49%, 9.99%, 0.10%, and 3.52% of improvement by RHACrackNet, respectively.

Furthermore, the floating-point operations per second (FLOPs) and processing time of the models are analyzed on all datasets, and the results are given in Tables 2–5. As shown in these tables, the proposed RHACrackNet achieves the best performance, and its FLOPs are much fewer than other compared models except for RHACrackNet*. RHACrackNet* is not only the fastest (more than two times faster than U-Net) among all the models but also obtains a similar performance similar to that of RHACrackNet with fewer FLOPs on four datasets. In addition, it is noted that Attention U-Net, DMA-Net, AttentionCrackNet, RHACrackNet, and RHACrackNet* as a group obtain better performance than the other methods. One possible reason for this is that they all employ attention mechanisms. From the visualization results on the four datasets, due to integrating the spatial and channel attention mechanisms, the proposed method focuses more on the crack region and better suppresses the background interference than the Attention U-Net and AttentionCrackNet, which only use the attention gate.

3.2 | Results of special cases

Even though the proposed model can achieve very satisfactory detection results in most images, it does fail in very difficult cases. Actually, no one model can guarantee to successfully detect cracks in all images. In particular, it should be noted that when the data distribution in the image samples is quite different from those of the training set, it is more likely to cause the proposed method to fail. As illustrated in Figure 11, some skid-proof stripes were falsely identified as cracks due to significant similarities between skid-proof stripes and cracks in terms of texture features. It should also be noted that some edges of the maintenance hole cover were also detected as cracks as shown in Figure 11. It is recommended to collect more crack images with skid-proof stripes and maintenance hole covers in different circumstances to expand the dataset for training to enhance the performance of the proposed method, among other methods to improve the model.

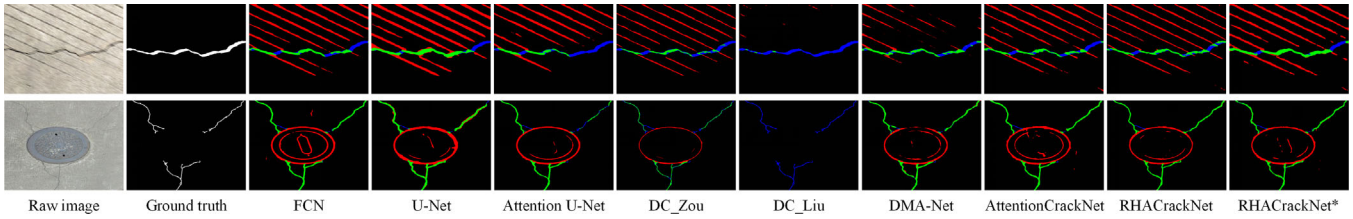

FIGURE 11 Two failure cases, including skid-proof stripes and maintenance hole cover interference.

TABLE 6 Results of comparison with different attention mechanisms on CamCrack789 and CFD.

Models	CamCrack789			CFD		
	Pr	***Re***	***F*1**	***Pr***	***Re***	***F*1**
SE	0.9139	0.9658	0.9366	0.9374	0.9359	0.9342
CBAM	0.9180	0.9621	0.9367	0.9429	0.9231	0.9303
ECA	0.9223	0.9640	0.9405	0.9269	0.9208	0.9161
DA	0.8835	**0.9784**	0.9258	0.9187	0.9107	0.9103
HAB	**0.9483**	0.9504	**0.9494**	**0.9629**	**0.9520**	**0.9574**

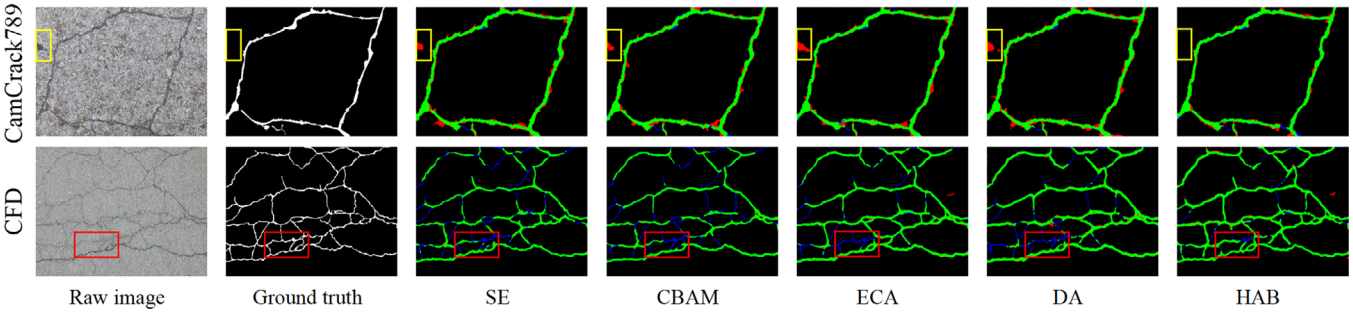

FIGURE 12 Visualization results of methods of different attention modules.

TABLE 7 Results of ablation experiments on CamCrack789.

Models	***Pr***	***Re***	***F*1**
Baseline	0.9155	0.9585	0.9323
Baseline+RBs	0.9256	0.9594	0.9393
Baseline+HAB	0.9298	**0.9601**	0.9426
Baseline+RBs+HAB	**0.9483**	0.9504	**0.9494**

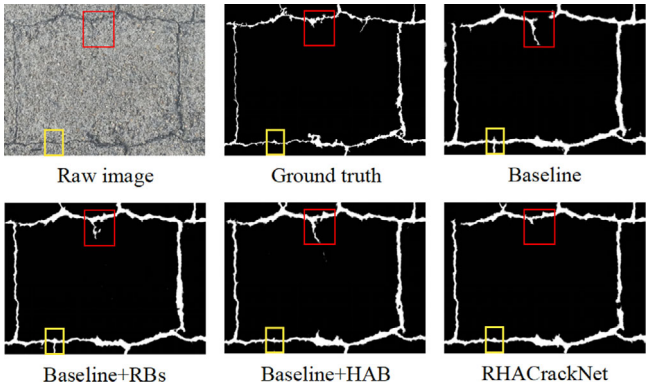

FIGURE 13 Visualization results of ablation experiment.

3.3 | Results with different attention modules

To demonstrate the effectiveness of the proposed attention module, the HAB in the model is replaced with the other current mainstream attention mechanisms, such as SE, CBAM, efficient channel attention (ECA; Q. Wang et al., 2020), and dual attention (DA; Fu et al., 2019). As shown in Table 6, the proposed method outperforms the other variants integrated with current mainstream attention methods on the CamCrack789 and CFD datasets. Additionally, it demonstrates that the attention mechanism has improved the model's ability to extract features. ECA is similar to HAB in terms of *F*1 on the CamCrack789, but the proposed attention module takes into account the characteristics of both spatial and channel attentions of

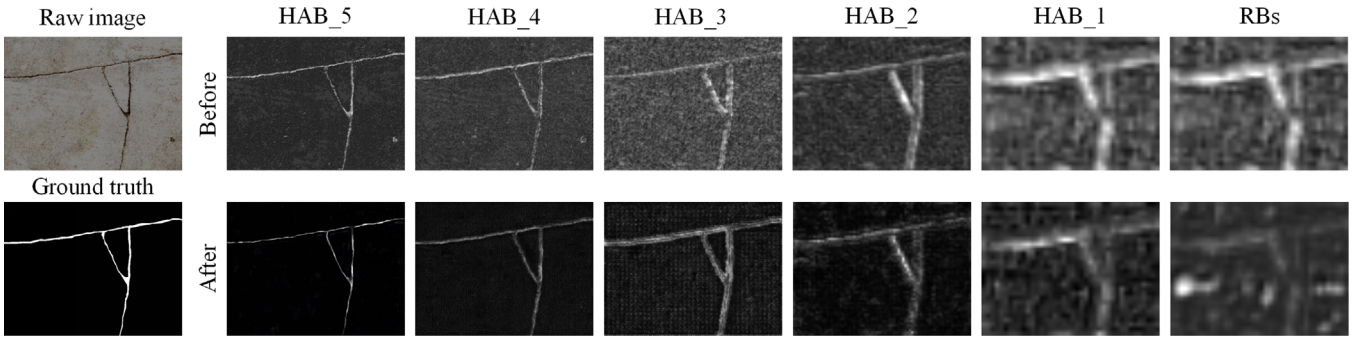

FIGURE 14 Visualization of feature maps before and after each HAB and RBs.

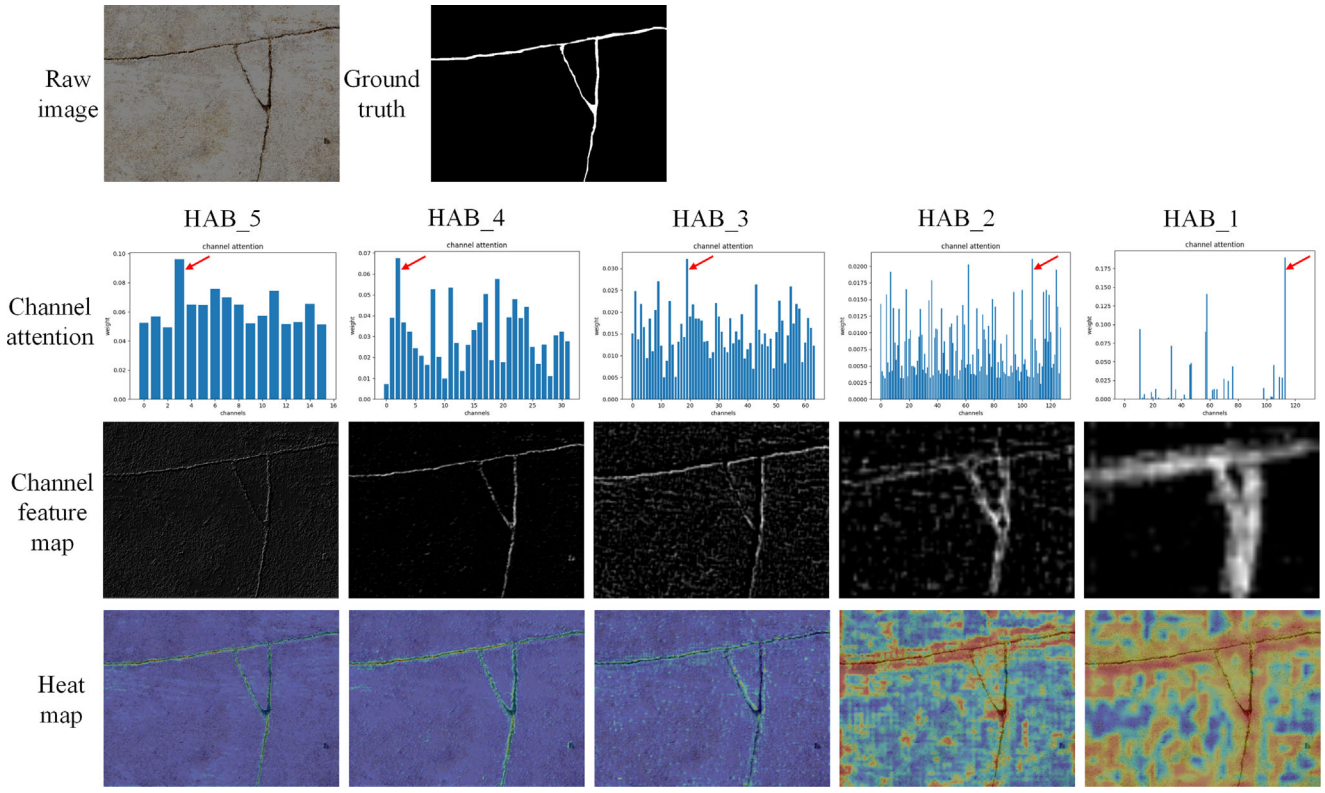

FIGURE 15 Visualization of channel attention maps and spatial attention maps for each HAB.

cracks. From the upper row images of Figure 12, it can be seen that the proposed method performs better than other attention mechanisms in suppressing background interference (see the region marked by the yellow rectangular box). As shown in the lower row images of Figure 12, the crack detection result of HAB is the best. In addition, from the region marked by the red rectangular box, the continuity of the cracks detected by the proposed method is better, compared with other methods of attention. From Figure 12 and Table 5, the proposed HAB algorithm is more suitable for tiny crack detection than the current mainstream attention methods.

3.4 | Ablation study

The ablation study is used to demonstrate the effectiveness of RBs and HABs for crack detection. The experimental results on the CamCrack789 dataset are given in Table 7, in which RBs and HAB denote the RBs and HAB, respectively. It can be seen that both Baseline+RBs and Baseline+HAB obtain better results than Baseline concerning the comprehensive measurement indicator $F1$. This indicates that both RBs and hybrid attention mechanisms contribute to improving the Baseline model's performance. It is noted that the proposed HAB can

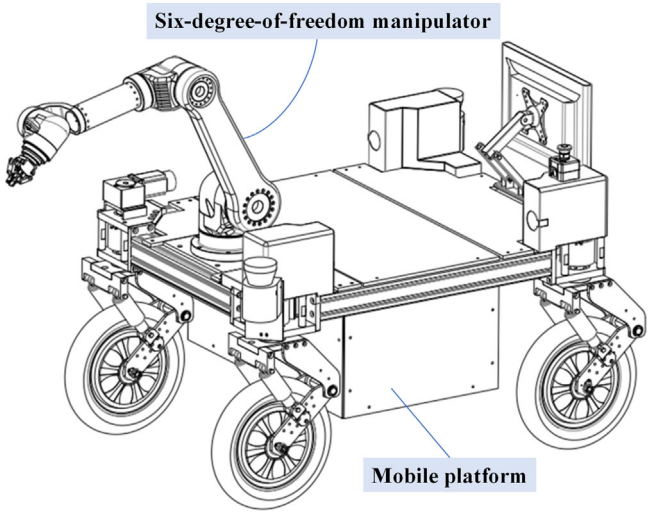

FIGURE 16 Design diagram of the mobile robot.

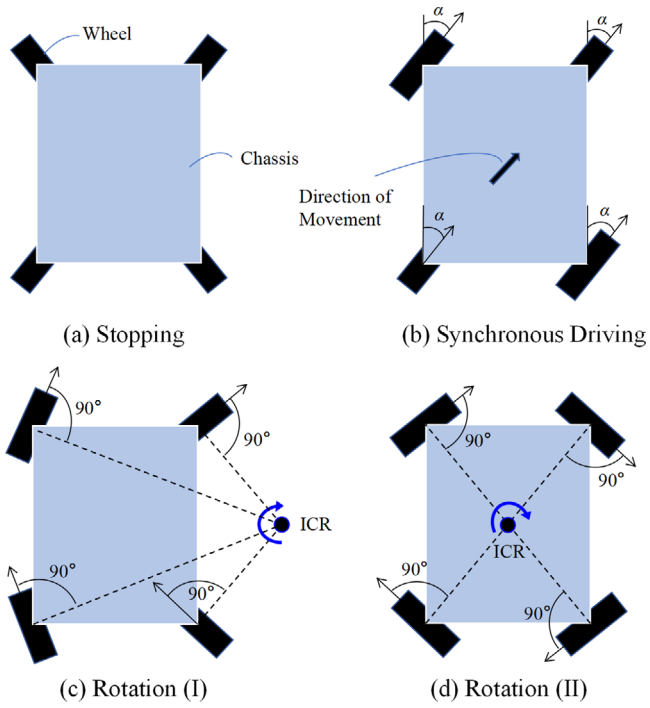

FIGURE 17 Different motion patterns of the mobile platform. ICR denotes the instantaneous center of rotation.

significantly improve the Baseline, increasing the $F1$ value from 0.9323 to 0.9426. Moreover, RHACrackNet performs 3.28% and 1.71% better than Baseline in Pr and $F1$, respectively, demonstrating that integrating both RBs and hybrid attention mechanisms into the Baseline model can further improve its performance. From the visualization result in Figure 13, it can be observed that: (1) the crack shape in the prediction result of Baseline is coarse and has poor continuity (see the region marked by the red rectangular box); (2) Baseline+RBs has better continuity, compared with Baseline, but the result still contains interference in the background (see the region marked by the yellow rectangular box); (3) the background of the result predicted by Baseline+HAB is clean, but the detail of crack is not fine; and (4) RHACrackNet obtains best performance in terms of both capturing details and suppressing background interference.

To intuitively observe the effect of HAB and RBs in the proposed encode–decode architecture, the feature maps before and after each HAB and RBs are given in Figure 14, respectively. It can be seen that after being processed by the modules of HAB and RBs, the feature maps are clearly enhanced on the crack areas, and the noises on the background areas are effectively suppressed. Furthermore, this effect is more evident when the level of the layer becomes higher, which demonstrates that the capabilities of the processed features are increasingly strengthened in the hierarchical architecture of the proposed network. Based on the enhanced processed features, the classifier at the end of the decoder can more readily recognize the crack areas and perform an even more accurate crack segmentation. As shown in Figure 2, HAB is composed of two attention procedures, that is, channel attention and spatial attention. The channel attention can guide the network to select the more discriminate feature channels to represent cracks, which enables us to slim the network by removing abundant feature channels and achieve lightweight computation. The spatial attention can guide the network to focus on the crack areas while avoiding the background inferences such as water stains, shadows, and oil stains.

To further reveal the mechanism of HAB, a visualization study is conducted. By visualizing the channel attention weights and the selected feature channel with the largest attention weight at different layers of the network architecture, as depicted in Figure 15, it can be discerned that the selected feature channel can highlight the presence of cracks (the higher the level of layer, the more clearly the crack features are represented). Then in the subsequent spatial channel module, the feature map filtered by the channel attention mechanism is further enhanced through the spatial attention mechanism. By visualizing the spatial attention map using heat map transformation, it can be clearly observed that the discrimination between crack pixels and non-crack pixels is greatly facilitated by focusing on the significantly reduced regions highlighted by the proposed attention mechanism. In particular, when the depth of the layer increases, the interfering background noises are more effectively suppressed by the spatial attention mechanism, leading to a very clean markup of the focused regions to be paid attention to, which almost overlap with the crack regions. This provides a great advantage because crack regions normally only occupy a small portion of the whole image. Therefore, by introducing the

FIGURE 18 The mobile robot keeps maneuverable in different scenes.

TABLE 8 Comparison of parameters and FLOPs of all models.

Models	Params	FLOPs
FCN [CVPR 2015]	18.64 M	239.46 G
U-Net [MICCAI 2015]	17.25 M	375.24 G
Attention U-Net [MIDL 2018]	34.88 M	624.72 G
DC_Zou [TIP 2018]	30.91 M	1283.64 G
DC_Liu [Neurocomputing 2019]	14.72 M	188.56 G
DMA-Net [TITS 2022]	60.46 M	212.12 G
AttentionCrackNet [CACAIE 2022]	23.47 M	329.02 G
RHACrackNet	**1.67 M**	**21.6 G**
RHACrackNet*	**0.57 M**	**9.68 G**

TABLE 9 Comparison of parameters, FLOPs, and inference time of all models.

Methods	Parameters	FLOPs	Time/image (s)
FCN	18.64 M	145.22 G	0.272
U-Net	17.25 M	187.62 G	0.302
Attention U-Net	34.88 M	312.36 G	0.598
DC_Zou	30.91 M	641.82 G	0.911
DC_Liu	14.72 M	94.28G	0.166
DMA-Net	60.46 M	106.06 G	0.279
AttentionCrackNet	23.47 M	164.52 G	0.348
MobileNet V1	6.59 M	27.44 G	0.091
MobileNet V2	6.8 M	24.7 G	0.106
EfficientNet	8.64 M	15.96 G	0.140
RHACrackNet	1.67 M	21.60 G	0.047
RHACrackNet*	**0.57 M**	**9.68 G**	**0.042**

proposed attention mechanism in this work, the resulting network architecture has a potential of significantly reducing its parameter size, leading to a much faster inference speed while maintaining a high detection performance.

3.5 | Results of real-time deployment

To exemplify the practicability of the developed model, RHACrackNet* trained on CamCrack789 is deployed to a Jetson TX2 mounted on the self-designed mobile robot system to perform a real-world test on the campus road. As illustrated in Figure 16, the mobile robot system mainly consists of an omnidirectional mobile platform and a six-degree-of-freedom manipulator mounted on the platform.

The omnidirectional mobile platform is equipped with four actively steerable wheels. The entire mobile platform weighs approximately 220 kg with a maximum payload of 200 kg and can run for 2 h continuously without payload. As shown in Figure 17, the mobile platform has four different motion patterns, which ensures the flexibility of the whole mobile robot system. The motion patterns consist of stopping, synchronous driving pattern, and two different rotation patterns, that is, rotating around the arbitrary instantaneous center of rotation and rotating around the origin of the robot reference coordinate system.

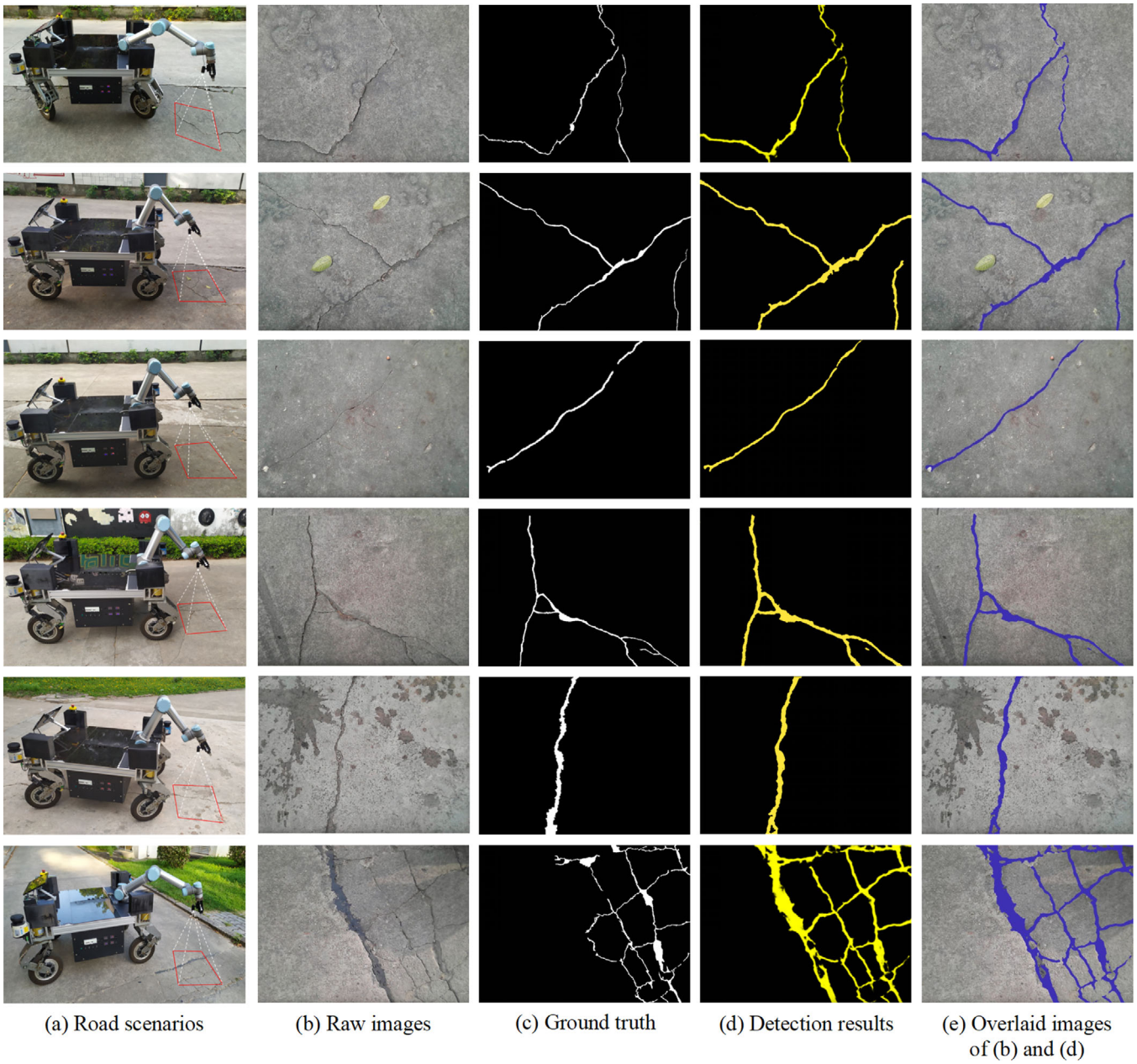

FIGURE 19 Examples of real-time detection results of pavement cracks in different road scenarios.

In different motion patterns, the orientations of the four wheels of the platform are different. As depicted in Figure 18, the mobile robot system can pass the speed bump with about 5-cm height on the road, climb a slope of about 20 degrees, and move on the concrete pavement with some sand and gravel, and even maneuver on the uneven lawn. Therefore, based on the strong motion ability of the mobile platform and the flexibility of the robotic arm, the camera mounted at the end of the manipulator can collect and detect detailed cracks on the road from different heights and perspectives. More information about collecting and detecting pavement cracks using the mobile robot system can be found at https://youtu.be/56P6CdBdirI.

In this work, to deploy the proposed model to embedded devices, DS_Convs are utilized to replace conventional convolutions of Baseline+RBs+HAB except that of HAB. The parameters and FLOPs of all models are listed in Table 8. The two metrics are computed based on an input size of $3 \times 640 \times 480$. The parameters and FLOPs of the proposed RHACrackNet are 1.67 MB and 21.6 GB, which have approximately 90.31% and 94.24% reduction, compared to the 17.25 MB parameters and 375.24 GB FLOPs for U-Net, respectively. In addition, the parameters and FLOPs of RHACrackNet* are only 0.57 MB and 9.68 GB, which have 65.86% and 55.18% reduction, compared to the 1.67 MB parameters and 9.68 GB FLOPs for RHACrackNet,

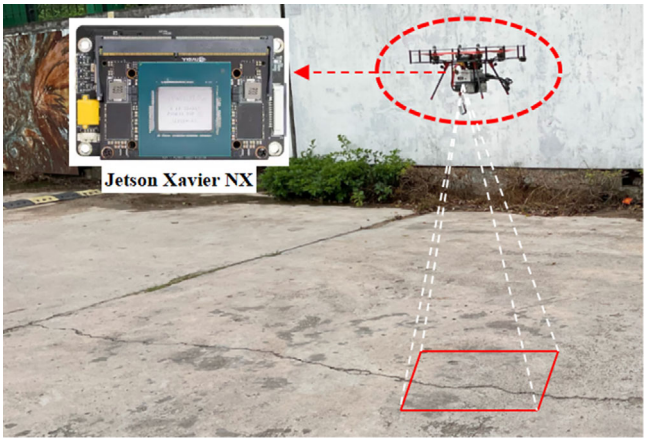

FIGURE 20 Unmanned aerial vehicle (UAV) used for pavement crack detection.

respectively. RHACrackNet* uses DS_Convs instead of conventional convolutions, significantly decreasing the model parameters and computational expense. In summary, the proposed model obtains substantially fewer parameters and higher computational efficiency than other state-of-the-art models. Moreover, RHACrackNet* achieves a better balance between accuracy and computing complexity so as to facilitate model deployment to embedded devices. As a result, the mobile robot system for pavement crack detection reaches about 25 FPS on Jetson TX2. Visualization results of the pavement cracks detected by a self-designed mobile robot on the campus road are shown in Figure 19. From the first and second rows of Figure 19, RHACrackNet* can not only detect coarse cracks but also perform well in detecting tiny cracks. The second to fifth rows in Figure 19 illustrate crack images with various pavement features, including tree leaves, branches, rut marks, and water stains.

It can be observed that RHACrackNet* can accurately detect pavement cracks and cause few false positives, which implies that it can effectively localize pavement cracks and suppress the influence of background noises.

Although the proposed model has improved the detection accuracy of pavement cracks, it still faces challenges in detecting cracks with complex topology. For the last row of Figure 19, some blurred and shallow cracks at the edges of the image are not well identified by the network, which may be caused by the motion of the mobile robot and the uneven illumination. In addition, it is noted that sealed crack pixels are misidentified as crack pixels. This is because the images with sealed cracks are not seen in the training set. Therefore, there is still room for improving the detection performance for complex cracks.

To further illustrate the practicality of the proposed approach, all models (including three mainstream lightweight models such as MobileNet V1 (Howard et al., 2017), MobileNet V2 (Sandler et al., 2018), and EfficientNet (Tan & Le, 2019)) are tested on a UAV (see Figure 20). In particular, the UAV is equipped with a widely used embedded device, NVIDIA Jetson Xavier NX. The camera used for data acquisition is the same on both mobile platforms and can record 2304×1728 pixels of video at 30 FPS. The model input size is set to 480×320. Under this experimental setting, different models are deployed in NVIDIA Jetson Xavier NX installed on the UAV to detect pavement cracks. As shown in Table 9, the proposed RHACrackNet* achieves the fastest inference speed of 23.8 FPS. It is notable that even in comparison with the other three lightweight models, the proposed RHACrackNet* has fewer model parameters, lower computational complexity, and faster inference speed. As shown in Figure 21, the proposed RHACrackNet achieves the best crack segmentation performance. The segmentation result of RHACrackNet* is similar to that of RHACrackNet and significantly better than those of the three lightweight methods. However, although the proposed RHACrackNet* has an acceptable detection speed, in challenging scenarios, such as in the case of high outdoor temperatures in summer, the heating of computing devices may lead to a decrease in detection speed. More work needs to be done to further reduce the size of the model and make it more lightweight in practical future applications.

4 | CONCLUSION

In this work, for automated pixel-wise pavement crack detection, a carefully designed encoder–decoder network RHACrackNet is proposed by integrating RBs and the newly proposed HABs, to achieve an extremely lightweight model that can be deployed in an embedded device. The main contributions of the work can be summarized as follows:

1. Compared with existing models on CamCrack789, Crack500, CFD, and DeepCrack237 datasets, the proposed network obtains the best $F1$ value with 94.94%, 82.95%, 95.74%, and 92.51%, respectively. In addition, the proposed model not only has a fewer parameter size (1.67 M, which is only 1/20 of the parameters of Attention U-Net) but also performs a faster inference speed on all four datasets, compared to the networks in comparison.
2. An even more lightweight version RHACrackNet* with only 0.57 M (about 34% of the parameters of RHACrackNet) model parameters is generated by replacing conventional convolutions with DS_Convs, achieving the fastest inference speed in all the tested crack datasets.

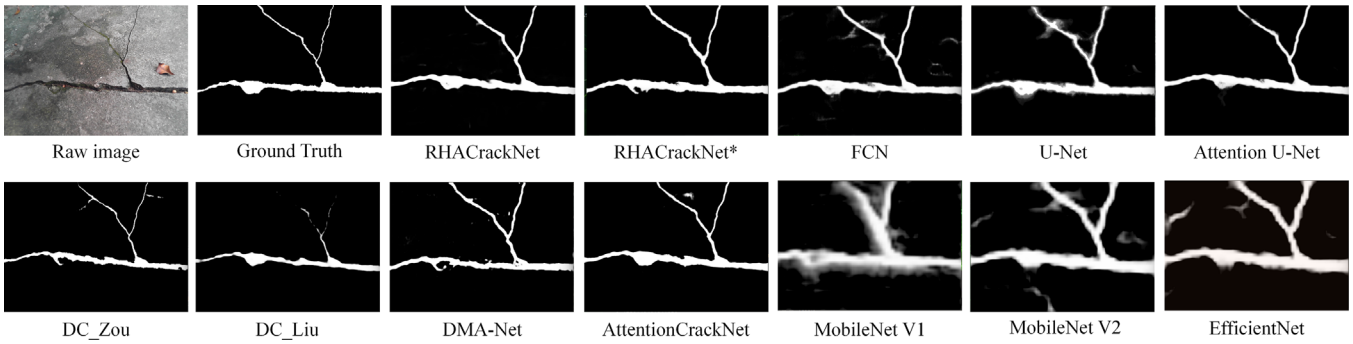

FIGURE 21 Crack detection results from the UAV with Jetson Xavier NX.

3. The practicability of the proposed method is demonstrated by deploying RHACrackNet* trained on CamCrack789 to a Jetson TX2 on a terrain robot and a Jetson Xavier NX on a UAV. Real-world testing shows that the proposed method is capable of detecting pavement cracks in real-time at 25 FPS on the mobile terrain robot and about 24 FPS on the drone.

The work of this paper can be improved in several lines. First, the dataset constructed in this paper only involves the concrete pavement crack dataset, which contains insufficient samples for the proposed model to deal with all actual road crack scenarios. For example, the proposed method has poor detection of crack images with skid-proof stripes, sealed cracks, maintenance hole covers, and so forth. In addition, the self-designed mobile robot will be installed with a drone to collect crack damages in distance in different infrastructures to expand the working range in real-world deployment of the system. Furthermore, it is important to improve the algorithm to enable crack detection in various civil infrastructures. Some promising supervised machine learning algorithms, such as the enhanced probabilistic neural network (Ahmadlou & Adeli, 2010), neural dynamic classification algorithm (Rafiei & Adeli, 2017), dynamic ensemble learning algorithm (Alam et al., 2020), finite element machine for fast learning (Pereira et al., 2020), neural architecture search (Li et al., 2023), and self-supervised learning (Rafiei et al., 2022), will be investigated for the future extension of the research to enhance the model performance for detecting cracks in various construction materials.

ACKNOWLEDGMENTS

This work is financially supported by the National Key Research and Development Program of China (Grant Number 2021ZD0111501, 2021ZD0111502), the National Natural Science Foundation of China (Grant Number 62176147), the Science and Technology Planning Project of Guangdong Province of China (Grant Number 2019A050520001, 2021A0505030072), and the Science and Technology Special Funds Project of Guangdong Province of China (Grant Number STKJ2021216, STKJ2021019).

REFERENCES

Ahmadlou, M., & Adeli, H. (2010). Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integrated Computer-Aided Engineering*, *17*(3), 197–210.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing with Applications*, *32*(10), 8675–8690.

Amhaz, R., Chambon, S., Idier, J., & Baltazart, V. (2016). Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2718–2729.

Celik, F., & König, M. (2022). A sigmoid-optimized encoder-decoder network for crack segmentation with copy-edit-paste transfer learning. *Computer-Aided Civil and Infrastructure Engineering*, *37*(14), 1875–1890.

Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378.

Chen, F. C., & Jahanshahi, M. R. (2018). NB-CNN: Deep learning-based crack detection using convolutional neural network and naïve Bayes data fusion. *IEEE Transactions on Industrial Electronics*, *65*(5), 4392–4400.

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, J., Liu, Y., & Hou, J. A. (2023). A lightweight deep learning network based on knowledge distillation for applications of efficient crack segmentation on embedded devices. *Structural Health Monitoring*, *22*, 14759217221139730.

Chen, T., Cai, Z., Zhao, X., Chen, C., Liang, X., Zou, T., & Wang, P. (2020). Pavement crack detection and recognition using the architecture of segNet. *Journal of Industrial Information Integration*, *18*, 100144.

Cheng, J., Xiong, W., Chen, W., Gu, Y., & Li, Y. (2018). Pixel-level crack detection using U-net. *TENCON 2018-2018 IEEE Region 10 Conference*, Jeju, Korea.

Choi, W., & Cha, Y.-J. (2019). SDDNet: Real-time crack segmentation. *IEEE Transactions on Industrial Electronics*, *67*(9), 8016–8025.

Chu, H., Wang, W., & Deng, L. (2022). Tiny-Crack-Net: A multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks. *Computer-Aided Civil and Infrastructure Engineering*, *37*(14), 1914–1931.

Deng, J., Lu, Y., & Lee, V. C. S. (2020). Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *35*(4), 373–388.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA.

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., 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*.

Fei, Y., Wang, K. C., Zhang, A., Chen, C., Li, J., Liu, Y., Yang, G., & Li, B. (2020). Pixel-level cracking detection on 3D asphalt pavement images through deep-learning-based CrackNet-V. *IEEE Transactions on Intelligent Transportation Systems*, *21*(1), 273–284.

Giglioni, V., Venanzi, I., Poggioni, V., Milani, A., & Ubertini, F. (2023). Autoencoders for unsupervised real-time bridge health assessment, *Computer-Aided Civil and Infrastructure Engineering*, *38*(8), 959–974.

Hu, J., Shen, L., Albanie, S., Sun, G., & Wu, E. (2020). Squeeze-and-excitation networks. *IEEE transactions on pattern analysis and machine intelligence*, *42*(8), 2011–2023.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Ju, H. Y., Li, W., Tighe, S. S., Xu, Z. C., & Zhai, J. Z. (2020). CrackU-Net: A novel deep convolutional neural network for pixelwise pavement crack detection. *Structural Control & Health Monitoring*, *27*(8), e2551.

Jang, K., An, Y. K., Kim, B., & Cho, S. (2021). Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Computer-Aided Civil and Infrastructure Engineering*, *36*(1), 14–29.

Jiang, S., & Zhang, J. (2020). Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system. *Computer-Aided Civil and Infrastructure Engineering*, *35*(6), 549–564.

Kang, D. H., & Cha, Y. J. (2022). Efficient attention-based deep encoder and decoder for automatic crack segmentation. *Structural Health Monitoring*, *21*(5), 2190–2205.

Kim, B., & Cho, S. (2019). Image-based concrete crack assessment using mask and region-based convolutional neural network. *Structural Control and Health Monitoring*, *26*(8), e2381.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kong, S. Y., Fan, J. S., Liu, Y. F., Wei, X. C., & Ma, X. W. (2021). Automated crack assessment and quantitative growth monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *36*(5), 656–674.

Li, W., Wang, Z., Mai, R., Ren, P., Zhang, Q., Zhou, Y., Xu, N., Zhuang, J., Xin, B., Hao, Z., & Fan, Z. (2023). Modular design automation of the morphologies, controllers, and vision systems for intelligent robots: a survey. *Visual Intelligence*, *1*(1), 2.

Liao, J., Yue, Y., Zhang, D., Tu, W., Cao, R., Zou, Q., & Li, Q. (2022). Automatic tunnel crack inspection using an efficient mobile imaging module and a lightweight CNN. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 15190–15203.

Lin, F., Yang, J., Shu, J., & Scherer, R. J. (2021). Crack semantic segmentation using the u-net with full attention strategy. *arXiv preprint arXiv:2104.14586*.

Liu, C., Xu, N., Weng, Z., Li, Y., Du, Y., & Cao, J. (2023). Effective pavement skid resistance measurement using multi-scale textures and deep fusion network, *Computer-Aided Civil and Infrastructure Engineering*, *38*(8), 1041–1058.

Liu, J., Yang, X., Lau, S., Wang, X., Luo, S., Lee, V. C. S., & 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.

Liu, C., & Xu, B. (2022). A night pavement crack detection method based on image-to-image translation. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1737–1753.

Liu, Y., Yao, J., Lu, X., Xie, R., & Li, L. (2019). DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, *338*, 139–153.

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.

Meng, S., Gao, Z., Zhou, Y., He, B., & Djerrad, A. (2023). Real-time automatic crack detection method based on drone. *Computer-Aided Civil and Infrastructure Engineering*, *38*(7), 849–872.

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. *International Conference on Medical Imaging with Deep Learning*, Amsterdam, Denmark.

Pan, X., & Yang, T. (2020). Postdisaster image-based damage detection and repair cost estimation of reinforced concrete buildings using dual convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *35*(5), 495–510.

Pan, Y., & Zhang, L. (2022). Dual attention deep learning network for automatic steel surface defect segmentation. *Computer-Aided Civil and Infrastructure Engineering*, *37*(11), 1468–1487.

Peng, C., Yang, M., Zheng, Q., Zhang, J., Wang, D., Yan, R., Wang, J., & Li, B. (2020). A triple-thresholds pavement crack detection method leveraging random structured forest. *Construction and Building Materials*, *263*, 120080.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404.

Qu, Z., Chen, W., Wang, S. Y., Yi, T. M., & Liu, L. (2021). A crack detection algorithm for concrete pavement based on attention mechanism and multi-features fusion. *IEEE Transactions on Intelligent Transportation Systems*, *23*(8), 11710–11719.

Qu, Z., Wang, C. Y., Wang, S. Y., & Ju, F. R. (2022). A method of hierarchical feature fusion and connected attention architecture

for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 16038–16047.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2022). Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*, Advance online publication. http://doi.org/10.1109/TNNLS.2022.3190448

Ranzato, M., Huang, F. J., Boureau, Y. L., & LeCun, Y. (2007). Unsupervised learning of invariant feature hierarchies with applications to object recognition. *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN.

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

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT.

Shim, S., Kim, J., Lee, S. W., & Cho, G. C. (2021). Road surface damage detection based on hierarchical architecture using lightweight auto-encoder network. *Automation in Construction*, *130*, 103833.

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.

Sun, X., Xie, Y., Jiang, L., Cao, Y., & Liu, B. (2022). DMA-Net: Deeplab with multi-scale attention for pavement crack segmentation. *IEEE Transactions on Intelligent Transportation Systems*, *23*(10), 18392–18403.

Tanaka, N., & Uematsu, K. (1998). A crack detection method in road surface images using morphology, *Proceedings of the Workshop on Machine Vision Applications*, Chiba, Japan.

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*, Long Beach, CA.

Wang, N., Zhao, Q., Li, S., Zhao, X., & Zhao, P. (2018). Damage classification for masonry historic structures using convolutional neural networks based on still images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1073–1089.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA.

Wang, W., & Su, C. (2021). Semi-supervised semantic segmentation network for surface crack detection. *Automation in Construction*, *128*, 103786.

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.

Xu, J., Gui, C., & Han, Q. (2020). Recognition of rust grade and rust ratio of steel structures based on ensembled convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *35*(10), 1160–1174.

Yang, E., Zhang, H., Dong, Z., He, A., Xu, J., & Shang, J. (2022). Intelligent pixel-level detection of multiple distresses and surface design features on asphalt pavements, *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1654–1673.

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2020). Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *21*(4), 1525–1535.

Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Tao, S., Chen, C., Li, J., & Li, B. (2019). Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet. *Journal of Computing in Civil Engineering*, *32*(5), 04018041.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J., & Chen, C. (2017). Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, *32*(10), 805–819.

Zhang, A. A., Wang, K. C., Liu, Y., Zhan, Y., Yang, G., Wang, G., Yang, E., Zhang, H., Dong, Z., He, A., Xu, J., & Shang, J. (2022). Intelligent pixel-level detection of multiple distresses and surface design features on asphalt pavements. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1654–1673.

Zhang, C., Chang, C. C., & Jamshidi, M. (2020). Concrete bridge surface damage detection using a single-stage detector. *Computer-Aided Civil and Infrastructure Engineering*, *35*(4), 389–409.

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ.

Zhang, X., Rajan, D., & Story, B. (2019). Concrete crack detection using context-aware deep semantic segmentation network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(11), 951–971.

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 (CISP2010)*, Yantai, China.

Zhu, W., Zhang, H., Eastwood, J., Qi, X., Jia, J., & Cao, Y. (2023). Concrete crack detection using lightweight attention feature fusion single shot multibox detector. *Knowledge-Based Systems*, *261*, 110216.

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: Zhu, G., Liu, J., Fan, Z., Yuan, D., Ma, P., Wang, M., Sheng, W., & Wang, K. C. P. (2023). A lightweight encoder–decoder network for automatic pavement crack detection. *Computer-Aided Civil and Infrastructure Engineering*, 1–23. https://doi.org/10.1111/mice.13103