Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2024.048175

ARTICLE

A Lightweight Network with Dual Encoder and Cross Feature Fusion for Cement Pavement Crack Detection

Zhong Qu[1,*], Guoqing Mu[1] and Bin Yuan[2]

[1]School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

[2]School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

*Corresponding Author: Zhong Qu. Email: quzhong@cqupt.edu.cn

Received: 29 November 2023 Accepted: 23 January 2024 Published: 16 April 2024

ABSTRACT

Automatic crack detection of cement pavement chiefly benefits from the rapid development of deep learning, with convolutional neural networks (CNN) playing an important role in this field. However, as the performance of crack detection in cement pavement improves, the depth and width of the network structure are significantly increased, which necessitates more computing power and storage space. This limitation hampers the practical implementation of crack detection models on various platforms, particularly portable devices like small mobile devices. To solve these problems, we propose a dual-encoder-based network architecture that focuses on extracting more comprehensive fracture feature information and combines cross-fusion modules and coordinated attention mechanisms for more efficient feature fusion. Firstly, we use small channel convolution to construct shallow feature extraction module (SFEM) to extract low-level feature information of cracks in cement pavement images, in order to obtain more information about cracks in the shallow features of images. In addition, we construct large kernel atrous convolution (LKAC) to enhance crack information, which incorporates coordination attention mechanism for non-crack information filtering, and large kernel atrous convolution with different cores, using different receptive fields to extract more detailed edge and context information. Finally, the three-stage feature map outputs from the shallow feature extraction module is cross-fused with the two-stage feature map outputs from the large kernel atrous convolution module, and the shallow feature and detailed edge feature are fully fused to obtain the final crack prediction map. We evaluate our method on three public crack datasets: DeepCrack, CFD, and Crack500. Experimental results on the DeepCrack dataset demonstrate the effectiveness of our proposed method compared to state-of-the-art crack detection methods, which achieves *Precision* (P) 87.2%, *Recall* (R) 87.7%, and *F-score* (F_1) 87.4%. Thanks to our lightweight crack detection model, the parameter count of the model in real-world detection scenarios has been significantly reduced to less than 2M. This advancement also facilitates technical support for portable scene detection.

KEYWORDS

Shallow feature extraction module; large kernel atrous convolution; dual encoder; lightweight network; crack detection

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Crack detection is an important aspect of ensuring the safety and security of various types of infrastructure [1]. It is crucial to promptly detect, locate, and repair them based on the severity of the damage to prevent the progressive deterioration of cracks and the catastrophic destruction of infrastructure [1]. Therefore, in order to ensure the safety of infrastructure, regular crack detection of cement pavement is necessary.

With the rapid development of computer vision, many researchers have joined this field. Due to the contrast between the background and crack areas, some researchers [2,3] proposed a threshold-based method to detect cracks. Subsequently, other researchers [4,5] considered using edge detection algorithms to reduce noise impact and better detect discontinuous cracks. While these heuristic algorithms achieve superior results in specific scenarios, they are difficult to handle with noise and complex backgrounds with low contrast. To address these deficiencies, the researcher proposed a crack detection algorithm based on random structure forest. By learning the inherent structure information of cracks, the influence of background noise of crack image on crack detection can be suppressed [6], and the crack pixels with uneven gray value distribution can be better extracted. In practice, the background noise of cracks is extremely complex, and it is difficult to distinguish cracks based on the above manual characteristics and traditional machine learning methods.

Due to the rapid development of deep learning technology and its powerful feature extraction ability, researchers have begun to combine deep learning technology with crack detection in order to solve the problems encountered using traditional digital image processing technology [7]. This has effectively improved the reliability and accuracy of crack detection. Full convolutional neural networks (FCN) have been widely used in road detection tasks [8–10] and have achieved state-of-the-art (SOTA) performance. Some works treat crack detection as a segmentation task based on advanced network models such as U-Net [11] or SegNet [12]. Currently, advanced network models improve performance based on multi-scale feature fusion architecture with a powerful backbone [13–15], such as VGG, ResNet, DeepLabV3+.

Enhancing the precision of network models, however, entails a trade-off, as it leads to a rise in network parameters and significantly increases computational demands. For realistic application scenarios that require embedded or mobile devices for crack detection tasks, storage space, computing units and power supplies are extremely limited. Consequently, deploying substantial crack detection models on these devices poses considerable challenges. For the aforementioned issues, we believe that in evaluating crack detection performance, both the number of model parameters and running speed are as important as accuracy. Motivated by these insights, we believe it is essential to develop a streamlined and efficient crack detection model to significantly improve its usability and applicability in real-world scenarios.

Accordingly, this research designs an extremely lightweight network model for crack detection in this paper. As shown in Fig. 1 below, our new network architecture comprises a dual encoder feature extraction module and feature cross-fusion module. In Table 1, the full names corresponding to the abbreviations used in Fig. 1 are provided, offering a clearer understanding of the network structure depicted in the figure. The dual encoder consists of a shallow feature extraction module and large kernel atrous convolution module [16] that integrates the attention mechanism [17]. Shallow convolution is composed of convolution, rectified linear unit (ReLU) and batch normalization (BN) layer, inheriting the advantages of a convolutional neural network. Simultaneously, by minimizing the number of layers, we can effectively reduce the model's parameter size, leading to a more compact and efficient design. By leveraging the characteristics of small operation cost and large receptive field

of large kernel atrous convolution, we have developed a hybrid atrous convolution module combined with coordination attention mechanism to extract more features without increasing computation or parameters. As is shown in Fig. 2, three side outputs of cross-fusion are shown, and the red boxes indicate the noise in the prediction map. Finally, we obtain the final prediction map through our feature cross-fusion module. Our primary contributions are summarized below.

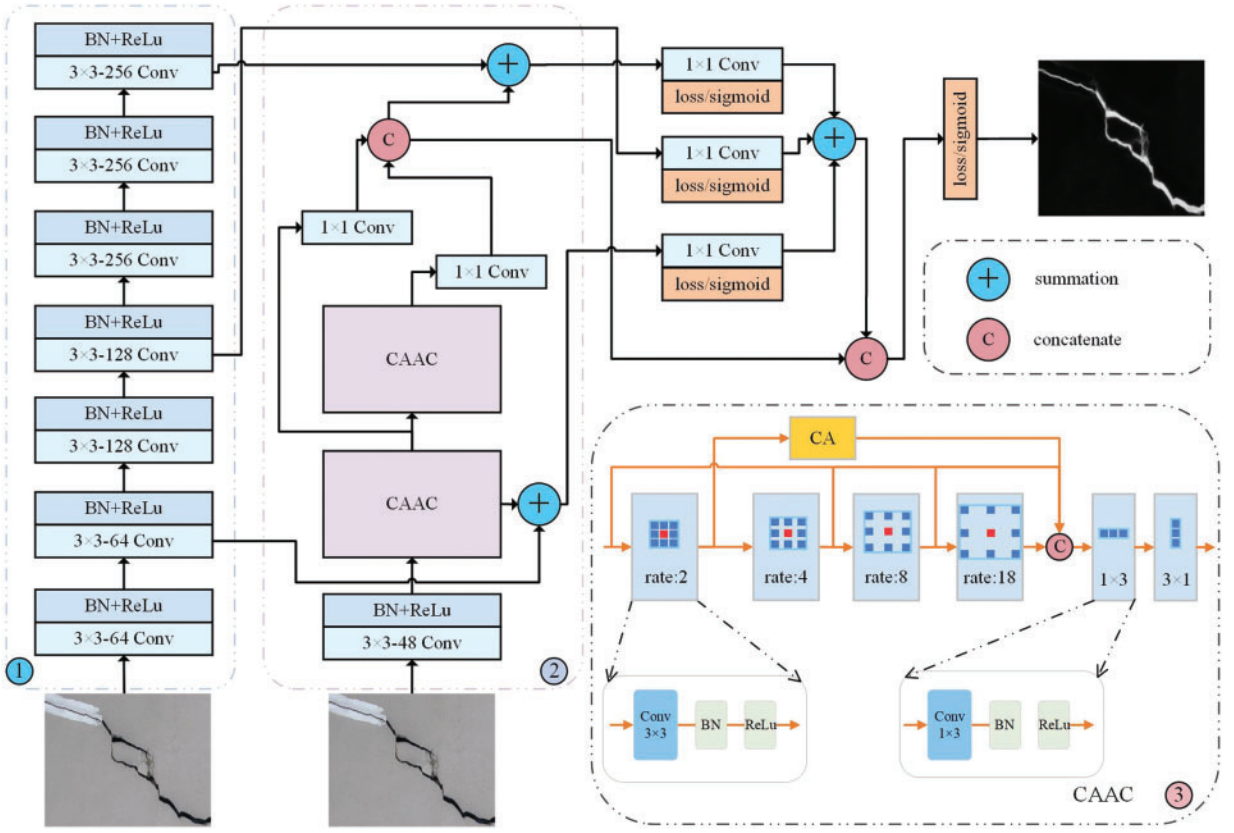

Figure 1: The above overall network architecture is our proposed lightweight crack detection model. Shallow feature extraction module (SFEM), coordination attention atrous convolution (CAAC) and large kernel atrous convolution (LKAC) are its three primary components. (1) Use SFEM to extract shallow features, (2) use LKAC to get crack edge and other contextual information, (3) the CAAC further refine features that is extracted

Table 1: Every abbreviation is matched with its complete spelling for easy reference in Fig. 1

Abbreviation	Complete spelling	Abbreviation	Complete spelling
BN	Batch normalization	ReLU	Rectified linear unit
Conv	Convolution	CAAC	Coordination attention atrous convolution
SFEM	Shallow feature extraction module	LKAC	Large kernel atrous convolution

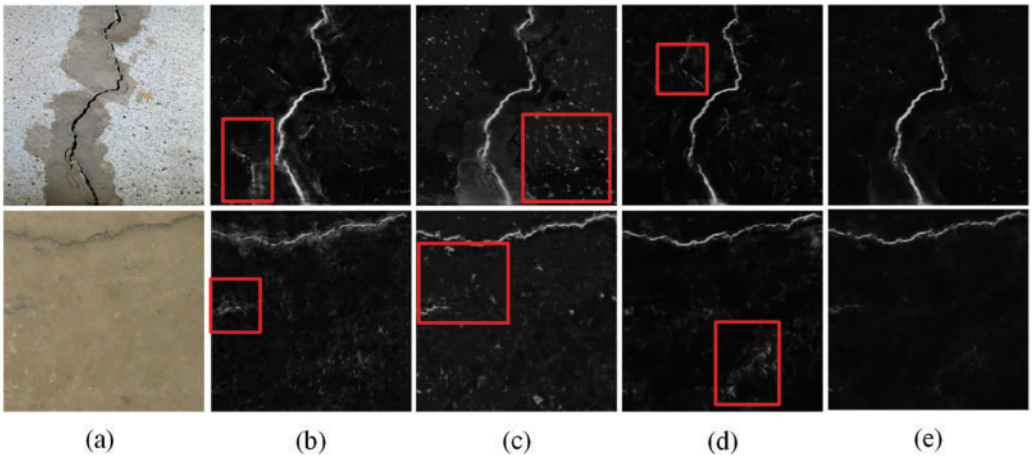

Figure 2: The results of using cross feature fusion and side outputs were compared. (a) Raw images, (b) side output1, (c) side output2, (d) side output3, (e) the fused map after using cross featre fusion

(1) We have designed a new lightweight and highly accurate network architecture for crack detection, which can fully extract crack characteristics while significantly reducing parameters. To capture shallow features of crack information, we use a shallow feature extraction module (SFEM) to improve model accuracy without increasing tedious calculations.

(2) We propose the use of large kernel atrous convolution (LKAC) with lightweight attention to capture deep-level semantic information about cracks for detection. The large kernel atrous convolution has a wide receptive field, and coordinated attention can capture richer context information.

(3) Numerous experiments on public datasets demonstrate the superior performance of our model and fewer parameters than other models in detecting cracks. In the DeepCrack dataset, the F_1 value reaches 87.4%, with only 1.95M parameters.

Other sections for this paper are as follows. Section 2 reviews the work on crack detection and lightweight networks. Section 3 introduces our proposed crack detection architecture. The experimental results are shown in Section 4. Section 5 gives the conclusions and prospects.

2 Related Works

In the early stages of cement pavement crack detection, the primary method of detection and maintenance relied heavily on manual inspection. Manual inspection methods are not only time-consuming but also require significant human, material, and financial resources. Additionally, they suffer from drawbacks such as low detection accuracy and considerable susceptibility to human-induced variations in results. With the continuous advancement of deep learning technologies, various methods and models have been applied to crack detection.

2.1 Crack Detection Methods Based on Encoder Feature Extraction

Since the threshold value of the pavement gray map is related to the average value of pixel brightness, Cheng et al. [18] proposed a real-time image threshold algorithm that reduces sample space and differences to determine appropriate thresholds while reducing sample space. Uneven shadows and lighting are common in photos taken in real scenes, which can seriously affect crack detection by threshold segmentation. Then, in order to reduce the influence of noise in the image background,

the researchers [4,5] used an edge detection algorithm to detect cracks. With the rapid development of deep learning, unprecedented breakthroughs have been made in the field of computer vision. Many deep learning methods have also been applied to crack detection tasks. Dung et al. [19] proposed a coding-decoding full convolutional network for crack detection, which has shown improved accuracy in predicting crack path and density. Liu et al. [20] proposed a network architecture composed of a full convolutional network and a deep supervision network. Zou et al. [21] proposed DeepCrack, an end-to-end trainable deep convolutional neural network for automatic crack detection by learning advanced features of crack representation. Due to the limitations of the receptive field of convolution, some researchers have attempted to use atrous convolution [22]. Hybrid atrous convolutional network (HACNet) had been proposed by Chen et al. [16], which used an atrous convolutional network with an appropriate expansion rate to expand the receiving field while maintaining the same spatial resolution. Zhou et al. [23] proposed an attention mechanism and hybrid pool module to capture both long-range and short-range dependence in crack detection. Qu et al. [24] proposed a concrete pavement crack detection algorithm based on attention mechanism and multi-feature fusion. Yang et al. [25] designed an end-to-end deep crack segmentation network is proposed, which combined progressive and hierarchical context fusion.

The methods previously described predominantly utilize a singular encoder-decoder structure, which fails to capture certain edge detail information and global information effectively. Therefore, we propose a network structure based on dual-encoder feature extraction for crack characterization, designed for dual-level extraction of feature information, ensuring the generation of more precise and accurate prediction maps.

2.2 Crack Detection Methods Based on Lightweight

Due to limited computing resources in actual application scenarios, almost no deep learning network models have been used for crack detection tasks. So a lightweight network for crack detection is urgently needed. In computer vision tasks, knowledge distillation [26] and network pruning are commonly used to build lightweight models; however, these methods are only effective for complex models with high efficiency. Therefore, the most feasible way to achieve a lightweight crack detection network is by building an efficient network structure. For example, MobileNets use intermediate expansion layers that employ lightweight depthwise convolutions to filter nonlinear feature sources [27]. Recently, many lightweight networks based on deep learning have emerged. Liao et al. [28] used the modified residual network to build a lightweight network architecture with an encoder-decoder structure. Zhang et al. [29] proposed a lightweight U-Net model based on attention fusion. Deng et al. [30] proposed asymmetric architectures by gradually fusing information from astrous convolutional layers to reduce network parameters and improve computational and detection performance. These lightweight networks confirm their utility by reducing the size and running time of the model while maintaining similar performance parameters.

Although the aforementioned lightweight networks have achieved certain effects, there still exists an issue of low accuracy, which creates a gap between their performance and the practical application requirements in real-world scenarios. Therefore, dual-encoder network with low parameter count is proposed to address the issues of large parameter size and low accuracy.

2.3 Attentional Mechanism Feature Filtering

There is some unimportant information in the feature extracted by the encoder, which makes it difficult to filter out irrelevant information about cracks. A “Squeeze and Excitation” (SE) block was proposed by Hu et al. [31] to model the interdependencies among channels by adaptively recalibrating

the channel feature response mode and enhancing the representation capability of CNN by improving the spatial coding quality of the entire feature hierarchy. Chen et al. [32] proposed feature maps extracted from the convolutional neural network would be used in Transformer as input sequence to extract global context information, and at last, encoded features would be fused with CNN feature graphs to generate feature maps. In order to capture large receptive field contextual information, Liu et al. [33] designed a self-attention module with 1×1 convolution kernel was proposed to extract context information efficiently across feature channels. For inadequate local feature processing and information loss caused by pooling operations, Yang et al. [34] proposed a multi-scale triple attention network for end-to-end pixel-level crack detection. Zhao et al. [35] made the model adaptively combine local crack features with their global dependencies by establishing a connection between feature interdependencies in channel and spatial dimensions.

In the methods described above, the employed attention mechanisms, while effective in filtering out non-crack information, concurrently increase the complexity of the network model due to their substantial parameter size. Consequently, the implementation of attention mechanisms with smaller parameter sizes, capable of efficiently filtering irrelevant information within the network, can enhance performance without increasing the complexity of the model.

3 Proposed Approach

3.1 Network Architecture

In our paper, crack detection is treated as a pixel-level segmentation task, where "0" represents "non-crack pixels" and "1" represents "crack pixels". This is shown in Table 2, Our network model achieves the highest F_1 value of 0.874 and $MIoU$ of 0.883 on the DeepCrack dataset. Compared with all the contrastive methods listed in Table 2, the image segmentation evaluation metrics employed in this study have shown optimal outcomes in every aspect. Benefiting from atrous convolution and lightweight attention [17], as demonstrated in Fig. 1, We have developed a lightweight network featuring dual-encoder structure, consisting of shallow feature extraction module and large kernel atrous convolution module integrated with lightweight attention mechanism. On this basis, features extracted from the two encoders are cross-fused in the manner depicted in Fig. 1, effectively generating the final predictive image. The shallow feature extraction module (SFEM) is divided into three stages and comprises seven convolution blocks. As shown in Fig. 3, the detailed information of our model and the specific configurations of each module are elaborately depicted. Each block consists of one convolution layer with 3×3 kernel, followed by one batch normalization (BN) layer and one rectified linear unit (ReLU) layer. Specifically, the first stage consists of two convolution blocks with 64 channels each, the second stage is composed of two convolution blocks with 128 channels each, and the third stage includes three convolution blocks with 256 channels each. These three stages together form the SFEM, serving as one of the encoders. The large kernel atrous convolution (LKAC) module is divided into upper, middle, and lower parts. The upper part consists of one convolution layer with 3×3 kernel and 48 channels, followed by one batch normalization (BN) layer and one ReLU layer. The middle part is primarily composed of the coordination attention atrous convolution (CAAC) module. The lower part includes two convolution layers, each with 1×1 kernels and one channel, corresponding to the outputs of the two CAAC modules. Finally, the results from these two layers are concatenated. In the network's decoder, the feature cross-fusion stage is employed. Specifically, the output feature map from the first stage of the SFEM is added to the result of the first CAAC module to generate the first side output. The output feature map from the second stage of SFEM is directly used as the second side output. The output from the third stage of SFEM is added to the output of the LKAC encoder

to form the third side output. Finally, the feature map obtained by adding these three side outputs is concatenated with the output of the LKAC encoder to produce the final prediction map.

Table 2: Evaluation metrics with testing and training for each method on the DeepCrack dataset

Methods	P	R	F_1	PA	MA	$MIoU$
CrackW-Net [36]	0.795	0.821	0.808	0.983	0.906	0.830
DeepCrack [20]	0.825	0.838	0.832	0.985	0.915	0.848
U-Net [11]	0.848	0.849	0.848	0.987	0.921	0.861
DeepLabV3+ [15]	0.801	0.810	0.805	0.983	0.900	0.828
UHDN [37]	0.842	0.772	0.805	0.984	0.886	0.829
CrackSegNet [38]	0.780	0.829	0.804	0.983	0.910	0.827
FPHBN [39]	0.807	0.834	0.821	0.984	0.913	0.840
HACNet [16]	0.855	0.864	0.859	0.990	0.930	0.870
Ours	**0.872**	**0.877**	**0.874**	**0.991**	**0.941**	**0.883**

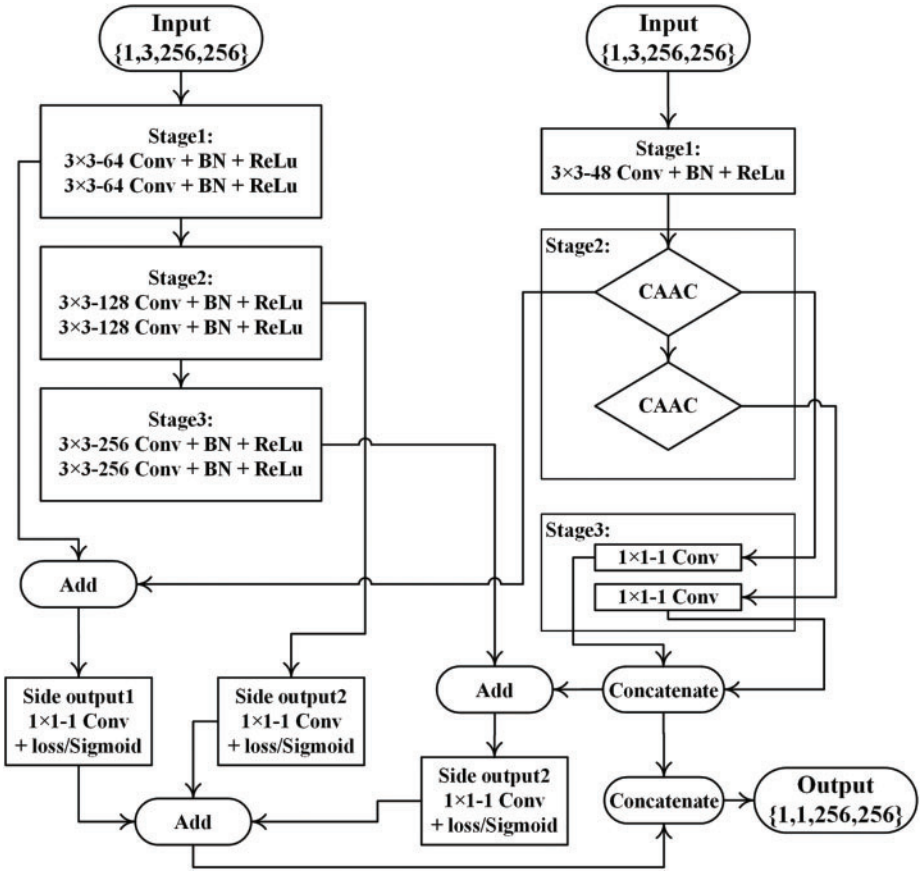

Figure 3: Flowchart of the research methodology steps. The flowchart includes detailed annotations of the input and output dimensions, as well as the implementation specifics of the module parameters

3.2 Coordination Attention Atrous Convolution

Ordinary convolution has some fatal defects, such as frequent upsampling and pooling operations, which will lead to the loss of internal data structure and loss of spatial hierarchical information. We will avoid these operations and use atrous convolutions for feature extraction. As shown in Fig. 1, the CAAC module comprises four groups of atrous convolution blocks, each characterized by varying dilation rates. Each group predominantly consists of one 3×3 atrous convolution layer, followed by one BN layer and one ReLU layer. After cascading these four groups of atrous convolution blocks, the output is processed through two 1-D convolution blocks. The first 1-D convolution block includes one 1×3 convolution layer, one BN layer, and one ReLU layer, while the second 1-D convolution block is composed of one 3×1 convolution layer, one BN layer, and one ReLU layer.

Inspired by HACNet [16], the dilation rates for these four groups of atrous convolution blocks are sequentially set as $\{2, 4, 8, 18\}$. For the atrous convolution block with dilation rate of 2, coordination attention (CA) mechanism is adopted. This is due to the more complex nature of the information features extracted in the atrous convolution layers with smaller dilation rates. Since 1-D convolution has the advantages of fewer parameters and less computing resources, we cascade and fuse the outputs of all atrous convolution blocks and pass a pair of 1-D convolution before the final output to reduce the number of channels of the feature map and achieve the effect of refining the features. In this module, the receptive field (RF) size in the j th convolutional layer is calculated as follows:

$$RF_j = (RF_{j-1} - 1) \times s_j + 1 + (k_j - 1) \times r_j, \tag{1}$$

where RF_j is the size of the RF in the j th layer and the same RF_{j-1} is the RF size at layer $(j-1)$. In the j th layer, s_j, k_j, r_j represent the dilation rate, kernel size, and stride, respectively.

3.3 Lightweight Attention Mechanism

Recently, the attention mechanism has been widely used in crack detection methods based on deep learning [23,24,29]. The features extracted by the crack detection network model contain a lot of detailed information, but also contain a lot of noise, which leads to the final result is not clear. Currently, prevailing crack detection methodologies employ the attention mechanism. This approach enhances performance while simultaneously increasing the parameter count and computational resource requirements. To solve this problem, coordinated attention with a small number of parameters and occupying small computational resources is used in our proposed network. The diagram of coordinated attention intends to be designed as shown in Fig. 4. In order for the attention block to capture long-distance interactions with precise location information, we decompose the global pooling into a pair of 1-D average pooling to aggregate features along the horizontal and vertical directions, respectively, to generate bidirectional spatial feature perception. For example, if we set the pooling kernels to be $(H,1)$, $(1,W)$, and input T, the output of the c-th channel at height h can be written as,

$$m_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} t_c(h, i), \tag{2}$$

Similarly, the output of the c-th channel at width w can be formulated as,

$$m_c^w(w) = \frac{1}{H} \sum_{0 \le i < H} t_c(j, w). \tag{3}$$

The feature aggregation of formulas (2) and (3) in the generation process can be expressed as,

$$n = \zeta\left(C\left(\left\langle m^h, m^w \right\rangle\right)\right), \tag{4}$$

where $\langle \cdot, \cdot \rangle$ represents the concatenation them along the horizontal and vertical, ζ is a non-linear activation function, C is the convolution operation. Then, we divide n into horizontal n^w and vertical n^h. What's more, C_h and C_w are 1×1 convolutions that transform n^h and n^w to have the same number of channels as the input T. The outputs k^h and k^w are then expanded and used as attention weights, respectively. Finally, the output O via attention can be formulated as,

$$k^h = \gamma\left(C_h(n^h)\right), \tag{5}$$

$$k^w = \gamma\left(C_w(n^w)\right), \tag{6}$$

$$O_c(i,j) = k_c^h(i) \times k_c^w(j) \times t_c(i,j), \tag{7}$$

where γ denotes the sigmoid function.

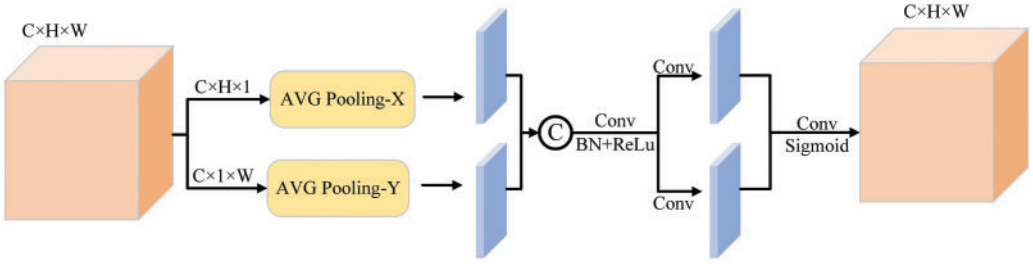

Figure 4: The details of the coordination attention mechanism. The input of feature maps are divided into horizontal and vertical average pooling parts, called AVG Pooling-X and AVG Pooling-Y, respectively. For these two parts, convolution is performed separately and then concatenated

3.4 Loss Function

In our proposed model, the loss function is an integral part, supervising our side outputs as well as the final output, so that the prediction results are closer to the label map quickly. We denote the training set by $S = (R^n, K^n), n = 1, \ldots, N$, where R^n stands for the original image and K^n for the labeled image. W denotes all parameters of the whole network, and all the side output layers can be viewed as classifiers with parameter $w^{(m)}$. $\overline{F}$ denotes the result of the model prediction. Then the side loss function L_{side} can be formulated as,

$$\begin{aligned} L_{side}(R, K, W, w) &= \sum_{m=1}^{3} \alpha^m l_{side}^m \left(R, K, W, w^{(m)}\right) \\ &= \sum_{m=1}^{3} \alpha^m \theta\left(\overline{F}, K, W, w^{(m)}\right), \end{aligned} \tag{8}$$

where θ denotes the modified cross-entropy function, α^m is a hyper-parameters representing the loss weight for each side output layer, which can be rewritten as,

$$\begin{aligned} \theta = &-u_0 \sum_{i \in G^+} \log \Pr\left(k_i = 1 | R; W, w^{(m)}\right) \\ &-u_1 \sum_{i \in G^-} \log \Pr\left(k_i = 0 | R; W, w^{(m)}\right), \end{aligned} \tag{9}$$

where u_0, u_1 denote the class loss weights for cracks and non-cracks, respectively, G^+ and G^- represent the total number of cracked and non-cracked pixels. $\Pr(\cdot)$ means the probability of positive or negative for a pixel in the predicted map. $k_i \in \{1, 0\}$ refers to the predicted results of the m-th side output layer. The final fusion loss obtained by concatenating all the side output layers can be expressed as,

$$\begin{aligned} L_{fuse}(R, K, W) = &-u_0 \sum_{i \in G^+} \log \Pr(k_i = 1 | R; W) \\ &-u_1 \sum_{i \in G^-} \log \Pr(k_i = 0 | R; W), \end{aligned} \tag{10}$$

Therefore, the total loss function of the model can be simplified as,

$$L = L_{side}(R, K, W, w) + L_{fuse}(R, K, W). \tag{11}$$

4 Expiremental Results and Discussion

In this section, we mainly focus on the presentation of experimental results. Firstly, let us briefly describe the implementation details, then introduce the relevant datasets. Finally, we introduce the compared model methods as well as the evaluation metrics and the ablation studies.

4.1 Implementation Details

Both our proposed model and compared models are implemented on PyTorch, a public deep learning framework. In the proposed network, batch normalization and ReLU are used after each convolutional layer in order to make the model converge faster during training. In the model, the initial learning rate is set to 1e-4 and is reduced to 10 times every 50 epochs, the training epoch is set to 500. We adopt stochastic gradient descent (SGD) as the optimizer with weight decay (2e-4) and momentum (0.9). Experiments are implemented with the 4-core Inter (R) Xeno (R) Sliver CPU and the Tesla A100 40 GB GPU on the Ubuntu 16.04 system.

4.2 Datasets

We train and test on the DeepCrack dataset, and verify the effectiveness of the model on CFD and Crack500 datasets. A concise introduction of each dataset is given below.

(1) *DeepCrack* [20]: This dataset contains 537 crack images with a resolution of 544 × 384. It is divided into two groups, the training set contains 300 images and the test set contains 234 images.

(2) *CFD* [39]: This dataset contains 118 crack images with a resolution of 480 × 320. Each image has its corresponding pixel-wise label image. Our model is tested on the CFD dataset.

(3) *Crack500* [40]: This dataset contains 500 images with a resolution of 2000 × 1500. These images were divided into 3 sets, 250 training sets, 50 validation sets, and the remaining 200 as the test set.

4.3 Comparison Methods

(1) *U-Net* [11]: The model is composed of U-shaped encoder-decoder structure and skip connection layer.

(2) *UHDN* [37]: The prediction of the image is realized by encoder-decoder architecture with hierarchical feature learning and dilated convolution.

(3) *DeepLabV3+* [15]: It is a combination of the advantages of the spatial pyramid pooling module and the encoder-decoder structure.

(4) *DeepCrack* [20]: This is a fully convolutional network and refines the result with guided filtering and conditional random fields.

(5) *HACNet* [16]: It uses a hybrid approach to concatenate atrous convolutions with different dilation rates to aggregate features.

(6) *FPHBN* [39]: The network aggregates contextual information into low-level features through a feature pyramid.

(7) *CrackSegNet* [38]: This network consists of backbone network, dilated convolution, spatial pyramid pooling and skip connection modules.

(8) *CrackW-Net* [36]: Based on U-Net, a skip-level round-trip sampling block is proposed.

4.4 Evaluation Metrics

Parameter count (*Params*) is an important evaluation criterion for a lightweight model. We measure the running speed of these models in frames per second (*FPS*). In addition, the *Params* and *FLOPs* of each convolution layer can be expressed as follows:

$$Params = (C_{in}T^2 + 1)C_{out}, \tag{12}$$

$$FLOPs = 2HW(C_{in}T^2 + 1)C_{out}, \tag{13}$$

where T denotes the size of convolution kernel, C_{in} and C_{out} indicate the nuber of channels for input and output feature maps, respectively. H and W respent the height and width of the feature map, respectively. These methods' performance is evaluated using precision (P), recall (R), and F-score (F_1), the pixel accuracy (PA), mean pixel accuracy (MA), mean intersection over union ($MIoU$). The equations are calculated as,

$$P = \frac{TP}{TP + FP}, \tag{14}$$

$$R = \frac{TP}{TP + FN}, \tag{15}$$

$$F_1 = \frac{2PR}{P + R}, \tag{16}$$

where TP denotes true positives, FP means false positives and FN refers false negatives.

$$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}}, \tag{17}$$

$$MA = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}, \tag{18}$$

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}}, \tag{19}$$

where k means the number of classes, p_{ii} denotes the number of pixels with classification i that are predicted to be i, while p_{ij} indicates the number of pixels with classification i but predicted to be j.

4.5 Experimental Results

To evaluate the validity of our experiment, we introduce eight metrics used in crack detection: *Precision*, *Recall*, *F-score*, *MIoU*, *Params*, *Training time*, *FPS*, *FLOPs*. Where, *Params*, *FPS* and *FLOPs* evaluate the complexity of the model, and *Training time* denotes the time for the network to run one epoch.

(1) *Results on DeepCrack*

Fig. 5a displays the Precision-Recall curves on the DeepCrack dataset, with our model reaching the upper right corner. It exceeds current crack detection methods by achieving the highest F_1 value of 0.874. As illustrated in Table 2, our method compared to FPHBN and CrackW-Net, shows an improvement of 5.3% and 6.6% on the F_1 value, and an enhancement of 4.3% and 5.3% on the *MIoU*, respectively. The two methods mentioned above can be attributed to the utilization of distinct fusion techniques combined with encoder-decoder network structures. What's more, our method achieves the highest *MIoU* value of 0.883. Fig. 6 presents the feature maps of test results on three datasets: DeepCrack, CFD, and Crack500, comparing our method with others. The first two rows are derived from the DeepCrack dataset, the middle two rows from the CFD dataset, and the final two rows from the Crack500 dataset. Notably, Fig. 6c illustrates the results from our model, where it can be observed that the edge details of the cracks are clearer and more abundant, showing the highest conformity with the label images. In our model, there are three sideoutput feature maps. Table 3 presents the evaluation metrics on the DeepCrack dataset, showing that all metrics gradually increase, with the results reaching their optimum after fusion.

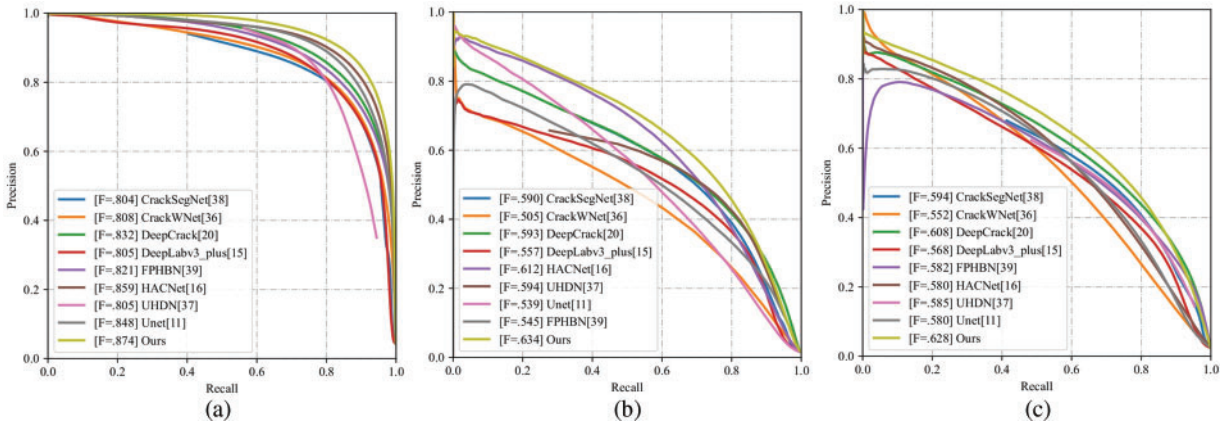

Figure 5: The precision and recall curves on the testing set of (a) DeepCrack, (b) CFD and (c) Crack500

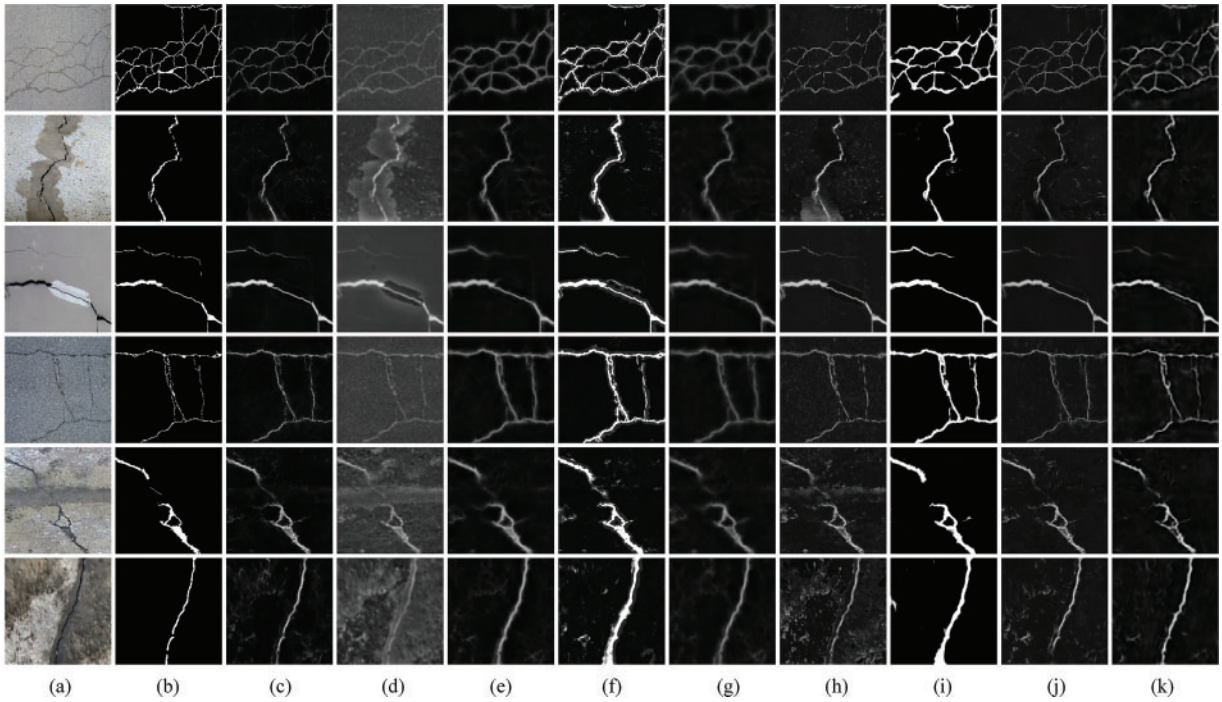

Figure 6: From DeepCrack, CFD and Crack500, visualization results of different methods were obtained. Columns from left to right as: (a) raw image (b) ground truth (c) ours (d) CrackW-Net [36] (e) DeepCrack [20] (f) CrackSegNet [38] (g) FPHBN [39] (h) U-Net [11] (i) UHDN [37] (j) HACNet [16] (k) DeepLabV3+ [15]

Table 3: Evaluation metrics of three side outputs on DeepCrack dataset

Methods	P	R	F_1	PA	MA	$MIoU$
Sideoutput1	0.845	0.852	0.848	0.986	0.922	0.861
Sideoutput2	0.858	0.840	0.849	0.987	0.917	0.862
Sideoutput3	0.871	0.849	0.859	0.988	0.924	0.871
Fused output	**0.872**	**0.877**	**0.874**	**0.991**	**0.941**	**0.883**

(2) *Results on CFD and Crack500*

We use these two datasets to verify the generalization of our approach. The trained models on DeepCrack dataset are used to predict the maps on CFD and Crack500 datasets. As demonstrated in Table 4 our approach achieves the highest F_1 value 0.640 and $MIoU$ value 0.736 on CFD dataset, respectively. As shown in Table 5, our method reaches the highest F_1 value 0.630 and $MIoU$ value 0.725 on Crack500 dataset, separately. Compared to HACNet [16], our method also employs a similar atrous convolution extraction module. However, in contrast to the single encoder feature extraction used by HACNet, our dual encoder approach significantly enhances performance. In terms of the F_1 and $MIoU$ metrics, our method shows an improvement of 3.2% and 2.2%, respectively, over HACNet on the CFD dataset. On the Crack500 dataset, the F_1 and $MIoU$ improved by 5% and 3.2%, respectively. Figs. 5b and 5c show that our method has a better Precision-Recall curve than other methods on both the

CFD and Crack500 datasets. Table 6 displays the sideoutput and post-fusion results for the CFD and Crack500 datasets. The fusion of the three sideoutputs achieved the highest F_1 and $MIoU$.

Table 4: Evaluation metrics of compared methods test on CFD dataset

Methods	P	R	F_1	PA	MA	$MIoU$
CrackW-Net [36]	0.465	0.553	0.505	0.983	0.771	0.660
DeepCrack [20]	0.547	0.648	0.593	0.986	0.820	0.704
U-Net [11]	0.524	0.556	0.540	0.985	0.774	0.667
DeepLabV3+ [15]	0.494	0.634	0.558	0.984	0.810	0.685
UHDN [37]	0.531	**0.674**	0.594	0.985	**0.832**	0.703
CrackSegNet [38]	0.550	0.636	0.590	0.986	0.820	0.702
FPHBN [39]	0.500	0.600	0.550	0.984	0.795	0.679
HACNet [16]	0.600	0.625	0.612	0.987	0.802	0.714
Ours	**0.610**	0.660	**0.640**	**0.996**	0.827	**0.736**

Table 5: Evaluation metrics of compared methods test on Crack500 dataset

Methods	P	R	F_1	PA	MA	$MIoU$
CrackW-Net [36]	0.541	0.562	0.552	0.978	0.760	0.680
DeepCrack [20]	0.564	0.660	0.608	0.980	0.811	0.707
U-Net [11]	0.573	0.586	0.580	0.977	0.796	0.693
DeepLabV3+ [15]	0.526	0.612	0.568	0.976	0.795	0.686
UHDN [37]	0.524	**0.664**	0.585	0.978	0.805	0.695
CrackSegNet [38]	0.543	0.656	0.594	0.977	0.815	0.701
FPHBN [39]	0.533	0.641	0.582	0.978	0.799	0.694
HACNet [16]	0.571	0.590	0.580	0.979	0.783	0.693
Ours	**0.610**	0.660	**0.630**	**0.982**	**0.824**	**0.725**

Table 6: Evaluation metrics of three side outputs on CFD and Crack500 datasets

Feature maps	CFD						Crack500					
	P	R	F_1	PA	MA	$MIoU$	P	R	F_1	PA	MA	$MIoU$
Sideoutput1	0.576	0.584	0.581	0.986	0.778	0.697	0.541	0.521	0.538	0.976	0.755	0.669
Sideoutput2	0.549	0.546	0.548	0.985	0.769	0.681	0.551	0.601	0.574	0.979	0.783	0.691
Sideoutput3	**0.615**	0.626	0.621	0.987	0.810	0.718	0.597	0.606	0.600	0.981	0.797	0.704
Fused output	0.610	**0.660**	**0.640**	**0.996**	**0.827**	**0.736**	**0.610**	**0.660**	**0.630**	**0.982**	**0.824**	**0.725**

(3) *Model Complexity*

In Table 7, *Params* and *training time* are shown. In terms of the *Params* metric, HACNet has the lowest at 0.21 M, showing a significant difference compared to other methods. However, the parameter count of our method is also modest at 1.95 M, closely aligning with that of HACNet. Notably, our *training time* is just 20 s, which represents a significant improvement in comparison. As shown in Table 8, the complexity of the model is evaluated using *FPS* and *FLOPs*. The above evaluation of the complexity of our model is based on the DeepCrack dataset. Benefiting from the lightweight design of the network architecture, our proposed method achieves the greatest *training time* value of 20 s and *params* only 1.95 M. We use lightweight modules CA, SFEM and LKAC to make the model much less complex than the comparison methods.

Table 7: The *Training time* and *Params* on DeepCrack dataset

Methods	*Training time* (*s*)	*Params* (*M*)
CrackW-Net [36]	50	28.37 M
DeepCrack [20]	29	14.72 M
U-Net [11]	45	26.36 M
DeepLabV3+ [15]	83	59.34 M
UHDN [37]	32	34.49 M
CrackSegNet [38]	44	14.97 M
FPHBN [39]	35	14.81 M
HACNet [16]	43	**0.21 M**
Ours	**20**	1.95 M

Table 8: The *FPS* and *FLOPs* on DeepCrack dataset

Methods	*FPS*	*FLOPs*
CrackW-Net [36]	19	26.33G
DeepCrack [20]	**20**	20.08G
U-Net [11]	13	55.85G
DeepLabV3+ [15]	8	65.16G
UHDN [37]	14	60.97G
CrackSegNet [38]	17	57.55G
FPHBN [39]	5	20.48G
HACNet [16]	14	**13.52G**
Ours	10	31.91G

4.6 Ablation Study

This section primarily serves to validate the effectiveness of each component of our model, which includes three principal modules. As shown in Table 9, crack detection can be effectively enhanced by proposed network architectures and modules. We used a shallow feature extraction module (SFEM)

as a feature extraction module and take large kernel atrous convolution (LKAC) as another feature extraction module, combined them and added coordination attention (CA) successively. As can be seen from the results in Table 9, the F_1 values of the two feature extraction modules are very similar on the DeepCrack dataset, and the F_1 value increased by about 1% when they are combined. But the F_1 value decreased by 2% and 5% because of adding CA to the LKAC network in DeepCrack and CFD, respectively. In order to reduce the non-crack information in the results after the combination of SFEM and LKAC, we try to add CA module to filter the characteristic information. In Table 9, the LKAC module exhibits a slightly higher precision on the CFD dataset compared to our model. This is attributed to the LKAC module primarily utilizing atrous convolution for crack feature extraction, without employing upsampling and pooling layers, thus leading to less information loss and a slight improvement in accuracy.

Table 9: The Ablation experiment on DeepCrack and CFD

Settings	DeepCrack						CFD					
	P	*R*	F_1	*PA*	*MA*	*MIoU*	*P*	*R*	F_1	*PA*	*MA*	*MIoU*
SFEM	0.852	0.860	0.856	0.985	0.925	0.854	0.583	0.624	0.603	0.986	0.786	0.692
LKAC	0.853	0.865	0.859	0.987	0.929	0.869	**0.614**	0.630	0.622	0.987	0.812	0.719
LKAC + CA	0.822	0.861	0.840	0.987	0.808	0.709	0.560	0.580	0.570	0.987	0.926	0.867
SFEM + LKAC	0.862	0.871	0.865	0.988	0.928	0.874	0.605	0.651	0.627	0.987	0.817	0.722
Ours	**0.872**	**0.877**	**0.874**	**0.991**	**0.941**	**0.883**	0.610	**0.660**	**0.640**	**0.996**	**0.827**	**0.736**

5 Conclusion

In this paper, we primarily propose a novel network architecture for crack detection, which is based on a dual encoder framework. The network structure is jointly composed of the shallow feature extraction module (SFEM) and large kernel atrous convolution (LKAC) module. The LKAC module is constructed using atrous convolution and coordination attention to extract context information from a large receptive field. Compared to advanced deep learning methods for crack detection, our approach significantly surpasses them in terms of computational complexity and detection accuracy. Numerous experiments have proven the superiority and generalization of our proposed network model. The minimal parameter requirement of our model, at only 1.95 M, significantly facilitates its application in practical scenarios, particularly in crack detection models used in road damage detection vehicles. This not only greatly reduces computational demands but also achieves optimal detection accuracy. Consequently, real-world detection becomes not only more cost-effective but also benefits from enhanced speed and precision in detection.

We hope this study will provide new ideas for lightweight crack detection research that can be applied to mobile detection equipment. However, there is still room for improvement in our model's operating speed and detection accuracy. In the future, we will persist in our investigation of lightweight network architectures for crack feature extraction to enhance detection speed while preserving high accuracy.

Acknowledgement: The authors wish to thank the associate editors and anonymous reviewers for their valuable comments and suggestions on this paper.

Funding Statement: This work was supported by the National Natural Science Foundation of China (No. 62176034), the Science and Technology Research Program of Chongqing Municipal Education

Commission (No. KJZD-M202300604) and the Natural Science Foundation of Chongqing (Nos. cstc2021jcyj-msxmX0518, 2023NSCQ-MSX1781).

Author Contributions: Conceptualization, Zhong Qu; methodology, Zhong Qu, Guoqing Mu; formal analysis, Guoqing Mu, Bin Yuan; data curation, Guoqing Mu; writing—original draft preparation, Zhong Qu, Guoqing Mu; supervision, Zhong Qu, Guoqing Mu, Bin Yuan. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author, upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Yang, Y. L., Xu, W. J., Zhu, Y. F., Su, L. L., Zhang, G. Q. (2023). A novel detection method for pavement crack with encoder-decoder architecture. *Computer Modeling in Engineering & Sciences, 137(1),* 761–773. https://doi.org/10.32604/cmes.2023.027010
2. Cheng, H. D., Shi, X. J., Glazier, C. (2003). Glazier. Real-time image thresholding based on sample space eduction and interpolation approach. *Computing in Civil Engineering, 17(4),* 264–272. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(264)
3. Lu, W., Zhao, D., Premebida, C., Zhang, L., Zhao, W. et al. (2023). Improving 3D vulnerable road user detection with point augmentation. *IEEE Transactions on Intelligent Vehicles, 8(5),* 3489–3505. https://doi.org/10.1109/TIV.2023.3246797
4. Manocha, D., Canny, J. F. (1994). Efficient inverse kinematics for general 6r manipulators. *IEEE Transactions on Robotics and Automation, 10(5),* 648–657. https://doi.org/10.1109/70.326569
5. Ju, B., Qu, W., Gu, Y. (2023). Boundary element analysis for mode III crack problems of thin-walled structures from micro- to nano-scales. *Computer Modeling in Engineering & Sciences, 136(3),* 2677–2690. https://doi.org/10.32604/cmes.2023.025886
6. 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. https://doi.org/10.1109/TITS.2016.2552248
7. Ukaegbu, U., Tartibu, L., Laseinde, T., Okwu, M., Olayode, I. (2020). A deep learning algorithm for detection of potassium deficiency in a red grapevine and spraying actuation using a raspberry PI3. *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (ICABCD)*, pp. 1–6. Durban, South Africa, IEEE. https://doi.org/10.1109/icABCD49160.2020.9183810
8. Su, Y., Gao, Y., Zhang, Y., Alvarez, J. M., Yang, J. et al. (2019). An illumination-invariant nonparametric model for urban road detection. *IEEE Transactions on Intelligent Vehicles, 4(1),* 14–23. https://doi.org/10.1109/TIV.2018.2886689
9. Wang, C., Xu, H., Zhou, Z., Deng, L., Yang, M. (2020). Shadow detection and removal for illumination consistency on the road. *IEEE Transactions on Intelligent Vehicles, 5(4),* 534–544. https://doi.org/10.1109/TIV.2020.2987440
10. Wang, Z., Cheng, G., Zheng, J. (2019). Road edge detection in all weather and illumination via driving video mining. *IEEE Transactions on Intelligent Vehicles, 4(2),* 232–243. https://doi.org/10.1109/TIV.2019.2904382

11. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *Proceedings of Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, pp. 234–241. Munich, Germany, Springer. https://doi.org/10.1007/978-3-319-24574-4_28
12. 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. https://doi.org/10.1109/TPAMI.2016.2644615
13. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.1409.1556
14. 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*, pp. 770–778. Las Vegas, NV, USA, IEEE.
15. 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*, pp. 801–818. Munich, Germany, Springer.
16. Chen, H., Lin, H. (2021). An effective hybrid atrous convolutional network for pixel-level crack detection. *IEEE Transactions on Instrumentation and Measurement, 70,* 1–12. https://doi.org/10.1109/TIM.2021.3075022
17. Hou, Q., Zhou, D., Feng, J. (2021). Coordinate attention for efficient mobile network design. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13713–13722. Nashville, TN, USA.
18. Cheng, H., Shi, X., Glazier, C. (2003). Real-time image thresholding based on sample space reduction and interpolation approach. *Journal of Computing in Civil Engineering, 17(4),* 264–272. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(264)
19. Dung, C. V., Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction, 99,* 52–58. https://doi.org/10.1016/j.autcon.2018.11.028
20. 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. https://doi.org/10.1016/j.neucom.2019.01.036
21. Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q. et al. (2018). Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing, 28(3),* 1498–1512. https://doi.org/10.1109/TIP.2018.2878966
22. Yu, F., Koltun, V., Funkhouser, T. (2017). Dilated residual networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 636–644. Honolulu, HI, USA.
23. Zhou, Q., Qu, Z., Cao, C. (2021). Mixed pooling and richer attention feature fusion for crack detection. *Pattern Recognition Letters, 145,* 96–102. https://doi.org/10.1109/TIP.2018.2878966
24. 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. https://doi.org/10.1109/TITS.2021.3106647
25. Yang, L., Huang, H., Kong, S., Liu, Y. (2023). A deep segmentation network for crack detection with progressive and hierarchical context fusion. *Journal of Building Engineering, 75,* 106886. https://doi.org/10.1016/j.jobe.2023.106886
26. Wang, X., Zhang, R., Sun, Y., Qi, J. (2019). Adversarial distillation for learning with privileged provisions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(3),* 786–797. https://doi.org/10.1109/TPAMI.2019.2942592
27. 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*, pp. 4510–4520. Salt Lake City, UT, USA.

28. Liao, J., Yue, Y., Zhang, D., Tu, W., Cao, R. et al. (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. https://doi.org/10.1109/TITS.2021.3138428
29. Zhang, X., Huang, H. (2022). LightAUNet: A lightweight fusing attention based UNet for crack detection. *Proceedings of 2022 7th International Conference on Image, Vision and Computing*, pp. 178–182. Xi'an, China. https://doi.org/10.1109/ICIVC55077.2022.9886163
30. Deng, J., Lu, Y., Lee, V. C. (2023). A hybrid lightweight encoder-decoder network for automatic bridge crack assessment with real-world interference. *Measurement, 216,* 112892. https://doi.org/10.1016/j.measurement.2023.112892
31. Hu, J., Shen, L., Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141. Salt Lake City, UT, USA.
32. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E. et al. (2021). TransUNet: Transformers make strong encoders for medical image segmentation. https://doi.org/10.48550/arXiv.2102.04306
33. Liu, H., Miao, X., Mertz, C., Xu, C., Kong, H. (2021). Crackformer: Transformer network for fine-grained crack detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3783–3792. Montreal, QC, Canada.
34. Yang, L., Bai, S., Liu, Y., Yu, H. (2023). Multi-scale triple-attention network for pixelwise crack segmentation. *Automation in Construction, 150,* 104853. https://doi.org/10.1016/j.autcon.2023.104853
35. Zhao, S., Zhang, G., Zhang, D., Tan, D., Huang, H. (2023). A hybrid attention deep learning network for refined segmentation of cracks from shield tunnel lining images. *Journal of Rock Mechanics and Geotechnical Engineering, 15(12),* 3105–3117. https://doi.org/10.1016/j.jrmge.2023.02.025
36. Han, C., Ma, T., Huyan, J., Huang, X., Zhang, Y. (2021). CrackW-Net: A novel pavement crack image segmentation convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems, 23(11),* 22135–22144. https://doi.org/10.1109/TITS.2021.3095507
37. Fan, Z., Li, C., Chen, Y., Wei, J., Loprencipe, G. et al. (2020). Automatic crack detection on road pavements using encoder-decoder architecture. *Materials, 13(13),* 2960. https://doi.org/10.3390/ma13132960
38. Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J. et al. (2020). Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials, 234,* 117367. https://doi.org/10.1016/j.conbuildmat.2019.117367
39. Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X. et al. (2019). Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems, 21(4),* 1525–1535. https://doi.org/10.1109/TITS.2019.2910595
40. Shi, Y., Cui, L., Oi, Z., Meng, F., Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems, 17(12),* 3434–3445. https://doi.org/10.1109/TITS.2016.2552248