RESEARCH ARTICLE

CrackNet: A novel multi-scale architecture for crack segmentation

Wubiao Zhu[1], Mengcai Ye[1], Jiawei Yin[2], Jingying Mo[1], Zhendi Ma[3], Ruibing Xie[1]*

1 Zhejiang Guangsha Vocational and Technical University of Construction, Zhejiang, China, **2** ShangHai University, Shanghai City, Shang Hai, China, **3** College of Computer Science and Technology, Zhejiang Normal University, Jinhua, China

* xrbxwx@163.com

OPEN ACCESS

Citation: Zhu W, Ye M, Yin J, Mo J, Ma Z, Xie R (2026) CrackNet: A novel multi-scale architecture for crack segmentation. PLoS One 21(4): e0346889. https://doi.org/10.1371/journal.pone.0346889

Editor: Musa Aydin, Samsun University: Samsun Universitesi, TÜRKIYE

Received: September 10, 2025

Accepted: March 25, 2026

Published: April 15, 2026

Data availability statement: All relevant data are available at https://github.com/guoguolord/CrackDataset.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Crack detection is essential for structural safety inspection but remains challenging due to noise, illumination variations, and complex backgrounds. In this paper, we propose CrackNet, a segmentation network specifically designed for concrete crack detection. CrackNet integrates three key modules: a lightweight multi-scale convolution enhancement block (LightMSCBlock) in the encoder to capture both local details and global context, a SAF attention module embedded in skip connections for scale-aware feature fusion and edge refinement, and a multi-scale feature fusion (MSFF) module in the decoder to enhance feature integration while reducing information loss. Extensive experiments on three public datasets—CFD, Crack500, and DeepCrack—demonstrate that CrackNet consistently outperforms state-of-the-art methods. Specifically, on CFD, F1 and IoU improve by 6.37% and 7.1% over SegFormer; on Crack500, F1 increases by 3.86% compared with MobileNetV3-UNet; and on DeepCrack, F1 and IoU gains reach 5.7% and 2.5%, respectively. Ablation studies further confirm the complementary effectiveness of LightMSCBlock, SAF, and MSFF. Overall, CrackNet achieves superior accuracy and robustness, showing strong potential for real-world engineering applications. The code is available at the following link: https://github.com/xzz-ya/CrackNet.git

1 Introduction

Surface cracks are one of the most common and significant types of damage in structural health monitoring. Timely and effective detection and assessment of these cracks are essential for ensuring the safety and durability of buildings and infrastructure [1,2]. Over time, structural materials may crack because of wear and tear, weathering, and heavy loads. If these cracks are not identified and addressed promptly, they may evolve into severe safety hazards and even lead to catastrophic failures. Traditional manual inspection methods are time-consuming, labor-intensive, and highly subjective, often resulting in low detection accuracy, which makes them

inadequate for the modern engineering demand for efficient and intelligent crack identification [3]. Therefore, accurately locating crack positions and reconstructing their geometric features has become a core technical challenge for ensuring structural safety [4–7].

Recently, crack detection has mainly relied on traditional image processing and handcrafted feature extraction. These approaches typically use grayscale, texture, edge, or frequency information combined with rule-based algorithms to isolate crack regions. Although efficient and easy to implement, they are sensitive to image quality, background complexity, and variations in crack appearance, limiting robustness and generalization in real-world scenarios. The Canny algorithm is widely adopted in edge detection methods for its strong noise suppression and accurate boundary localization [8]. Subsequent improvements in Gaussian filtering and thresholding have enhanced its sensitivity and robustness [9]. Sobel and Prewitt operators have also been used to localize cracks' edges. Morphological operations are often applied to eliminate minor artifacts and connect fragmented edges. Zhang et al. [10] combined morphological reconstruction with a distance-based shape descriptor to extract dark crack regions in complex backgrounds. Thresholding techniques also play a central role. Otsu's method is effective when there is a strong grayscale contrast between cracks and the background [11]. Later studies incorporated grayscale discrimination and regional histogram analysis to enhance segmentation in low-contrast cases [12]. Local adaptive thresholding and unsupervised methods such as maximum entropy have also been explored. Texture-based methods use statistical descriptors to represent crack patterns. The Gray-Level Co-occurrence Matrix (GLCM), for example, captures contrast and entropy and was used by Arya et al. [13] to develop an automatic classification method. Other descriptors, including Tamura features and the Histogram of Oriented Gradients (HOG), help capture directionality and local variations [14]. Fourier and wavelet transforms enable the separation of periodic and multiscale features in the frequency domain. Ranjbar Set al. [15] proposed a hybrid method combining wavelet features with deep neural networks for improved crack detection. Due to their directional sensitivity, Gabor filters have also been effective at enhancing noisy textures [16]. Zheng R et al. [17] proposed a hybrid technique that merges Histogram Equalization (HE) with Contrast-Limited Adaptive Histogram Equalization (CLAHE) to enhance image quality, making low-contrast images easier to see. Additional techniques, including entropy-based analysis and adaptive gamma correction, have been employed to highlight subtle crack details. Goo et al. [18] proposed Hybrid-Segmentor, which integrates CNN and Transformer to enable collaborative feature extraction and demonstrates excellent robustness. Tang et al. [19] introduced VM-UNet++, combining Mamba with the attention mechanism to strike a balance between accuracy and efficiency. Liu et al. [20] developed SCSegamba, featuring a lightweight design suitable for on-site detection.Template-matching methods detect cracks by comparing regions against predefined patterns. Kong Q et al. [21] enhanced this approach by integrating color features, improving the detection of repetitive structural cracks. Beyond traditional

image-domain techniques, graph models and energy optimization frameworks have been proposed to ensure edge continuity and regional coherence. Zhou Y [22] is enhancing segmentation accuracy and structural consistency.

With the rapid development of deep learning, CNN-based crack detection methods [23] have increasingly replaced traditional approaches, providing superior accuracy and robustness in complex environments. U-Net, initially developed for medical image segmentation, has proven effective for concrete crack detection due to its encoder-decoder architecture and skip connections [24] and has been widely adopted. Various U-Net variants have been proposed to improve segmentation performance, integrating multi-scale features and attention mechanisms to enhance the detection of fine cracks [25]. In object detection, Faster R-CNN is commonly used to localize cracks via its Region Proposal Network (RPN) [26]. Building on this, Mask R-CNN adds an instance segmentation branch, enabling pixel-level mask output and unifying detection and segmentation. Improved Mask R-CNN models have achieved better accuracy and generalization in bridge crack detection [27]. DeepLabv3 + incorporates Atrous Spatial Pyramid Pooling (ASPP) and a decoder to enhance multi-scale feature extraction. Its attention-based variants improve segmentation performance on concrete and asphalt cracks [28]. Backbone networks are also essential. ResNet addresses gradient vanishing through residual connections and is widely used in crack recognition [29]. DenseNet improves feature propagation with dense connectivity, enhancing segmentation efficiency [30]. VGG networks offer strong feature extraction due to their deep architecture, while EfficientNet balances precision and model size via compound scaling [31,32]. As research advances, transformer-based models have emerged for crack detection due to their global self-attention capabilities. Swin Transformer approaches work well even when they aren't pre-trained [33]. Enhanced Transformer models with multi-scale modules make it easier to segment fine cracks [34]. By integrating a dual encoder with wavelet feature enhancement and multi-stage supervision, this approach addresses the challenges of complex scene segmentation [35,36]. SegFormer, a Transformer-based segmentation model with a hierarchical encoder and lightweight decoder, offers high precision and efficiency. It has outperformed CNN models in detecting fine cracks on concrete and asphalt surfaces [37].

Although deep learning models have made significant progress in crack detection and segmentation, several challenges remain. Many existing networks still struggle to accurately detect fine or fragmented cracks, especially in noisy or low-contrast environments. Architectures such as U-Net and its variations frequently experience information loss during repeated downsampling, causing extremely thin cracks to disappear in deeper layers. Detection-based frameworks like Mask R-CNN [38] may also miss small crack patterns due to anchor constraints. Although transformer-based approaches such as Swin Transformer and SegFormer offer strong global modeling capabilities, they typically require large-scale datasets and substantial computational resources, limiting their practicality in resource-constrained settings. Despite this progress, two fundamental challenges remain insufficiently addressed. First, standard encoder–decoder architectures and lightweight backbones often incur severe feature loss during downsampling, making it difficult to retain low-contrast or hairline cracks. Second, most generic multi-scale and attention mechanisms are originally designed for high-level semantic segmentation and are not specifically adapted to the elongated, noisy, and discontinuous morphology of pavement cracks.

To address these challenges, this paper proposes a novel crack segmentation network, CrackNet, specifically designed for concrete crack recognition. The design of CrackNet draws on principles from information theory, aiming to maximize effective information flow while suppressing noise. CrackNet integrates multi-level feature extraction through three core modules. The LightMSCBlock is introduced into the encoder to capture both local details and global semantics through parallel multi-scale convolutional branches, thereby alleviating the feature loss commonly encountered in traditional convolutional backbones. The SAF attention modules are incorporated into the skip connections to provide crack-oriented recalibration, enhancing weak, fine-scale crack structures while suppressing irrelevant background noise. Finally, the MSFF module is integrated into the decoder to efficiently aggregate multi-scale contextual information with low computational overhead, improving the detection of long, thin, or discontinuous cracks.

2 Methodology

The network architecture proposed in this paper is designed explicitly for crack segmentation. The network has three innovative modules: LightMSCBlock, MSFF, and SAF. LightMSCBlock, the core module of the encoder, extracts multi-scale features from the input image and enhances the representation of important regions via an attention mechanism. The MSFF module, the main component of the decoder, fuses features from different convolutional operations and gradually restores the high-resolution output. The SAF attention module, which is used on skip connections, uses an attention mechanism to improve feature fusion across different scales. This makes sure that details and edge information are transferred correctly. Combining these three innovative modules enables the model to efficiently handle complex images in crack segmentation tasks while delivering high accuracy and robustness. The network architecture is shown in (Fig 1).

Overall, the proposed CrackNet forms a unified encoder–decoder framework in which multi-scale feature extraction, attention-guided feature transmission, and efficient feature fusion are tightly integrated. The LightMSCBlock enhances feature representation at different scales in the encoding stage, the SAF module preserves critical structural details during skip connections, and the MSFF module progressively refines and restores high-resolution features in the decoder. In the following section, we describe the experimental setup, datasets, and evaluation metrics, and present quantitative and qualitative results to validate the effectiveness of the proposed network.

2.1 Lightweight Multi-Scale Convolutional Enhancement Block (LightMSCBlock)

To simultaneously capture local details and global semantic information, we propose LightMSCBlock (Fig 2). The main design goal of this module is to improve feature representation via multiple convolutions and enhance feature output via attention, thereby improving robustness and generalization across different scenarios.

Given an input feature map $X \in \mathbb{R}^{B\times C\times H\times W}$, where B denotes the batch size, C the number of channels, and H,W the spatial resolution.The LightMSCBlock consists of two parallel depthwise separable convolution branches:

$$F_s = \varphi_{ds}(X; k = 3, d = 1), F_d = \varphi_{ds}(X; k = 3, d = 2) \tag{1}$$

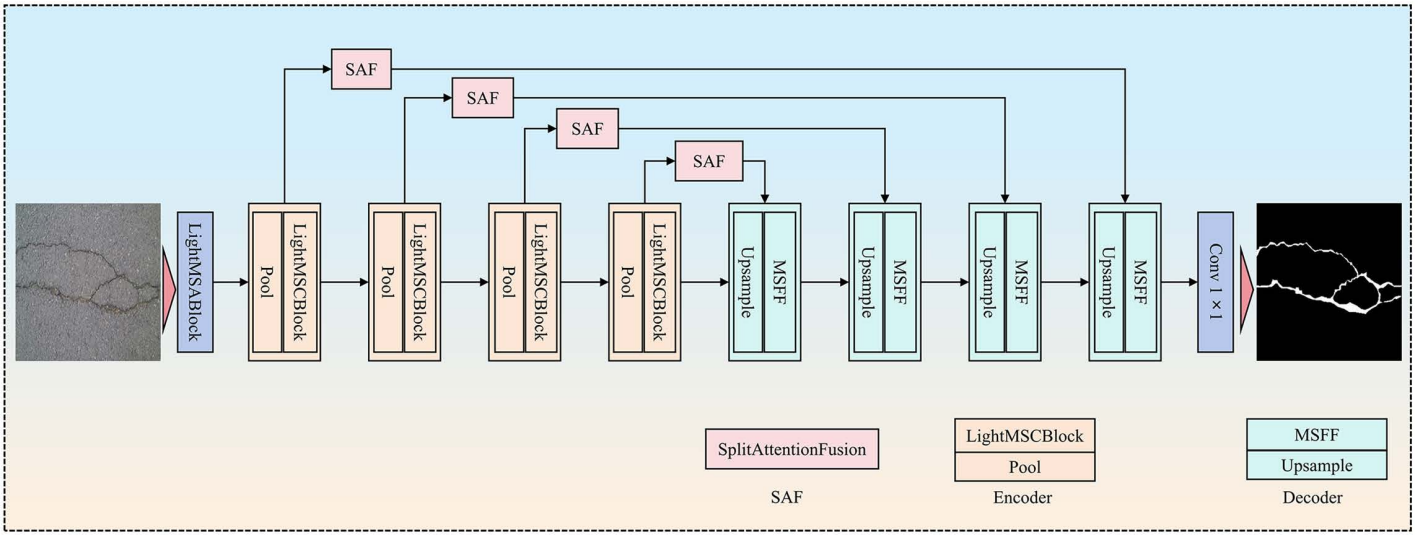

Fig 1. CrackNet architecture for crack segmentation, integrating the LightMSCBlock, MSFF, and SAF modules.

https://doi.org/10.1371/journal.pone.0346889.g001

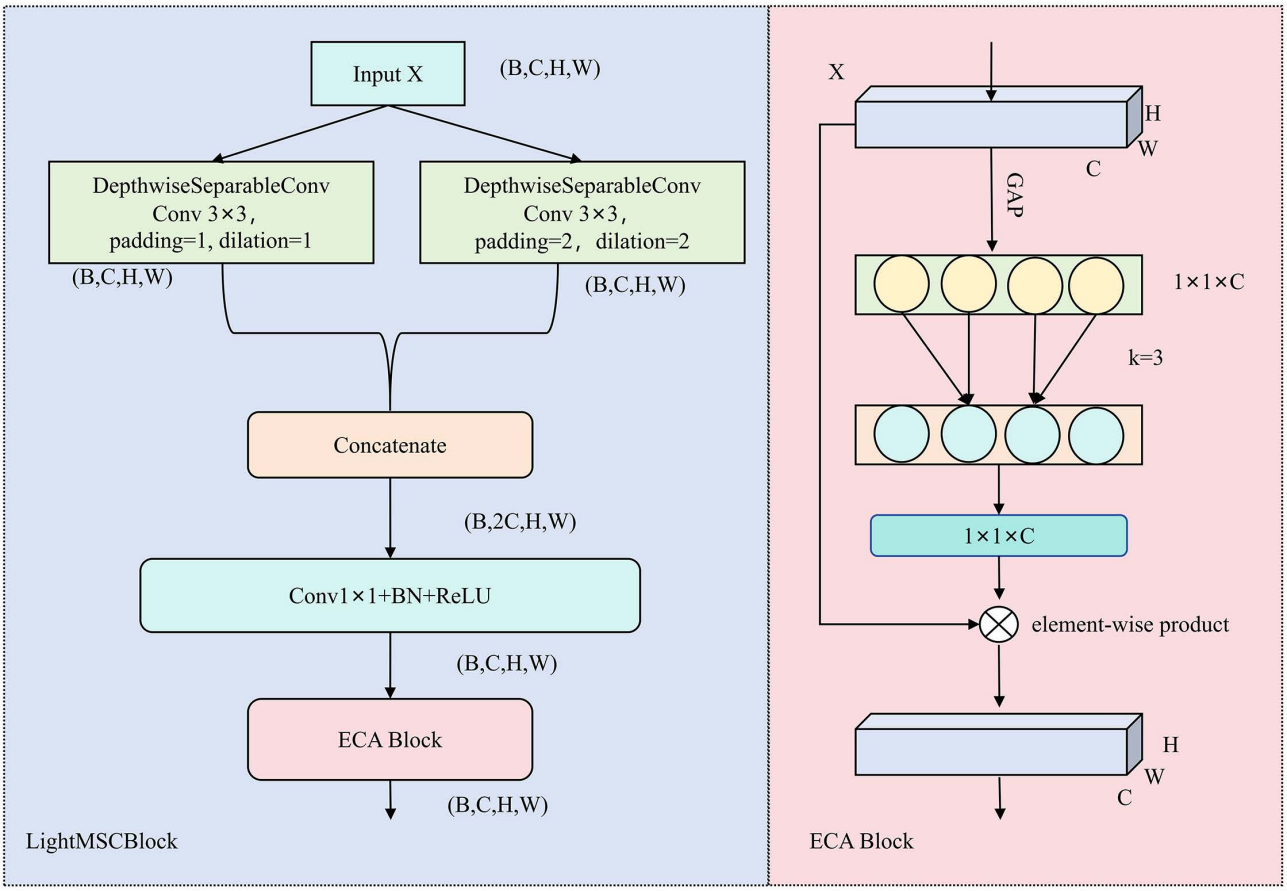

Fig 2. Lightweight Multi-Scale Convolutional Enhancement Block structure diagram.

https://doi.org/10.1371/journal.pone.0346889.g002

where $\varphi_{ds}(.)$ represents depthwise separable convolution, k is the kernel size, and d is the dilation rate. The standard branch F_s focuses on fine-grained local features, whereas the dilated branch F_d enlarges the receptive field to capture broader contextual information. This parallel design enables the module to simultaneously integrate local and global cues, providing discriminative representations for downstream tasks.

The outputs of both branches are concatenated and fused via a 1 × 1 convolution followed by batch normalization and a ReLU activation:

$$F = \sigma\left(BN\left(W_{1\times1}.[F_s; F_d]\right)\right) \tag{2}$$

where $[F_s; F_d]$ denotes concatenation, $W_1 \times W_1$ is the fusion kernel, BN denotes batch normalization, and $\sigma(.)$ is the ReLU function. This operation not only integrates multi-scale information but also suppresses redundancy and alleviates the computational burden associated with traditional multi-branch structures.

To further improve the informative features, an Efficient Channel Attention (ECA) [39] mechanism is integrated after the fusion. Specifically, channel descriptors are derived through global average pooling:

$$z_c = \frac{1}{H \times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_c\left(i,j\right), z \in \mathbb{R}^C \tag{3}$$

Where $F_c(i,j)$ denotes the activation of the c-th channel at position (i,j). A lightweight 1D convolution is then applied to model inter-channel dependencies:

$$s = \sigma\left(Conv1D\left(z;k\right)\right), s \in \mathbb{R}^C \tag{4}$$

with kernel size k and Sigmoid activation σ. Finally, the recalibrated feature is obtained by channel-wise multiplication:

$$F' = F \otimes s \tag{5}$$

where $\otimes$ denotes channel-wise multiplication.

LightMSCBlock seamlessly integrates lightweight computation, explicit multi-scale modeling, and efficient feature enhancement. Unlike traditional multi-scale convolutional architectures that introduce high computational cost, LightMSCBlock preserves sufficient representational power while extracting features at multiple receptive fields within a single block. In addition, compared to standard SE-style channel recalibration, the incorporation of ECA avoids information loss caused by dimensionality reduction and provides more flexible and accurate channel modeling. Beyond these general advantages, LightMSCBlock also differs fundamentally from the depthwise-separable inverted residual blocks in MobileNet and EfficientNet. First, MobileNet/EfficientNet capture multi-scale information implicitly through stacking blocks at different depths, whereas LightMSCBlock introduces explicit parallel multi-scale branches within a single block, enabling receptive-field aggregation before any downsampling takes place. Second, while EfficientNet applies squeeze-and-excitation to a single fused feature map, LightMSCBlock performs joint multi-scale feature fusion combined with lightweight attention, which better preserves the fine, low-contrast crack structures that are easily lost during downsampling. These characteristics make LightMSCBlock particularly suitable for crack detection tasks, where thin and subtle patterns require careful preservation of detailed information.

Algorithm 1. Lightweight Multi-Scale Convolutional Enhancement Block.

Input: Feature map $X \in \mathbb{R}^{B\times C\times H\times W}$.
Output: Enhanced feature map $F' \in \mathbb{R}^{B\times C\times H\times W}$.
Initialize standard depthwise separable branch $\varphi_{ds}(\cdot; k=3,\ d=1)$.
Initialize dilated depthwise separable branch $\varphi_{ds}(\cdot; k=3,\ d=2)$.
Initialize fusion module with 1×1 convolution $W_{1\times1}$, batch normalization BN, and ReLU activation $\sigma(\cdot)$.
Initialize Efficient Channel Attention (ECA) module with global average pooling, 1D convolution (kernel size k), and Sigmoid activation.
Given input feature map $X\in\mathbb{R}^{B\times C\times H\times W}$.
Compute the standard branch output using Eq. (1):
  $F_s = \phi_{ds}(X; k=3,\ d=1)$;
Compute the dilated branch output using Eq. (1):
  $F_d = \phi_{ds}(X; k=3,\ d=2)$;
Concatenate the outputs of the two branches along the channel dimension:
  $F_{cat} = [Fs;\ Fd]$;
Fuse multi-scale features using Eq. (2):
  $F=\sigma(BN(W_{1\times1}\ [F_s; F_d]))$;
Apply global average pooling on F to obtain channel descriptors using Eq. (3):
  for each channel $c=1$ to C_{out} do
    $z_c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} F_c(i,j),\ z\epsilon\mathbb{R}^C$;
  end;
Model inter-channel dependencies with a 1D convolution and Sigmoid activation using Eq. (4):

```
    s=σ(Conv1D(z;k)), s∈R^c;
  Reshape and broadcast s to match the dimensions of F.
  Recalibrate the fused feature map by channel-wise multiplication using Eq. (5):
    F' = F⊗s;
  return the enhanced feature map F';
```

2.2 SAF attention module

In this work, we propose the SAF attention module (Fig 3), a novel channel attention mechanism that enhances feature representation by efficiently combining multi-scale information. This module leverages adaptive average pooling and adaptive max pooling to extract features at different scales, followed by 1 × 1 and 3 × 3 convolutions to generate attention weights for each channel. Combining these operations allows the model to focus on the most relevant features while maintaining computational efficiency.

Given an input feature map $X \in \mathbb{R}^{B\times C\times H\times W}$, where B represents the batch size, C the number of channels, and H and W the spatial dimensions, the input is first split along the channel dimension:

$$X_1, X_2 = torch.split(X, \frac{C}{2}, \text{dim} = 1) \tag{6}$$

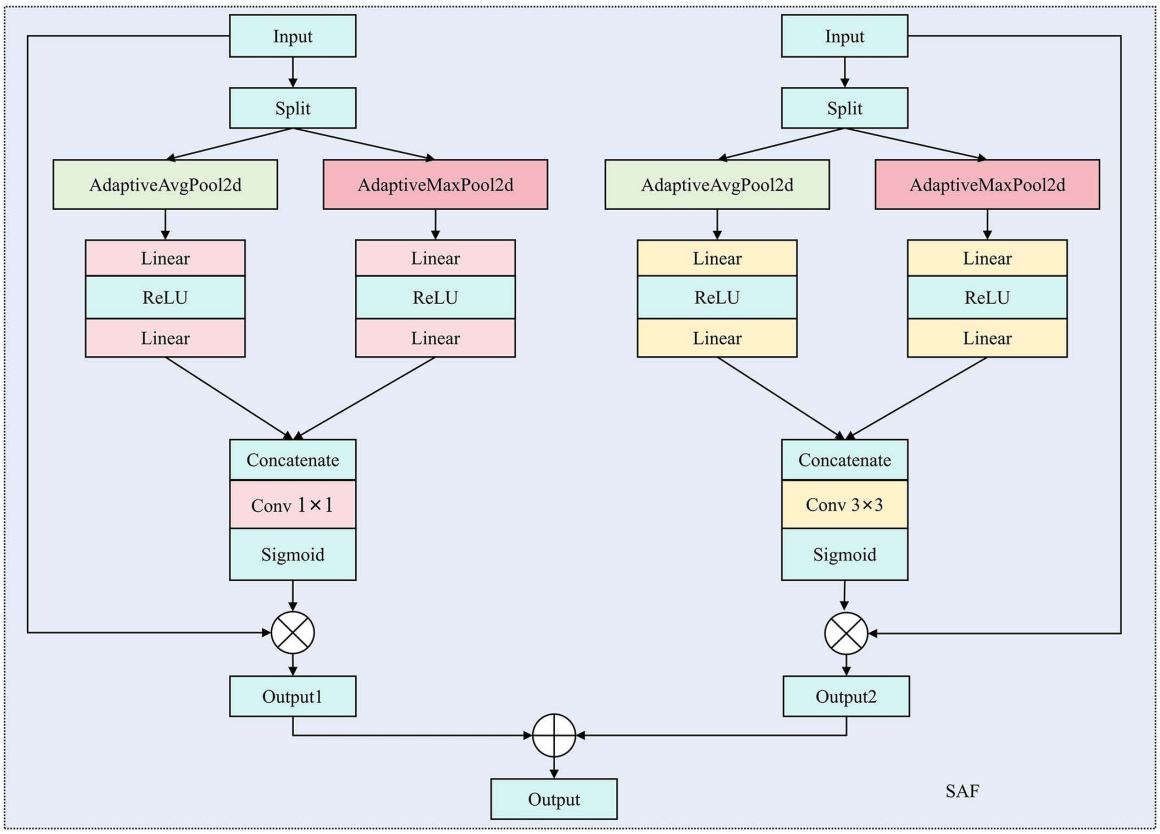

Fig 3. SAF attention structure diagram.

https://doi.org/10.1371/journal.pone.0346889.g003

where X_1 and X_2 each contain $\frac{C}{2}$ channels. The first branch applies adaptive average pooling to X_1, while the second branch applies adaptive max pooling to X_2:

$$avg\_out = avgpool\left(X_1\right),\ \max\_out = \max\ pool\left(X_2\right) \tag{7}$$

where avgpool and maxpool reduce the spatial dimensions of each branch to 1×1. These pooled outputs are then passed through fully connected layers for dimensionality reduction and recovery:

$$avgpool = FC_{avg}(avg\_out), \max\ pool = FC_{\max}(\max\_out) \tag{8}$$

where FC_{avg} and FC_{max} are composed of two linear layers (for reduction and recovery) and ReLU activations.

The outputs from both branches are concatenated:

$$merged = \left[avgout, \max\ out\right] \tag{9}$$

and passed through two separate convolutions 1×1 convolution and 3×3 convolution-to generate attention maps for different scales:

$$att_{11} = \left(Conv_{1\times1}\left(merged\right)\right), att_{33} = \sigma\left(Conv_{3\times3}\left(merged\right)\right) \tag{10}$$

where $\sigma(.)$ is the Sigmoid activation function, and conv1x1 and conv3x3 are the convolutional layers used to generate the attention weights at different scales. The final attention map is obtained by combining the two attention maps:

$$X' = X \otimes (att_{11} + att_{33}) \tag{11}$$

where $\otimes$ denotes element-wise multiplication, which applies the attention weights to the input feature map.

Unlike the traditional Squeeze-and-Excitation (SE) [40] module, the proposed SAF attention removes the dimensionality-reduction step during channel recalibration, thereby avoiding the loss of discriminative information. In SE, the use of fully connected layers for channel compression may discard critical responses, particularly when the feature maps are high-dimensional. SAF instead captures channel dependencies through multi-scale pooling combined with a convolution-based attention generator, preserving richer information and enabling more flexible and fine-grained channel modelling, which leads to improved performance. Beyond this advantage, our SAF modules are conceptually related to existing attention mechanisms such as CBAM and ECA, yet they are specifically designed for the characteristics of pavement cracks. CBAM applies channel and spatial attention sequentially and employs relatively heavy 2D pooling operations, increasing computational overhead while not being optimized for elongated, thin crack structures. ECA, which we also include as an internal baseline, performs only channel-wise modelling without incorporating spatial cues, making it less effective for capturing the fine geometry of cracks. In contrast, SAF introduce lightweight spatial attention with multi-scale pooling and joint feature weighting, enabling the network to selectively enhance long, thin, and discontinuous crack patterns while suppressing background noise. This design achieves a more favorable balance between accuracy and efficiency for crack detection compared to generic attention modules.

Algorithm 2. SAF Attention Module.

Input: Feature map $X \in \mathbb{R}^{B\times C\times H\times W}$.
Output: Refined feature map $X' \in \mathbb{R}^{B\times C\times H\times W}$.
1: Set split_channels=C/2.
2: Split the input feature map along the channel dimension using Eq. (6):

```
    X_1, X_2 = split(X, split_channels, dim=1);
  Apply adaptive average pooling and adaptive max pooling using Eq. (7):
    avg_out = avgpool(X_1), max_out = maxpool(X_2);
  Pass pooled features through branch-specific fully connected layers with reduction and recovery
using Eq. (8):
    avg_out = FC_avg(avg_out), max_out = FC_max(max_out);
  Concatenate the two branch outputs along the channel dimension using Eq. (9):
      merged = [avg_out; max_out];
  Generate multi-scale attention maps with 1×1 and 3×3 convolutions using Eq. (10):
      att_11 = Conv_1x1(merged), att_33 = Sigmoid(Conv_3x3(merged));
    where Sigmoid(·) denotes the Sigmoid activation function.
  Combine the attention maps and recalibrate the input feature map using Eq. (11):
    X' = X ⊗ (att_11 + att_33);
    where ⊙ denotes element-wise multiplication.
  return the refined feature map X';
```

2.3 Multi-Scale Feature Fusion Module (MSFF)

To explicitly model channel-wise dependencies in the fused multi-scale features, we integrate a Squeeze-and-Excitation (SE)-based channel attention mechanism into the MSFF module (Fig 4). Given the outputs of the two convolutional branches, $X_A, X_B \in \mathbb{R}^{B\times C\times H\times W}$, we first fuse them along the channel dimension to preserve the information from different scales:

$$X_f = Concat\left(X_A, X_B\right),\ X_A, X_B \in \mathbb{R}^{B\times C\times H\times W} \tag{12}$$

On top of X_f, the SE block performs a squeeze-excitation operation to produce channel attention weights. In the squeeze step, global average pooling is applied to aggregate spatial information of each channel into a scalar descriptor:

$$z_c = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} X_f^{(c)}(i,j),\ c = 1,\ldots,2C \tag{13}$$

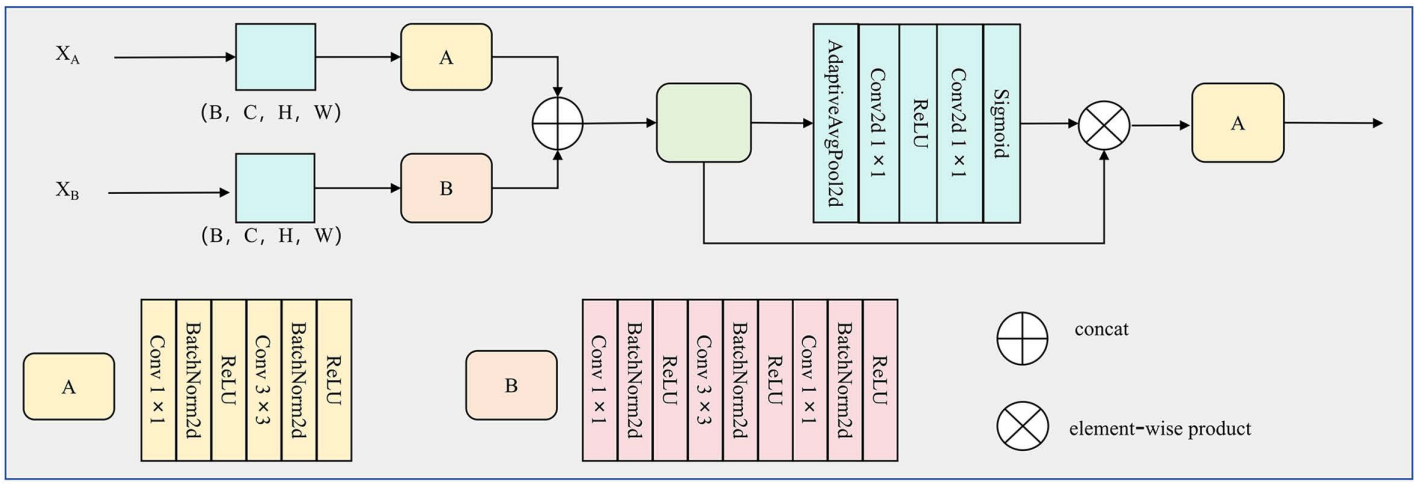

Fig 4. Multi-Scale Feature Fusion Module diagram.

https://doi.org/10.1371/journal.pone.0346889.g004

which yields $z \in \mathbb{R}^{B\times 2C\times 1\times 1}$. In the excitation step, z is passed through two successive 1×1 convolutions (equivalent to fully connected layers on the channel dimension) with a reduction ratio r, followed by ReLU and Sigmoid activations:

$$s = \sigma(W_2\delta(W_1 z)) \tag{14}$$

where $W_1 \in \mathbb{R}^{\frac{2C}{r}\times 2C}$ and $W_2 \in \mathbb{R}^{2C\times \frac{2C}{r}}$ are the learnable weights of the two 1×1 convolutions, δ denotes the ReLU function, and σ is the Sigmoid function. The resulting vector $s \in \mathbb{R}^{B\times 2C\times 1\times 1}$ encodes the importance of each channel. Finally, the attention weights are broadcast and applied to the fused feature map via channel-wise multiplication:

$$\widetilde{X}_f = X_f \bigodot s \tag{15}$$

where $\odot$ denotes element-wise multiplication. The reweighted feature $\widetilde{X}_f$ is then fed into the subsequent hybrid convolution layer to generate the final output of the MSFF module. In this way, the proposed channel attention mechanism adaptively emphasizes informative channels from different scales while suppressing redundant or irrelevant responses, leading to a more compact and discriminative multi-scale representation. Unlike traditional multi-scale fusion strategies that rely on repeatedly stacking convolutions or employing multiple scale-specific kernels, the MSFF module avoids redundant computation. By efficiently combining 1×1 and 3×3 convolutions with channel attention, MSFF extracts more precise multi-scale features while maintaining low computational overhead, offering a more efficient alternative to conventional multi-layer or multi-branch fusion designs. The proposed MSFF is conceptually related to the atrous spatial pyramid pooling (ASPP) module in DeepLab, as both aim to aggregate contextual information at multiple scales. However, ASPP employs several parallel dilated convolutions with large dilation rates, significantly increasing computational cost and potentially introducing gridding artifacts. Moreover, ASPP is designed for high-level semantic segmentation, whereas crack detection demands the preservation of fine, low-level geometric structures. In contrast, MSFF integrates lightweight multi-scale pooling and convolution with attention-based adaptive weighting of scale-specific features. This design not only reduces computation but also preserves local crack continuity more effectively, making the module particularly suitable for deployment in resource-constrained crack detection scenarios.

3 Experiments and results

3.1 Datasets

In this study, three datasets were used for model training: the CFD [41] dataset, the Crack500 [42] dataset, and the DeepCrack [43] dataset (Fig 5). The publicly available CFD dataset contains 118 crack images with a resolution of 480×320 pixels, which include noise such as water stains and shadows. The Crack500 dataset comprises 500 high-resolution images (2000×1500 pixels) of real road cracks. It features high scene complexity, including various crack types, diverse background materials, and lighting variations, making it suitable for evaluating model robustness in complex environments. The DeepCrack dataset comprises 591 images from multiple crack image collections, with an original resolution of 544×544 pixels. It includes a wide range of crack patterns and background noise, such as dust, scratches, and lighting interference, and is commonly used for training deep crack detection models.

Due to the relatively small size of the three datasets and the proposed model's requirement for input images of 512×512 pixels, data augmentation techniques were applied. These techniques included random cropping, brightness adjustment, contrast adjustment, and rotation. As a result, the CFD dataset was expanded to 2,000 images, the Crack500 dataset to 3,600 images, and the DeepCrack dataset to 3,000 images. Each dataset was first split into training, validation, and test sets in a ratio of 8:1:1. After the data split, augmentation was applied only to the training set, while the validation and test sets were resized deterministically to maintain their original content. This process ensures that no augmented images from the training set overlap with the validation or test sets, avoiding any potential data leakage.

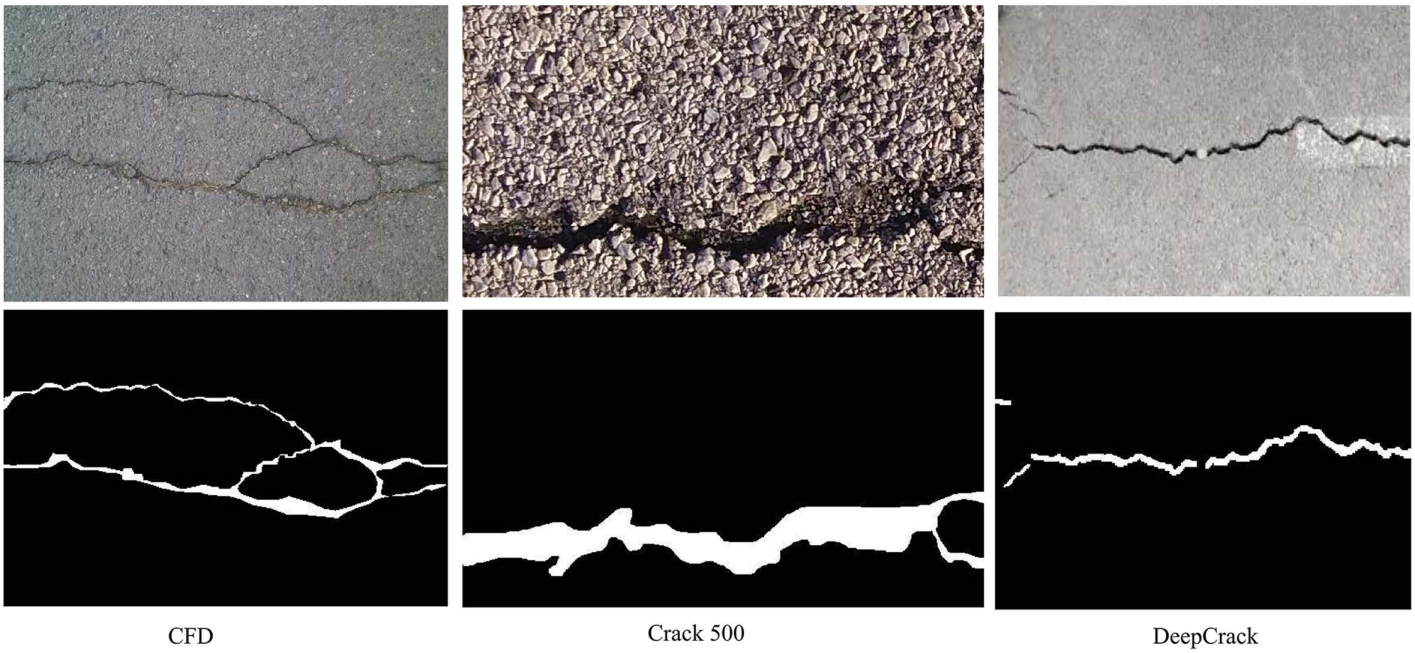

Fig 5. Dataset visualization (original images and label images of three types of data sets).

https://doi.org/10.1371/journal.pone.0346889.g005

3.2 Experimental setup and evaluation index

The proposed model was developed using Python 3.10 and built on the open-source deep learning framework PyTorch. The training process was accelerated using CUDA 12.6. The hardware environment consisted of an NVIDIA RTX 4090 GPU with 24 GB of VRAM. During training, the Adam optimizer was employed. The model was trained for 50 epochs with a batch size of 8 and an initial learning rate of 0.001. An early stopping mechanism was applied during training. Precision is defined as the ratio of correctly predicted positive samples to the total number of predicted positive samples:

$$precision = \frac{TP}{TP + FP} \tag{16}$$

The recall rate is calculated as the proportion of all actual targets correctly predicted:

$$recall = \frac{TP}{TP + FN} \tag{17}$$

Among them, TP is the number of targets correctly detected, FP is the number of targets incorrectly detected, and FN is the number of targets missed in the actual correct targets.

The F1 score combines precision and recall, providing a more comprehensive measure of the network's overall performance. It is calculated as a harmonic mean of these two metrics, as shown below:

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{18}$$

The standard definition of IoU is to calculate the ratio of the overlapping area to the union area between two regions (usually bounding boxes), as follows:

$$IoU = \frac{TP}{TP + FP + FN} \tag{19}$$

3.3 Loss function

To train the proposed model for crack detection, we employ a combined loss function that integrates both Cross-Entropy Loss and Dice Loss. The Cross-Entropy Loss is used for pixel-wise classification, encouraging the model to distinguish between crack and background pixels. The formula for Cross-Entropy Loss is given by:

$$Cross - Entropy\ Loss = -\sum_{i=1}^{N} [y_i log(p_i) + (1 - y_i) log(1 - p_i)] \tag{20}$$

Where y_i represents the ground truth label, p_i is the predicted probability, and N is the number of pixels. Since cracks often occupy a small portion of the image, class imbalance is a significant issue. To address this, Dice Loss is introduced, which enhances the model's ability to detect thin cracks by measuring the overlap between the predicted and ground truth masks. The Dice Coefficient is defined as:

$$Dice\ Coefficient = \frac{2 \sum_{i=1}^{N} y_i p_i}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} p_i} \tag{21}$$

The Dice Loss is then computed as:

$$Dice\ Loss = 1 - Dice\ Coefficient \tag{22}$$

The final loss function is the weighted sum of Cross-Entropy Loss and Dice Loss:

$$Total\ Loss = Cross - Entropy\ Loss + \lambda \cdot Dice\ Loss \tag{23}$$

Where λ is a hyperparameter controlling the relative weight of the Dice Loss. In our experiments, we set $\lambda = 1$, giving equal importance to both losses. This combined approach allows the model to balance pixel-wise accuracy with the preservation of fine, low-contrast crack structures.

3.4 Experimental results and analysis

3.4.1 Ablation experiment. We conducted ablation experiments on the CFD dataset to evaluate the contribution of each proposed module (Table 1). The base model, a standard U-Net architecture, achieved a precision of 0.7167, recall of 0.7258, F1 score of 0.7213, and IoU of 0.5641. Introducing the LightMSCBlock (A) improved multi-scale feature extraction, while the SAF attention module (B) enhanced feature discrimination by suppressing background noise. The MSFF module (C) further strengthened multi-scale contextual aggregation during decoding. Each module individually offered performance gains compared to the base model.

We also evaluated pairwise combinations of the modules. The A + C configuration improved recall and F1 by enhancing both encoder-level multi-scale representation and decoder-level contextual fusion. The B + C combination further boosted recall (0.7946) and F1 (0.7413), demonstrating that attention-guided feature recalibration (B) and multi-scale fusion (C)

Table 1. Ablation experiments of the network on CFD dataset.

model	Pr	recall	F1	Iou
Base	0.7167	0.7258	0.7213	0.5641
Base+A	0.7233	0.7471	0.7201	0.5741
Base+B	0.7196	0.7323	0.7243	0.5512
Base+C	0.7172	0.7412	0.7256	0.5638
Base+A+B	0.7285	0.7561	0.7289	0.5987
Base+A+C	0.7243	0.7689	0.7358	0.5941
Base+B+C	0.7266	0.7946	0.7413	0.5987
Base+A+B+C	**0.7325**	**0.8128**	**0.7550**	**0.6064**

https://doi.org/10.1371/journal.pone.0346889.t001

complement each other effectively. When all three modules were integrated (A+B+C), the full model achieved the best performance, yielding a precision of 0.7325, recall of 0.8128, F1 score of 0.7550, and IoU of 0.6064. This substantial improvement indicates that the three modules act synergistically: LightMSCBlock preserves fine crack details during downsampling, SAF enhances crack-relevant spatial-channel responses in skip connections, and MSFF improves structural continuity during upsampling. Overall, the results demonstrate that each module contributes to segmentation accuracy from different stages of the network pipeline, and their combination yields robust and significant performance gains, confirming the effectiveness of the proposed CrackNet architecture for crack segmentation.

Table 2 summarizes the ablation results on the Crack500 dataset. The baseline U-Net model achieved a precision of 0.6576, recall of 0.7542, F1 score of 0.7026, and IoU of 0.5416. Introducing the LightMSCBlock (A) yielded consistent improvements across all metrics, demonstrating its ability to enhance multi-scale feature representation. Incorporating the SAF attention module (B) slightly decreased precision to 0.6543 but improved recall and F1, indicating that the attention mechanism strengthens crack-focused feature responses while marginally increasing false positives. The MSFF module (C) further improved recall and F1 while maintaining stable precision, reflecting enhanced multi-scale contextual aggregation in the decoder. Pairwise combinations of the modules produced additional gains. The A+C configuration achieved higher recall (0.7851) and F1 (0.7546), showing the complementary effect of encoder-level multi-scale enhancement and decoder-level fusion. The B+C combination yielded further improvement, with an IoU of 0.6104, confirming that attention-guided recalibration (B) and multi-scale feature fusion (C) jointly strengthen crack continuity and structural consistency. When all three modules were integrated (A+B+C), the model achieved the highest performance across all metrics, with a notable increase in precision to 0.8765 and stable recall at 0.7964. This substantial precision increase can be attributed to the synergistic effect of the three modules: LightMSCBlock reduces information loss during downsampling, SAF suppresses background noise and enhances discriminative crack responses, and MSFF improves multi-scale

Table 2. Ablation experiments of network on Crack500 dataset.

model	Pr	recall	F1	Iou
Base	0.6576	0.7542	0.7026	0.5416
Base+A	0.6579	0.7643	0.7431	0.5712
Base+B	0.6543	0.7586	0.7325	0.5645
Base+C	0.6589	0.7557	0.7423	0.5632
Base+A+B	0.6601	0.7754	0.7504	0.5986
Base+A+C	0.6984	0.7851	0.7546	0.6013
Base+B+C	0.7578	0.7896	0.7612	0.6104
Base+A+B+C	**0.8765**	**0.7964**	**0.7636**	**0.6175**

https://doi.org/10.1371/journal.pone.0346889.t002

context aggregation. Together, these mechanisms significantly reduce false positives while maintaining strong crack detection capability, resulting in improved F1 (0.7636) and IoU (0.6175).

Table 3 presents the ablation results of different module combinations on the DeepCrack dataset. The baseline U-Net model achieved a precision of 0.7014, recall of 0.7965, F1 score of 0.7459, and IoU of 0.5948. When incorporating the LightMSCBlock (A), performance improved across all metrics, reaching an F1 of 0.7944 and IoU of 0.6476, indicating that encoder-level multi-scale representation substantially enhances feature retention and crack boundary detection. Adding the SAF module (B) also improved overall performance, achieving an F1 of 0.7856 and IoU of 0.6561. This result suggests that attention-based feature recalibration strengthens crack saliency while maintaining robust spatial segmentation. The MSFF module (C) provided additional benefits in multi-scale fusion, increasing IoU to 0.6778 and yielding more consistent segmentation outputs across varying crack widths and textures. Pairwise module combinations further validate the complementary nature of these components. The A + B configuration increased F1 to 0.8131 and IoU to 0.7070, demonstrating the synergy between multi-scale enhancement and attention-guided feature refinement. The A + C combination performed even better, with an F1 of 0.8294 and IoU of 0.7156, reflecting improved encoder–decoder coupling through stronger multi-scale fusion. The B + C combination achieved one of the highest pairwise results, with an F1 of 0.8369 and IoU of 0.7196, confirming that the combination of attention and multi-scale fusion effectively improves crack continuity and reduces misclassification. When all three modules were integrated (A + B + C), the model achieved the best performance: 0.8654 precision, 0.8625 recall, 0.8423 F1, and 0.7275 IoU. This demonstrates that the three modules address complementary aspects of crack segmentation—A enhances fine feature retention, B improves crack-focused recalibration, and C strengthens multi-scale integrative reasoning—resulting in a robust and comprehensive performance improvement. Overall, the ablation results confirm that each module contributes meaningfully to segmentation quality, and their full integration yields the most accurate, consistent, and reliable detection performance on the DeepCrack dataset.

3.4.2 Comparison experiment. To comprehensively evaluate the performance of different semantic segmentation models in crack detection tasks, we selected several mainstream architectures, including Unet, VGG16UNet, MobileV3Unet, SegFormer, FCN_ResNet50, and our proposed model, Ours. Comparative experiments were conducted on three publicly available crack datasets: CFD, Crack500, and DeepCrack. We adopted four quantitative metrics to ensure objective and thorough evaluation: precision, recall, F1, and intersection over union (IoU). Through these experiments, we aim to validate each model's strengths and its applicability to crack detection.

Table 4 presents a comparison of various models on the CFD dataset, evaluated using Pr, recall, F1, and IoU. The baseline UNet model achieves a precision of 0.7167, a recall of 0.7258, an F1 of 0.7213, and an IoU of 0.5641, serving as the benchmark for comparison. VGG16UNet shows a slight decrease in precision (0.6793) but achieves higher recall (0.7538) than UNet, resulting in a slightly lower F1 (0.7147) and a slight reduction in IoU (0.556). This indicates that while

Table 3. Ablation experiments of network on DeepCrack dataset.

model	Pr	recall	F1	Iou
Base	0.7014	0.7965	0.7459	0.5948
Base+A	0.7687	0.8021	0.7944	0.6476
Base+B	0.7546	0.7945	0.7856	0.6561
Base+C	0.7478	0.7906	0.7636	0.6778
Base+A+B	0.8346	0.8107	0.8131	0.7070
Base+A+C	0.8446	0.8303	0.8294	0.7156
Base+B+C	0.8512	0.8498	0.8369	0.7196
Base+A+B+C	**0.8654**	**0.8625**	**0.8423**	**0.7275**

https://doi.org/10.1371/journal.pone.0346889.t003

Table 4. Comparison test of network on CFD dataset.

model	pr	recall	F1	Iou
Unet	0.7167	0.7258	0.7213	0.5641
VGG16UNet	0.6793	0.7538	0.7147	0.5560
ResUNet	0.6743	0.7689	0.6911	0.5497
AttentionUNet	0.6987	0.7846	0.7102	0.5649
MobileV3Unet	0.7529	0.6208	0.6805	0.5157
SegFormer	0.6433	0.7471	0.6913	0.5352
fcn_resnet50	0.5775	0.7617	0.6570	0.4891
DeepCrack	0.7471	0.7956	0.7311	0.5945
OUrs	**0.7325**	**0.8128**	**0.7550**	**0.6064**

https://doi.org/10.1371/journal.pone.0346889.t004

VGG16UNet improves recall, it sacrifices some precision and spatial consistency. MobileV3Unet excels in precision (0.7529) but suffers from a significantly lower recall (0.6208) and F1 (0.6805), with a considerably lower IoU (0.5157). This suggests that while MobileV3Unet can correctly classify positives, it fails to capture many true positives, severely impacting its segmentation performance. SegFormer achieves a moderate Pr (0.6433) and recall (0.7471), with an F1 of 0.6913 and IoU of 0.5352, indicating decent recall performance but lower precision and overall segmentation accuracy. FCN_resnet50 shows the lowest precision (0.5775) but a high recall (0.7617), resulting in the lowest F1 (0.6570) and IoU (0.4891), suggesting that while fcn_resnet50 excels at detecting true positives, it struggles with accurate segmentation boundaries. Finally, ours outperforms all other models, achieving a precision of 0.7325, a recall of 0.8128, an F1 of 0.7550, and an IoU of 0.6064, demonstrating the best balance between classification accuracy and spatial consistency. In summary, while other models may excel in one or two metrics, ours provides a more well-rounded performance, achieving superior results across all evaluated metrics, making it the most effective model for the CFD dataset. Fig 6 shows a visual comparison of the model with other models on the CFD dataset.

Table 5 compares various models on the Crack500 dataset, evaluated using Pr, recall, F1, and IoU. The baseline UNet model achieves a precision of 0.6576, recall of 0.7542, an F1 of 0.7026, and an IoU of 0.5416, providing a solid reference for comparison. VGG16UNet achieves a precision of 0.6807, a recall of 0.7067, an F1 of 0.6934, and an IoU of 0.5307, showing improved precision but lower recall and IoU compared to UNet, indicating that while VGG16UNet improves precision, it sacrifices recall and spatial consistency. MobileV3Unet shows a precision of 0.6966, a recall of 0.7559, an F1 of 0.725, and an IoU of 0.5687, demonstrating an improvement in recall and F1 over VGG16UNet, with a higher IoU than both VGG16UNet and UNet, suggesting better overall performance in segmentation tasks. SegFormer achieves a precision of 0.8334, a recall of 0.6529, an F1 of 0.7322, and an IoU of 0.5776, which shows the highest precision among all models tested, but at the cost of significantly lower recall, resulting in an imbalance in classification and recall. fcn_resnet50 achieves a precision of 0.8236, a recall of 0.6839, an F1 of 0.7473, and an IoU of 0.5966, showing strong performance in precision and recall, with relatively high F1 and IoU, but still slightly behind MobileV3Unet and Ours in terms of overall segmentation accuracy. Finally, Ours outperforms all other models, achieving a precision of 0.8765, a recall of 0.7964, an F1 of 0.7636, and an IoU of 0.6175, demonstrating the best balance among precision, recall, and spatial accuracy. In summary, Ours achieves the best overall performance on the Crack500 dataset, with significant improvements in recall, F1, and IoU, outperforming all other models across all evaluated metrics, highlighting its effectiveness in segmentation tasks. Fig 7 visually compares the model with other models on the Crack500 dataset.

Table 6 compares DeepCrack models evaluated using Pr, recall, F1, and IoU. The baseline UNet model achieves precision of 0.7014, recall of 0.7965, F1 of 0.7459, and IoU of 0.5948, serving as the reference point for comparison. VGG16UNet demonstrates significant improvements, achieving a precision of 0.8324, a recall of 0.8293, an F1 of 0.8156, and an IoU of 0.6652, outperforming UNet in classification and recall. MobileV3Unet achieves precision of 0.8236, recall of

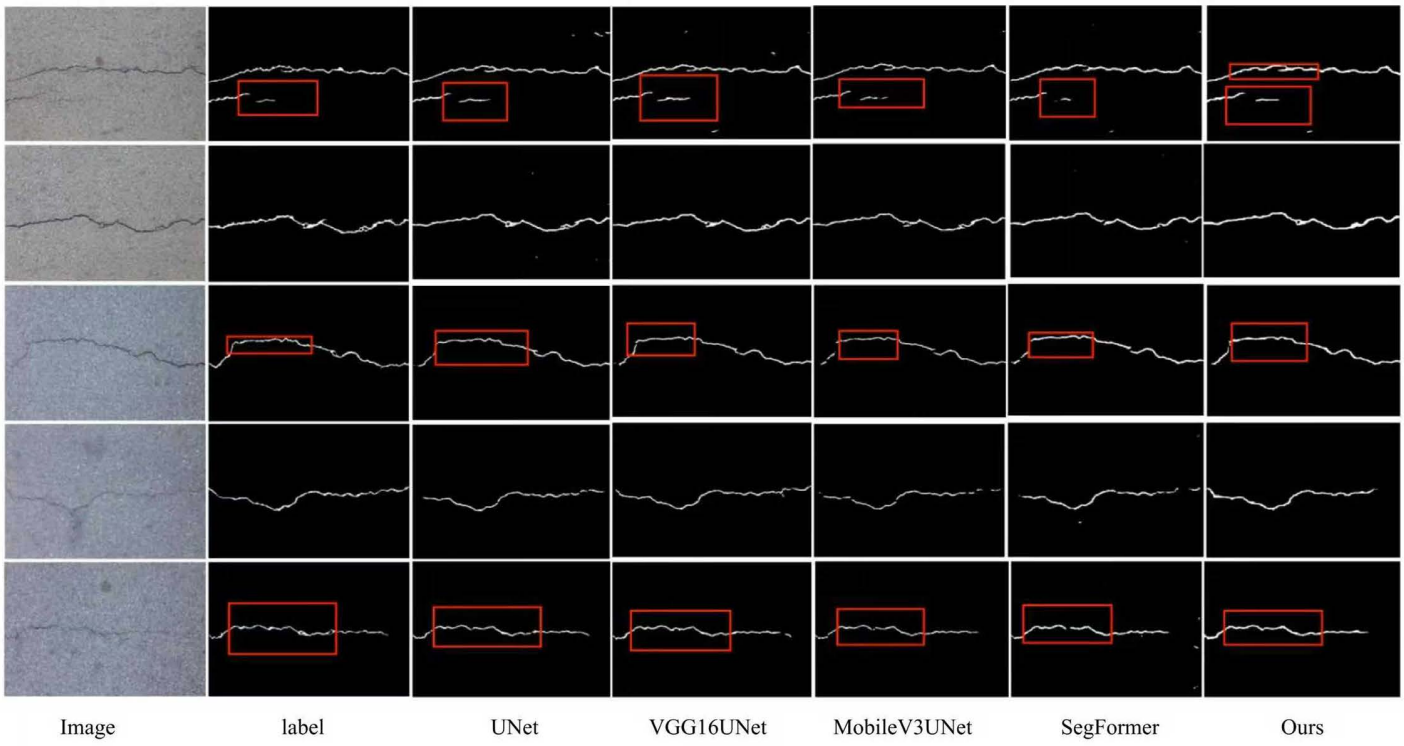

Fig 6. Visualization of the model compared with other models on the CFD dataset.

https://doi.org/10.1371/journal.pone.0346889.g006

Table 5. Comparison test of network on Crack500 dataset.

model	pr	recall	F1	Iou
Unet	0.6576	0.7542	0.7026	0.5416
VGG16UNet	0.6807	0.7067	0.6934	0.5707
ResUNet	0.6821	0.7164	0.7243	0.5369
AttentionUNet	0.6832	0.7236	0.7169	0.5443
MobileV3Unet	0.6966	0.7559	0.7250	0.5687
SegFormer	0.8334	0.6529	0.7322	0.5776
Fcn_resnet50	0.8236	0.6839	0.7473	0.5966
DeepCrack	0.8463	0.7489	0.7529	0.6111
OUrs	**0.8765**	**0.7964**	**0.7636**	**0.6175**

https://doi.org/10.1371/journal.pone.0346889.t005

0.8454, F1 of 0.7856, and IoU of 0.7021, outperforming both VGG16UNet and UNet in recall and IoU, indicating a better balance between precision and spatial accuracy. SegFormer shows a precision of 0.7602, recall of 0.8459, F1 of 0.8008, and IoU of 0.6678, performing similarly to MobileV3Unet in recall but with slightly lower precision and IoU. fcn_resnet50 achieves a precision of 0.8312, a recall of 0.7623, an F1 of 0.8049, and an IoU of 0.6735, showing strong performance in precision and F1, but lower recall than MobileV3Unet. Finally, Ours outperforms all other models, achieving a precision of 0.8654, a recall of 0.8625, an F1 of 0.8423, and an IoU of 0.7275, demonstrating the best balance among precision, recall, and spatial accuracy. In summary, Ours performs best on the DeepCrack dataset, surpassing all other models

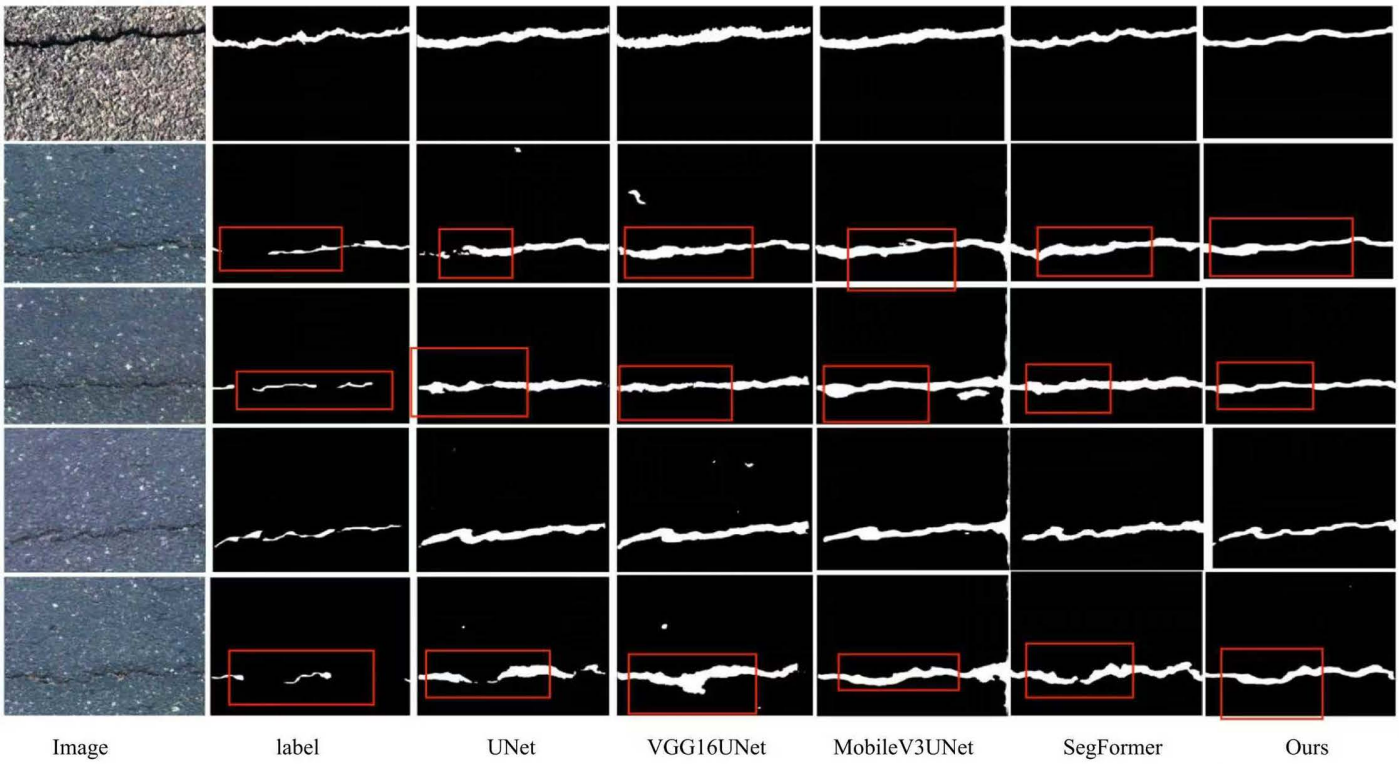

Fig 7. Visualization of the model compared with other models on the Crack500 dataset.

https://doi.org/10.1371/journal.pone.0346889.g007

Table 6. Comparison test of network on DeepCrack dataset.

model	pr	recall	F1	Iou
Unet	0.7014	0.7965	0.7459	0.5948
VGG16UNet	0.8324	0.8293	0.8156	0.6652
ResUNet	0.8354	0.8348	0.8179	0.6541
AttentionUNet	0.8226	0.8354	0.8263	0.6897
MobileV3Unet	0.8236	0.8454	0.7856	0.7021
SegFormer	0.7602	0.8459	0.8008	0.6678
fcn_resnet50	0.8312	0.7623	0.8049	0.6735
DeepCrack	0.8346	0.8515	0.8215	0.7112
OUrs	**0.8654**	**0.8625**	**0.8423**	**0.7275**

https://doi.org/10.1371/journal.pone.0346889.t006

across all metrics, highlighting its superior ability to achieve high classification accuracy and precise segmentation. Fig 8 visually compares the model with other models on the DeepCrack dataset.

3.4.3 Comparative experiment between SAF attention and other attention. Table 7 summarizes the comparative results of different attention mechanisms and the proposed SAF attention on three crack segmentation datasets. The CFD dataset's SE, CAM, and CE performances were relatively close. In contrast, SAF attention produced a slightly higher F1 (0.755) than the others, suggesting a more favorable balance between precision and recall. For the DeepCrack dataset, SAF attention achieved an accuracy of 0.8654, marginally higher than the alternative mechanisms, with a recall similar

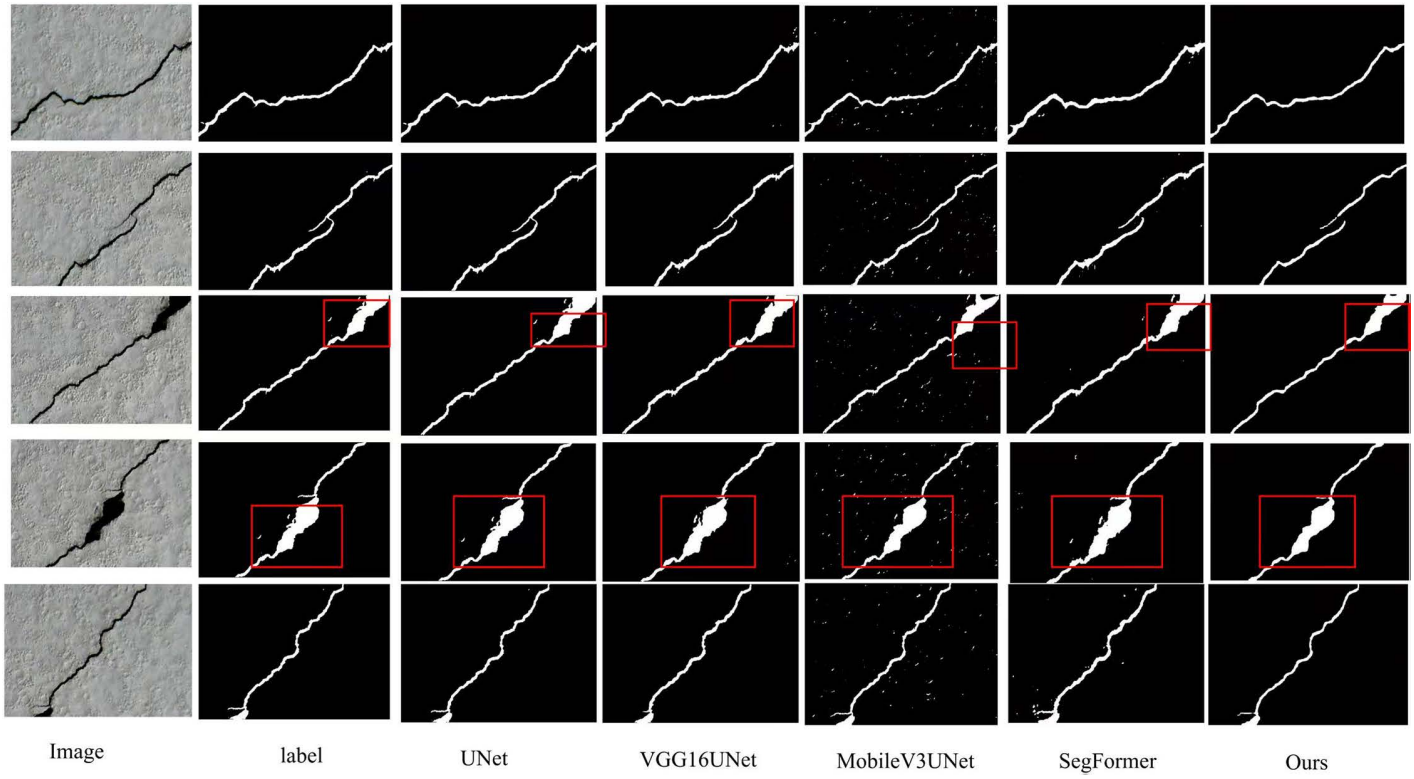

Fig 8. Visualization of the model compared with other models on the DeepCrack dataset.

https://doi.org/10.1371/journal.pone.0346889.g008

Table 7. Comparative experiments between different attention and SAF attention on three datasets.

	CFD			DeepCrack			Crack500		
Method	**Pr**	**Recall**	**F1**	**Pr**	**Recall**	**F1**	**Pr**	**Recall**	**F1**
with/SE	0.7301	0.8111	0.7356	0.8538	0.8056	0.8203	0.8590	0.7854	0.7531
with/CAM	0.7321	0.8124	0.7421	0.8524	0.8123	0.8208	0.8474	0.7824	0.7534
with/CE	0.7213	0.8107	0.7422	0.8540	0.8111	0.8178	0.8452	0.7825	0.7521
with/SAF	**0.7325**	**0.8128**	**0.755**	**0.8654**	**0.8126**	**0.8214**	**0.8765**	**0.7964**	**0.7636**

https://doi.org/10.1371/journal.pone.0346889.t007

to theirs, leading to the highest F1 (0.8214) among all methods. On the Crack500 dataset, SAF attention again achieved better values, with both precision (0.8765) and recall (0.7964) higher than those of the other mechanisms, resulting in an F1 of 0.7636, compared to approximately 0.753 for SE, CAM, and CE. Overall, these findings indicate that although differences across methods are not always significant, SAF attention tends to yield more consistent improvements in segmentation performance, particularly on datasets with diverse crack scales and complex backgrounds, thereby enhancing the model's robustness under varying conditions.

Fig 9 illustrates the attention visualization results of different mechanisms on a representative crack image. The SE and CAM modules emphasize localized regions, capturing only parts of the crack and occasionally overlooking its continuity. CE produces a more dispersed activation that covers broader areas but lacks precise alignment with the crack boundaries. In contrast, the SAF attention provides a more coherent, continuous focus across the entire crack, effectively highlighting both the central structure and its extensions. These qualitative observations are consistent with the quantitative

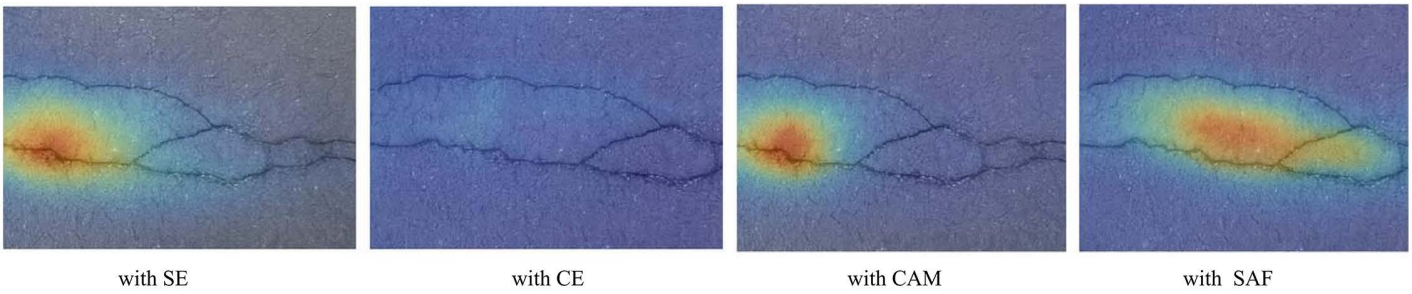

Fig 9. Visualization of attention responses on a sample crack image using different mechanisms. From left to right and top to bottom: SE, CAM, CE, and SAF. The heatmaps highlight the regions where the model focuses during crack segmentation.

https://doi.org/10.1371/journal.pone.0346889.g009

improvements reported in Table 7, suggesting that incorporating SAF attention enhances the model's ability to capture long-range dependencies and represent crack structures more completely.

3.4.4 Model complexity comparison. Fig 10 compares the computational complexity of different segmentation networks in terms of FLOPs and parameter counts. Classic or non-UNet models (blue circles) span a wide range of complexity: lightweight architectures such as MobileNetV3-UNet and SegFormer-B1 are located in the lower-left region, while VGG16-UNet lies in the upper-right corner with the highest FLOPs and parameters. UNet variants (orange diamonds), including Residual UNet, Attention UNet, R2UNet and Nested UNet, are mainly distributed in the upper-right area, showing that they generally incur larger computational and memory costs.

Our method (red square) is positioned noticeably to the left of most UNet variants, with only 44.5G FLOPs and 18.32M parameters. This complexity is much lower than that of typical UNet-based models, yet, as shown in later experiments,

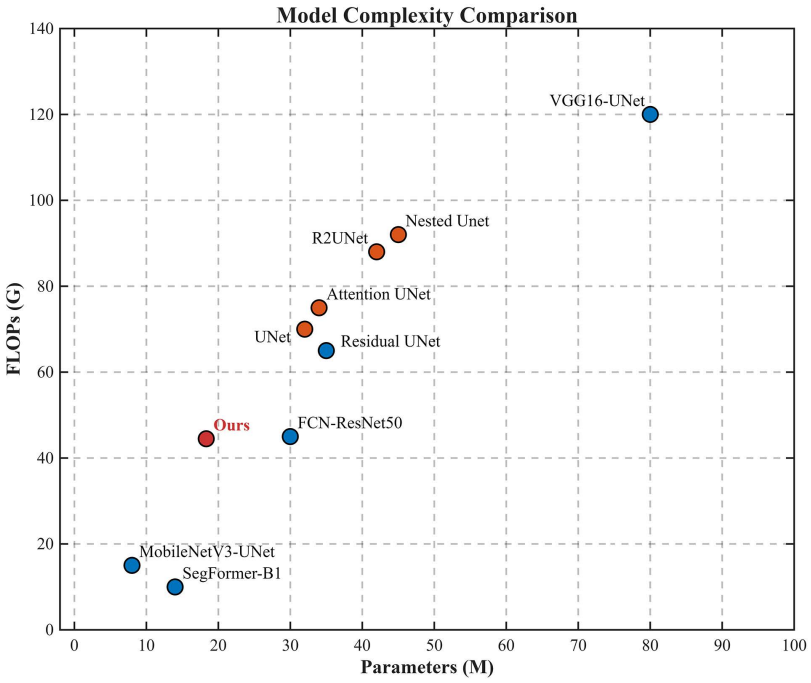

Fig 10. Model complexity comparison in terms of FLOPs and parameters.

https://doi.org/10.1371/journal.pone.0346889.g010

the proposed network maintains competitive or superior segmentation performance. Therefore, the figure highlights that our model achieves a better balance between accuracy and efficiency, which is advantageous for deployment in resource-constrained or real-time applications.

3.4.5 Statistical significance analysis. To verify that the performance improvements of the proposed CrackNet are robust rather than resulting from random initialization, we conducted paired t-tests over repeated runs on all three datasets, with the corresponding p-values reported in Table 8. On the CFD dataset, both F1 and precision achieve strong statistical significance at $\alpha < 0.01$ ($p = 0.0043$ and $p = 0.0086$, respectively), while recall is significant at $\alpha < 0.05$ ($p = 0.0187$). These results indicate that the improvements on CFD are highly stable, though recall exhibits slightly higher variability due to the sensitivity of thin-crack detection to local noise. For the DeepCrack dataset, all three metrics show significance at $\alpha < 0.05$, with precision and recall further reaching $\alpha < 0.01$ ($p = 0.0064$ and $p = 0.0079$). This confirms that the proposed modules consistently reduce false positives and false negatives even under the challenging conditions and structural diversity in DeepCrack. On the Crack500 dataset, the improvements in F1 and recall are strongly significant at $\alpha < 0.01$ ($p = 0.0063$ and $p = 0.0011$), while precision remains significant at $\alpha < 0.05$ ($p = 0.0152$), reflecting expected fluctuations in false-positive rates across repeated runs. Overall, the statistical significance analysis demonstrates that the observed performance gains are consistent, reproducible, and not attributable to random chance, reinforcing the robustness and reliability of CrackNet across different datasets and crack characteristics.

3.4.6 Failure case analysis. Fig 11 presents representative failure cases of the proposed method, where the model produces incorrect predictions under challenging surface conditions. In this example, the original image contains highly textured aggregate patterns with strong intensity variations that visually resemble crack structures. Although the ground truth annotation indicates the presence of a crack only in a limited region, the model incorrectly predicts several false-positive responses in the surrounding textured background.

This failure can be attributed to the fact that coarse aggregates and stone boundaries exhibit elongated and high-contrast edges that are visually similar to cracks, making them difficult to distinguish using appearance cues alone. In such scenarios, the model tends to confuse crack-like textures with actual cracks, leading to over-segmentation. These cases highlight a current limitation of the proposed method when dealing with complex backgrounds containing strong texture noise. Incorporating additional contextual constraints or material-aware features may help alleviate this issue in future work.

3.4.7 Exploring the impact of ECA attention on the LightMSCBlock module. Table 9 presents the ablation study results comparing LightMSCBlock without and with the ECA module across three crack segmentation datasets. On the

Table 8. Statistical significance (p-values) of performance improvements based on paired t-tests over repeated runs. ✓ indicates significance at $\alpha = 0.05$ or $\alpha = 0.01$.

Dataset	Metrics	p-value	$\alpha < 0.05$	$\alpha < 0.01$
CFD	F1	0.0043	√	√
	Pr	0.0086	√	√
	Recall	0.0187	√	×
DeepCrack	F1	0.0236	√	×
	Pr	0.0064	√	√
	Recall	0.0079	√	√
Crack500	F1	0.0063	√	√
	Pr	0.0152	√	×
	Recall	0.0011	√	√

https://doi.org/10.1371/journal.pone.0346889.t008

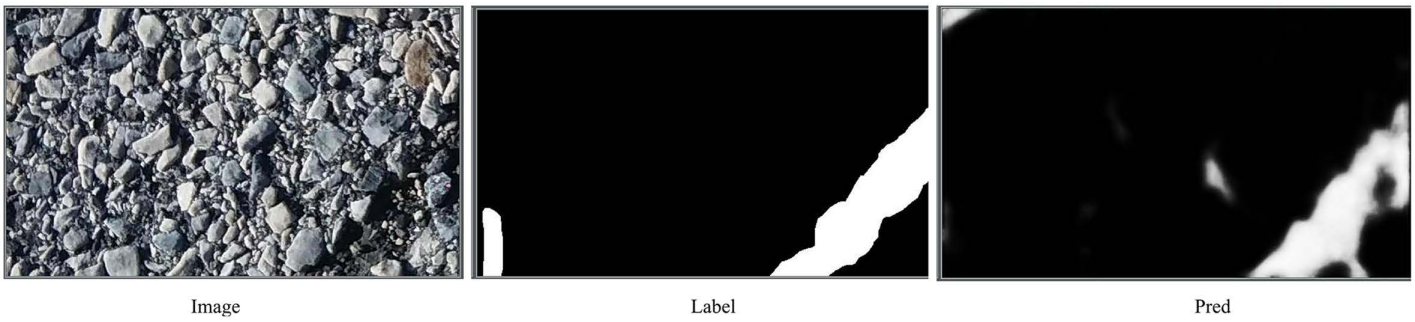

Fig 11. Representative failure case on complex textured pavement. From left to right: input image, ground-truth label, and predicted segmentation result.

https://doi.org/10.1371/journal.pone.0346889.g011

CFD dataset, the ECA version achieved a higher F1 (0.755) than the non-ECA version (0.7287), with only marginal differences in precision and recall, suggesting that ECA contributes to a more favorable balance between these metrics. For the DeepCrack dataset, LightMSCBlock with ECA achieved slightly higher precision (0.8654 vs. 0.8543) while maintaining similar recall, resulting in improved F1 (0.8214 vs. 0.8178). The effect was more pronounced on the Crack500 dataset, where both precision (0.8765 vs. 0.8664) and recall (0.7964 vs. 0.7776) increased, resulting in a higher F1 (0.7636 vs. 0.7553). These results indicate that incorporating ECA into LightMSCBlock generally improves segmentation performance across different datasets, with notable gains in more challenging scenarios, likely due to enhanced channel interaction and stronger feature representation.

As shown in Table 9 and the visualization Fig 12, LightMSCBlock consistently outperforms LightMSCBlock(w/o ECA) across all three datasets in terms of Precision, Recall, and F1, demonstrating the effectiveness of the attention mechanism.

Table 9. Comparative experiments between different attention and SAF attention on three datasets.

	CFD			DeepCrack			Crack500		
Method	**Pr**	**Recall**	**F1**	**Pr**	**Recall**	**F1**	**Pr**	**Recall**	**F1**
LightMSCBlock(w/o ECA)	0.7304	0.8103	0.7287	0.8543	0.8125	0.8178	0.8664	0.7776	0.7553
LightMSCBlock	0.7325	0.8128	0.7550	0.8654	0.8126	0.8214	0.8765	0.7964	0.7636

https://doi.org/10.1371/journal.pone.0346889.t009

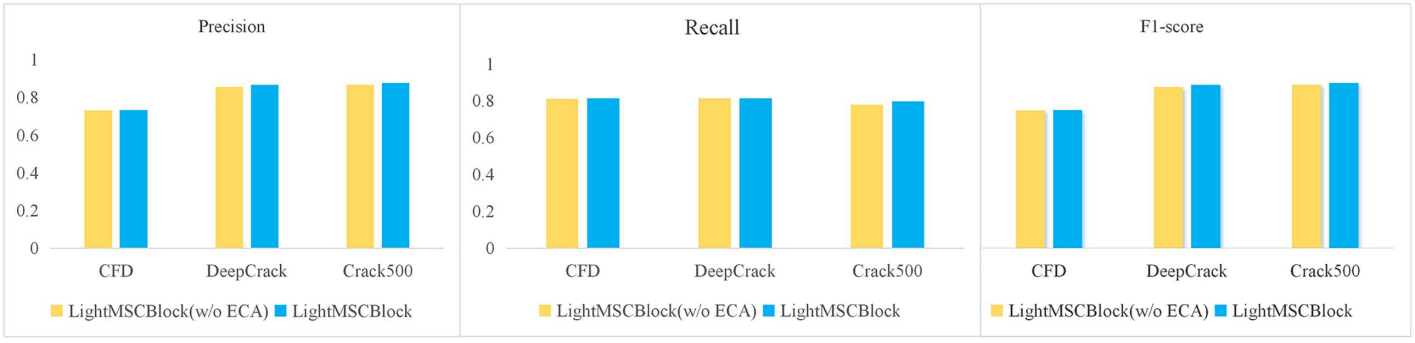

Fig 12. Performance Comparison of LightMSCBlock and LightMSCBlock(w/o ECA) across CFD, DeepCrack, and Crack500.

https://doi.org/10.1371/journal.pone.0346889.g012

4 Conclusions

With aging infrastructure and increasing concerns about structural safety, crack detection has become a critical component of structural health monitoring. Traditional manual inspection methods are time-consuming, subjective, and inefficient. At the same time, existing deep learning-based approaches, though promising, often struggle with accuracy and robustness under real-world conditions involving noise, lighting variations, and complex textures. Therefore, there is an urgent need for more accurate, robust, and practically deployable crack detection models to meet engineering demands.

To address these challenges, we propose CrackNet, a novel segmentation network tailored for concrete cracks, inspired by information theory to address uncertainty and maximize information flow. CrackNet integrates three modules: a Light multi-scale convolution enhancement block (LightMSCBlock) in the encoder for local and global feature capture, a SAF mechanism in skip connections for scale fusion and edge refinement, and a multi-scale feature fusion (MSFF) module in the decoder for enhanced integration. These designs improve feature clarity and reduce information loss. Experimental results on three public datasets—CFD, Crack500, and DeepCrack—demonstrate that CrackNet consistently outperforms baseline and state-of-the-art methods regarding accuracy, F1-score, and IoU, especially in noisy and cluttered scenarios.

Despite the notable improvements, the proposed model has several limitations. First, the relatively complex network structure increases computational cost, potentially hindering deployment on resource-constrained devices or in real-time applications. Second, the model may still struggle to detect fine cracks with low contrast or blurred edges, leading to potential false negatives. Third, this study primarily focuses on binary segmentation and does not include modules for extracting geometric crack features, such as width and length, which limits its application for quantitative damage assessment.

Future work will aim to (1) optimize the model's architecture for lightweight and real-time performance through techniques such as efficient convolutions and knowledge distillation, (2) explore multi-modal fusion with additional data sources to improve detection robustness, and (3) integrate crack measurement and tracking components to extend the model from simple detection to comprehensive structural damage assessment. Additionally, we plan to explore potential extensions to other materials (e.g., asphalt or metal) or modalities (e.g., thermal imaging), which could provide new insights and further enhance the model's generalizability and deployment in diverse real-world conditions. These improvements will contribute toward a more intelligent and complete structural health monitoring system.

Author contributions

Resources: Wubiao Zhu.

Software: Wubiao Zhu, Jiawei Yin.

Supervision: Jingying Mo, Ruibing Xie.

Validation: Mengcai Ye, Jingying Mo, Ruibing Xie.

Visualization: Jiawei Yin, Zhendi Ma.

Writing – original draft: Wubiao Zhu, Mengcai Ye, Jiawei Yin, Zhendi Ma.

Writing – review & editing: Wubiao Zhu, Mengcai Ye, Jiawei Yin.

References

1. Yamaguchi T, Hashimoto S. Fast crack detection method for large-size concrete surface images using percolation-based image processing. Machine Vision and Applications. 2009;21(5):797–809. https://doi.org/10.1007/s00138-009-0189-8
2. Belsky M, Sacks R, Brilakis I. Semantic Enrichment for Building Information Modeling. Computer-Aided Civil and Infrastructure Engineering. 2016;31(4):261–74. https://doi.org/10.1111/mice.12128

3. Cha Y, Choi W, Büyüköztürk O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. Computer-Aided Civil and Infrastructure Engineering. 2017;32(5):361–78. https://doi.org/10.1111/mice.12263
4. Dung CV, Anh LD. Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction. 2019;99:52–8. https://doi.org/10.1016/j.autcon.2018.11.028
5. Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell. 1986;8(6):679–98. https://doi.org/10.1109/tpami.1986.4767851 PMID: 21869365
6. Biswas R, Sil J. An Improved Canny Edge Detection Algorithm Based on Type-2 Fuzzy Sets. Procedia Technology. 2012;4:820–4. https://doi.org/10.1016/j.protcy.2012.05.134
7. Gaona I R, Mello-Román J C, Noguera J L V, et al. Enhanced medical images through multi-scale mathematical morphology by reconstruction[C]2023 18th Iberian Conference on Information Systems and Technologies (CISTI). IEEE, 2023:1–5.
8. Yu X, Wang Z, Wang Y. Edge detection of agricultural products based on morphologically improved canny algorithm. Mathematical Problems in Engineering. 2021;2021(1):6664970.
9. Xu Z, Ji X, Wang M, Sun X. Edge detection algorithm of medical image based on Canny operator. J Phys: Conf Ser. 2021;1955(1):012080. https://doi.org/10.1088/1742-6596/1955/1/012080
10. Jing P, Yu H, Hua Z, Xie S, Song C. Road Crack Detection Using Deep Neural Network Based on Attention Mechanism and Residual Structure. IEEE Access. 2023;11:919–29. https://doi.org/10.1109/access.2022.3233072
11. Xing H, Zhu L, Chen B, Liu C, Niu J, Li X, et al. A comparative study of threshold selection methods for change detection from very high-resolution remote sensing images. Earth Sci Inform. 2022;15(1):369–81. https://doi.org/10.1007/s12145-021-00734-y
12. Nguyen SD, Tran TS, Tran VP, Lee HJ, Piran MdJ, Le VP. Deep Learning-Based Crack Detection: A Survey. Int J Pavement Res Technol. 2022;16(4):943–67. https://doi.org/10.1007/s42947-022-00172-z
13. Roberti de Siqueira F, Robson Schwartz W, Pedrini H. Multi-scale gray level co-occurrence matrices for texture description. Neurocomputing. 2013;120:336–45. https://doi.org/10.1016/j.neucom.2012.09.042
14. Kobayashi T, Hidaka A, Kurita T. Selection of histograms of oriented gradients features for pedestrian detection[C]International conference on neural information processing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 598–607.
15. Ranjbar S, Nejad FM, Zakeri H. An image-based system for pavement crack evaluation using transfer learning and wavelet transform. Int J Pavement Res Technol. 2020;14(4):437–49. https://doi.org/10.1007/s42947-020-0098-9
16. Fujita Y, Hamamoto Y. A robust automatic crack detection method from noisy concrete surfaces. Machine Vision and Applications. 2010;22(2):245–54. https://doi.org/10.1007/s00138-009-0244-5
17. Zheng R, Guo Q, Gao C, et al. A hybrid contrast limited adaptive histogram equalization (clahe) for parathyroid ultrasonic image enhancement[C] 2019 Chinese Control Conference (CCC). IEEE. 2019: 3577–3582.
18. Goo JM, Milidonis X, Artusi A, Boehm J, Ciliberto C. Hybrid-Segmentor: Hybrid approach for automated fine-grained crack segmentation in civil infrastructure. Automation in Construction. 2025;170:105960. https://doi.org/10.1016/j.autcon.2024.105960
19. Tang W, Wu Z, Wang W, Pan Y, Gan W. VM-UNet++ research on crack image segmentation based on improved VM-UNet. Sci Rep. 2025;15(1):8938. https://doi.org/10.1038/s41598-025-92994-7 PMID: 40089495
20. Liu H, Jia C, Shi F, et al. SCSegamba: lightweight structure-aware vision mamba for crack segmentation in structures. Proceedings of the Computer Vision and Pattern Recognition Conference. 2025. p. 29406–16. 20.
21. Kong Q, Wu Z, Song Y. Online detection of external thread surface defects based on an improved template matching algorithm. Measurement. 2022;195:111087. https://doi.org/10.1016/j.measurement.2022.111087
22. Zhou Y, Huang Y, Chen Q, Yang D. Graph-based change detection of pavement cracks. Automation in Construction. 2025;174:106110. https://doi.org/10.1016/j.autcon.2025.106110
23. Ali L, Alnajjar F, Jassmi HA, Gocho M, Khan W, Serhani MA. Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. Sensors (Basel). 2021;21(5):1688. https://doi.org/10.3390/s21051688 PMID: 33804490
24. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Cham: Springer international publishing, 2015. p. 234–41.
25. Zhang Q, Chen S, Wu Y, Ji Z, Yan F, Huang S, et al. Improved U-net network asphalt pavement crack detection method. PLoS One. 2024;19(5):e0300679. https://doi.org/10.1371/journal.pone.0300679 PMID: 38820536
26. Xu X, Zhao M, Shi P, Ren R, He X, Wei X, et al. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. Sensors (Basel). 2022;22(3):1215. https://doi.org/10.3390/s22031215 PMID: 35161961
27. Attard L, Debono C J, Valentino G, et al. Automatic crack detection using mask R-CNN[C]2019 11th international symposium on image and signal processing and analysis (ISPA). IEEE, 2019. p. 152–7.
28. Sun X, Xie Y, Jiang L, Cao Y, Liu B. DMA-Net: DeepLab With Multi-Scale Attention for Pavement Crack Segmentation. IEEE Trans Intell Transport Syst. 2022;23(10):18392–403. https://doi.org/10.1109/tits.2022.3158670
29. Zhang J, Bao T. An Improved ResNet-Based Algorithm for Crack Detection of Concrete Dams Using Dynamic Knowledge Distillation. Water. 2023;15(15):2839. https://doi.org/10.3390/w15152839

30. Akgül İ. Mobile-DenseNet: Detection of building concrete surface cracks using a new fusion technique based on deep learning. Heliyon. 2023;9(10):e21097. https://doi.org/10.1016/j.heliyon.2023.e21097 PMID: 37886768
31. Rani R, Bharany S, Elkamchouchi DH, Ur Rehman A, Singh R, Hussen S. VGG-EffAttnNet: Hybrid Deep Learning Model for Automated Chili Plant Disease Classification Using VGG16 and EfficientNetB0 With Attention Mechanism. Food Sci Nutr. 2025;13(7):e70653. https://doi.org/10.1002/fsn3.70653 PMID: 40708782
32. Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]International conference on machine learning. PMLR; 2019. p. 6105–14.
33. Lu W, Qian M, et al. Crack _ PSTU: Crack detection based on the U-Net framework combined with Swin Transformer. Structures. Elsevier; 2024. p. 106241.
34. Zhang P, Dai X, Yang J. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 2998–3008.
35. Zhang J, Zeng Z, Sharma PK, Alfarraj O, Tolba A, Wang J. A dual encoder crack segmentation network with Haar wavelet-based high–low frequency attention. Expert Systems with Applications. 2024;256:124950. https://doi.org/10.1016/j.eswa.2024.124950
36. Wang J, Yao H, Hu J, Ma Y, Wang J. Dual-encoder network for pavement concrete crack segmentation with multi-stage supervision. Automation in Construction. 2025;169:105884. https://doi.org/10.1016/j.autcon.2024.105884
37. Pan Z, Lau SLH, Yang X, Guo N, Wang X. Automatic pavement crack segmentation using a generative adversarial network (GAN)-based convolutional neural network. Results in Engineering. 2023;19:101267. https://doi.org/10.1016/j.rineng.2023.101267
38. He K, Gkioxari G, Dollár P, et al. Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision, 2017. p. 2961–9.
39. Wang Q, Wu B, Zhu P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. p. 11534–42.
40. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 7132–7141. 7132-7141.
41. Shi Y, Cui L, Qi Z, Meng F, Chen Z. Automatic Road Crack Detection Using Random Structured Forests. IEEE Trans Intell Transport Syst. 2016;17(12):3434–45. https://doi.org/10.1109/tits.2016.2552248
42. Yang F, Zhang L, Yu S, Prokhorov D, Mei X, Ling H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. IEEE Trans Intell Transport Syst. 2020;21(4):1525–35. https://doi.org/10.1109/tits.2019.2910595
43. Liu Y, Yao J, Lu X, Xie R, Li L. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing. 2019;338:139–53. https://doi.org/10.1016/j.neucom.2019.01.036