

Article

A Novel Lightweight Framework for Real-Time Pavement Crack Segmentation Based on Knowledge Distillation

Ning Xu ^{1,*}, Jinghui Qiao ^{1,*} and Yunze Tang ²¹ School of Mechanical Engineering, Shenyang University of Technology, Shenyang 110870, China² School of Mechanical Engineering, Liaoning Petrochemical University, Fushun 113001, China; yz_t1995@163.com

* Correspondence: wlxxcd@163.com (N.X.); qiaojh2002@163.com (J.Q.)

Abstract

Accurate pavement crack segmentation is essential for structural health monitoring, yet existing methods often face a trade-off between segmentation accuracy and computational efficiency. To address this issue, a novel teacher–student framework, termed RTCS-T and RTCS-S, is proposed. The teacher network RTCS-T is constructed based on the Swin Transformer to capture long-range dependencies and multi-scale contextual information. To further enhance crack representation, a strip refinement module is introduced to model directional structural features, while a cascaded atrous spatial pyramid pooling module is employed to improve multi-scale feature aggregation. Based on the teacher network, a lightweight student model RTCS-S is developed by using depthwise separable convolutions to achieve efficient inference. In addition, a foreground-aware and boundary-aware knowledge distillation strategy is introduced to guide the transfer of structural and contextual information from the teacher to the student. Experiments on the Crack500, DeepCrack, and CFD datasets demonstrated competitive performance against representative segmentation models. On CFD, RTCS-S achieved an F1 Score of 0.7514 and an mIoU of 0.7962. Notably, RTCS-S required only 1.82 M parameters and 1.13 GFLOPs and achieved a model inference speed of 680 FPS on an RTX 4090 GPU. When deployed on an RDK X5 edge-computing platform, the complete pipeline achieved an end-to-end throughput of 34 FPS, with an average latency of approximately 29.4 ms and peak memory consumption of 1.8 GB. These results demonstrate that the proposed framework provides an efficient solution for automated pavement crack detection and shows strong potential for practical road inspection applications.

Keywords: pavement crack segmentation; lightweight framework; knowledge distillation; real-time detection; deep learning

Received: 11 July 2026

Revised: 4 August 2026

Accepted: 5 August 2026

Published: 6 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Pavement cracks are among the most common and representative forms of defects in road infrastructure. If not detected and repaired promptly, they may gradually propagate under repeated traffic loading and environmental effects, eventually leading to structural degradation, reduced service life, and increased maintenance costs. Therefore, accurate and efficient pavement crack segmentation is essential for structural health monitoring of transportation infrastructure [1].

Manual crack detection is labor-intensive, time-consuming, and costly, while its accuracy is highly dependent on the experience and attentiveness of inspectors. To overcome these limitations, researchers are increasingly interested in image processing technology. Early representative studies explored a variety of traditional image processing strategies for crack segmentation. Oliveira and Correia [2] developed a threshold-based crack segmentation framework that combined image preprocessing with dynamic thresholding to identify dark crack pixels in pavement images, thereby improving automatic crack detection under varying imaging conditions. Abdel-Qader et al. [3] investigated edge-based crack identification by comparing several edge detection operators, including Sobel [4] and Canny [5]. Yamaguchi et al. [6] proposed a method for crack segmentation on real concrete surfaces, which improved crack extraction by reducing background noise interference. In addition, Serra [7] systematically established the theoretical foundation of mathematical morphology, which later became an important tool for image filtering, shape analysis, and crack enhancement in surface defect detection tasks. Although these methods are relatively simple and easy to implement, their performance is often sensitive to illumination variation, stains, shadows, noise, and complex pavement textures [8]. In practical scenarios, crack patterns usually exhibit thin, elongated, tortuous, and sometimes discontinuous structures, while the contrast between crack regions and background may be weak. Under these conditions, conventional methods often struggle to achieve accurate crack extraction.

In recent years, deep learning-based crack segmentation has attracted increasing attention due to its ability to provide more robust segmentation results in noisy environmental conditions. Long et al. [9] proposed the fully convolutional network (FCN), which established an end-to-end framework for dense pixel-wise prediction and became a milestone in modern semantic segmentation. Ronneberger et al. [10] introduced U-Net, whose encoder–decoder architecture with skip connections significantly improved localization accuracy. Badrinarayanan et al. [11] proposed SegNet, which used pooling indices for decoder upsampling, thereby improving encoder–decoder segmentation efficiency. Zhao et al. [12] further introduced PSPNet to strengthen global contextual aggregation through pyramid pooling. In parallel, Chen et al. [13] developed the DeepLab series, which used atrous convolution and atrous spatial pyramid pooling (ASPP) to enlarge receptive fields and model multi-scale context, while DeepLabv3+ [14] further improved boundary recovery through an encoder–decoder design. More recently, Qin et al. [15] further introduced CrackMambaNet, a U-shaped network with a dual-branch encoder and adaptive feature fusion, in which visual state space blocks were employed to capture global contextual dependencies while a CNN branch was used to preserve fine-grained local details. Sun et al. [16] proposed CCBG-Net, which enhances crack representation through bidirectional cross-layer context fusion and a boundary-guided interaction module, thereby improving the segmentation of thin cracks and low-contrast boundaries.

The emergence of Transformer architectures has further advanced crack segmentation research by providing a more effective mechanism for modeling long-range dependencies and global contextual interactions. Vaswani et al. [17] introduced the Transformer architecture, establishing an attention-based framework for sequence modeling. Building on this idea, Dosovitskiy et al. [18] introduced the Vision Transformer (ViT), demonstrating that images can be represented as sequences of patches for visual representation learning. To better adapt Transformers to dense prediction tasks, Liu et al. [19] developed the Swin Transformer, which employs hierarchical feature representation and shifted-window self-attention to achieve a more favorable balance between contextual modeling capability and computational complexity. Inspired by these advances, Transformer-based designs have gradually been introduced into pavement crack detection and segmentation. For example, Guo et al. [20] proposed a Transformer-based network for pavement crack

detection, showing improved capability in capturing complex crack patterns under challenging background conditions. In addition, Wang et al. [21] developed SwinCrack, which integrates convolutional operations with the Swin Transformer to further enhance crack continuity representation and suppress background interference. Despite their promising performance, however, Transformer-based crack segmentation models usually involve relatively high computational complexity and large model size, which may limit their practical deployment on resource-constrained devices.

In parallel with the pursuit of higher segmentation accuracy, lightweight crack analysis has also attracted increasing attention. In practical applications, crack inspection systems are often expected to operate on mobile devices, embedded platforms, or edge hardware with limited computational resources. To meet this demand, a series of lightweight visual backbones, such as MobileNet [22], ShuffleNet [23], and EfficientNet [24], have been introduced into dense prediction tasks. Their core idea is to reduce model complexity through efficient convolutional design, channel optimization, or depthwise separable operations. For example, Liang et al. [25] proposed PCSNet, which combines a lightweight CNN backbone with multi-scale context modeling; Dong et al. [26] developed LPCD-Net to improve crack representation through progressive fusion and attention mechanisms; and Zim et al. [27] introduced EfficientCrackNet, which integrates depthwise separable convolutions with MobileViT blocks for efficient crack segmentation. Although such lightweight architectures can significantly improve inference speed and reduce memory consumption, their representation capability is often insufficient for handling subtle crack boundaries, discontinuous crack branches, and low-contrast crack regions. As a result, lightweight crack segmentation still faces a challenging trade-off between computational efficiency and segmentation accuracy.

Beyond conventional CNN backbones, lightweight Vision Transformers such as MobileViT [28] seek to combine global receptive fields with mobile-friendly computation. Foundation segmentation models, exemplified by the Segment Anything Model (SAM) [29], offer strong promptable segmentation capability, and lightweight derivatives such as MobileSAM [30] further demonstrate how knowledge distillation can substantially reduce encoder cost for mobile use. Nevertheless, these general-purpose models are not tailored to the extremely sparse foreground, thin topology, low contrast, and pixel-level boundary fidelity required by pavement-crack segmentation; their computational demand and prompt-dependent workflow also limit straightforward on-device use.

Knowledge distillation (KD) provides a complementary route for transferring the representational capacity of a large model to a compact model [31]. For semantic segmentation, existing KD approaches have progressed from pixel-wise supervision to the transfer of pairwise and holistic structural relations [32]. In crack analysis, Fan et al. [1] presented a lightweight concrete bridge-crack image-recognition method based on KD. However, that study addresses image-level recognition rather than dense road-pavement crack segmentation and therefore does not directly address the preservation of thin crack boundaries, discontinuous branches, or spatial crack topology. More recently, Zhang et al. [33] proposed a real-time pavement crack measurement framework for edge deployment, combining a lightweight distillation network with surface-feature measurement. Their IHD module integrated feature-based and relation-based distillation using crack instance information. Crack width and length were then estimated through distance mapping and edge-coordinate extraction using a mobile robot equipped with an edge-computing unit. More broadly, conventional KD losses can be dominated by background pixels in crack images, making indiscriminate feature or logit transfer insufficient for the highly imbalanced, boundary-sensitive segmentation setting considered here.

Recent advances in edge artificial intelligence (edge AI) are shifting deep-learning inference from remote cloud servers to embedded devices located close to the data source

[34]. For pavement inspection, local processing can reduce data-transmission demands and communication latency, thereby supporting timely analysis of continuously acquired road images. However, practical deployment remains constrained by the limited computing capability, memory capacity, and energy budget of edge hardware. Recent studies therefore emphasize the joint use of lightweight architecture design, model compression techniques such as knowledge distillation, hardware acceleration, and device-level evaluation [35]. In this context, parameter count, FLOPs, and inference speed measured on desktop GPUs alone are insufficient to characterize the performance of a complete embedded crack-segmentation pipeline, particularly in terms of latency, memory consumption, and end-to-end throughput. This gap motivates the development of lightweight crack-segmentation models that preserve crack structures and boundaries while being validated through end-to-end inference on resource-constrained edge platforms.

Consequently, although prior work has improved either contextual representation, model compactness, or knowledge transfer, existing methods do not sufficiently address these requirements jointly: modeling diverse crack morphologies, preserving fine boundaries and structural continuity, and transferring task-relevant knowledge to an efficient student suitable for edge deployment.

To overcome the above limitations, a teacher–student crack-segmentation framework, RTCS-T and RTCS-S, is proposed in this study. RTCS-T is constructed as a Swin Transformer-based teacher network to learn discriminative multi-scale crack representations. To characterize the directional and elongated morphology of pavement cracks, a strip refinement module (SRM) is introduced to enhance directional feature modeling and crack continuity. A cascaded atrous spatial pyramid pooling (CASPP) module is further used to strengthen multi-scale contextual aggregation and inter-scale information interaction. Based on the teacher network, the lightweight student network RTCS-S employs depthwise separable convolutions and a hierarchical encoder–decoder architecture to reduce computational complexity. Finally, a foreground-aware and boundary-aware KD strategy is designed to prioritize crack regions, boundary details, and structural continuity during teacher-to-student transfer. Thus, the proposed framework is designed to jointly improve morphology-aware representation, boundary preservation, and efficient knowledge transfer for practical pavement-crack segmentation. The main contributions of this paper are as follows:

1. A Swin Transformer-based teacher network, RTCS-T, is proposed for pavement crack segmentation, in which SRM and CASPP are incorporated to enhance crack-specific feature representation through directional refinement and multi-scale contextual modeling.
2. A lightweight student network, RTCS-S, is developed based on depthwise separable convolutions and a hierarchical encoder–decoder architecture, enabling efficient crack segmentation with low computational cost.
3. A foreground-aware and boundary-aware knowledge distillation framework is proposed to transfer discriminative structural knowledge from the teacher network to the student network, with particular emphasis on crack regions, boundary details, and structural continuity.
4. Extensive experiments on three public crack segmentation datasets demonstrate that the proposed framework achieves a favorable balance between segmentation accuracy and computational efficiency, indicating its potential for practical structural health monitoring applications.

The remainder of this paper is organized as follows. Section 2 presents the proposed framework, including the network architectures and the knowledge distillation strategy. Section 3 describes the datasets, experimental settings, and evaluation metrics. Section 4 reports and discusses the experimental results, followed by an analysis of the proposed method's limitations. Section 5 concludes the paper and suggests possible directions for future work.

2. Methodology

2.1. Overall Structure of Teacher Network

In this paper, a novel crack segmentation framework, RTCS-T, is designed as the teacher network for knowledge distillation. As illustrated in Figure 1, the proposed framework follows an encoder–decoder architecture. Specifically, the encoder is constructed based on the Swin Transformer, which hierarchically extracts multi-scale feature representations across four stages. To enhance feature representation capability, the SRMs are introduced to refine contextual information within each stage. Meanwhile, a CASPP module is employed to further aggregate multi-scale contextual features through atrous convolutions with different receptive fields. During the decoding process, the encoded feature maps are progressively fused with the corresponding high-resolution features from the encoder through skip connections to recover spatial details and generate the final crack segmentation mask. The important components of the proposed network are described in detail in the following subsections.

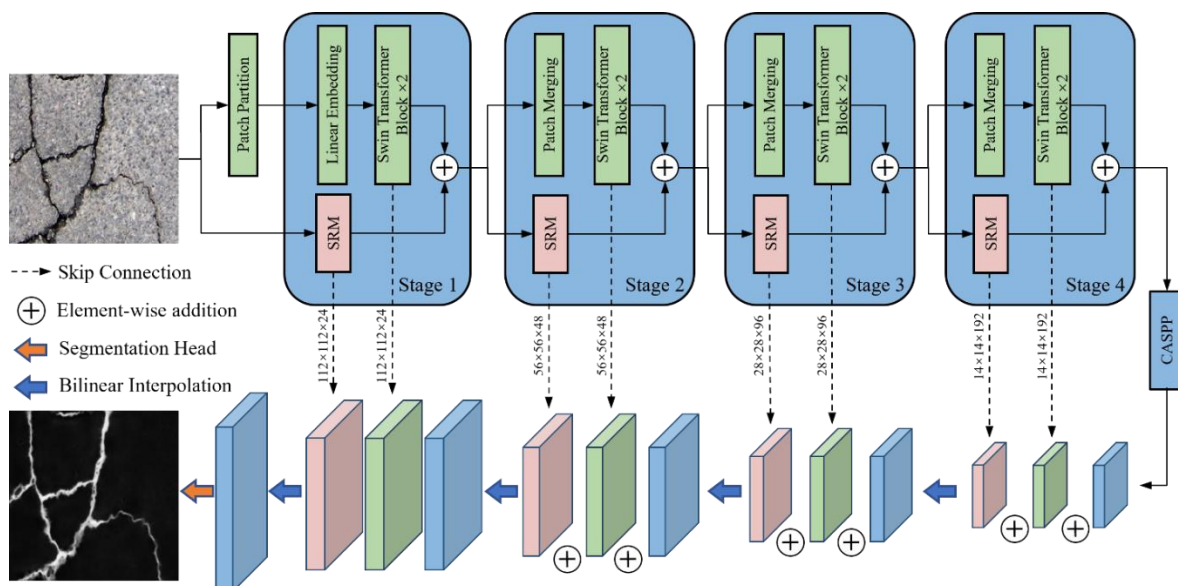


Figure 1. The overall structure of the teacher model RTCS-T.

2.1.1. Swin Transformer Block

To model long-range dependencies in crack images, RTCS-T employs a customized lightweight hierarchical encoder derived from the Swin Transformer (tiny version), as illustrated in Figure 1. The raw crack image with the dimension of $(224, 224, 3)$ is partitioned into non-overlapping patches in the patch partition layer. Each patch with a size of 4×4 pixels are flattened along the channel dimensions to form a feature vector. Consequently, each patch contains $4 \times 4 \times 3 = 48$ dimensions. Therefore, the input image is transformed from $(224, 224, 3)$ to $(56, 56, 48)$. Subsequently, a linear embedding layer is employed to project the flattened patch features into a feature representation. In the original Swin Transformer, this process yields a feature map of size $(56, 56, 96)$. However, since pavement cracks usually present thin and elongated structures, such a representation may not be sufficiently effective in preserving fine spatial details at the early stage. To address this issue, the feature map is reformulated as $(112, 112, 24)$ in the proposed framework. Notably, although the total number of feature elements remains unchanged, this transformation reallocates part of the representation from the channel dimension to the spatial dimension, thereby generating a higher-resolution feature map for subsequent processing. As a result, more detailed structural information of crack regions can be

preserved, which is beneficial for fine-grained crack segmentation while maintaining a reasonable computational burden. The resulting feature map is then processed by the Swin Transformer block for further feature extraction. In each encoder stage, the Swin Transformer block performs attention-based feature extraction while keeping the spatial resolution and channel dimension unchanged. Therefore, the output of the first stage remains (112, 112, 24).

From the second stage, the patch merging operation halves the spatial resolution of the feature map from the previous stage while doubling the channel dimension. Thus, the output feature maps of stages 2, 3, and 4 are (56, 56, 48), (28, 28, 96), and (14, 14, 192), respectively.

The structure of the Swin Transformer block can be seen in Figure 2, including the multi-head self-attention (MSA), the multiple layer perceptron (MLP), and the layer normalization (LN). It should be noted that the standard Transformer is limited in dense prediction and high-resolution image tasks due to the expensive computational cost of global self-attention in MSA. To address this issue, the Swin Transformer adopts the window-based multi-head self-attention (W-MSA) mechanism, in which self-attention is computed within non-overlapping local windows rather than over the entire feature map. In this manner, the computational burden is significantly reduced. Nevertheless, W-MSA still suffers from limited information interaction across different windows, since each window is processed independently. As a result, the dependency modeling across window boundaries may be insufficient. Hence, shifted window-based multi-head self-attention (SW-MSA) is further proposed. By introducing a shifted-window partitioning scheme across consecutive blocks, SW-MSA establishes connections among neighboring windows and thereby enhances cross-window information exchange without substantially increasing computational complexity.

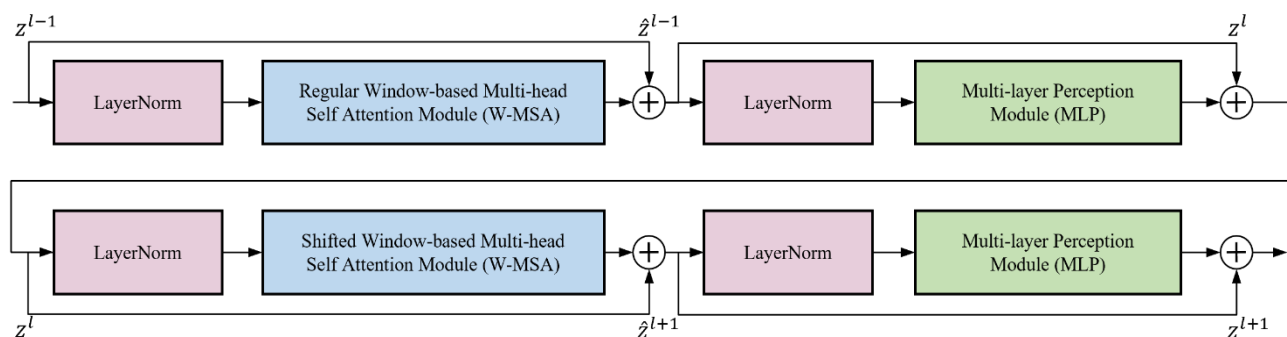


Figure 2. The structure of the Swin Transformer block.

As illustrated in Figure 3, the left side shows the regular non-overlapping window partitioning used in W-MSA, where self-attention is computed independently within each local window. In contrast, the right side represents the shifted window partitioning scheme used in SW-MSA, in which the window configuration is shifted with respect to that in W-MSA. Through this shifted windowing scheme, cross-window connections can be established among neighboring non-overlapping windows, thereby enhancing information exchange across window boundaries. The computational procedure of each Swin Transformer block is described in Equations (1)–(4).

$$\hat{z}^l = \text{W-MSA}\left(\text{LN}\left(z^{l-1}\right)\right) + z^{l-1} \quad (1)$$

$$z^l = \text{MLP}\left(\text{LN}\left(\hat{z}^l\right)\right) + \hat{z}^l \quad (2)$$

$$\hat{z}^{l+1} = \text{SW-MSA}\left(\text{LN}\left(z^l\right)\right) + z^l \quad (3)$$

$$z^{l+1} = \text{MLP}\left(\text{LN}\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1} \quad (4)$$

where $\hat{z}^l$ denotes the output feature maps of the W-MSA module. $\hat{z}^{l+1}$ represents the output feature maps of the SW-MSA module. z^l and z^{l+1} are the output feature maps of MLP. The number of Swin Transformer blocks in Stages 1–4 is 2, 2, 6, and 2, respectively.

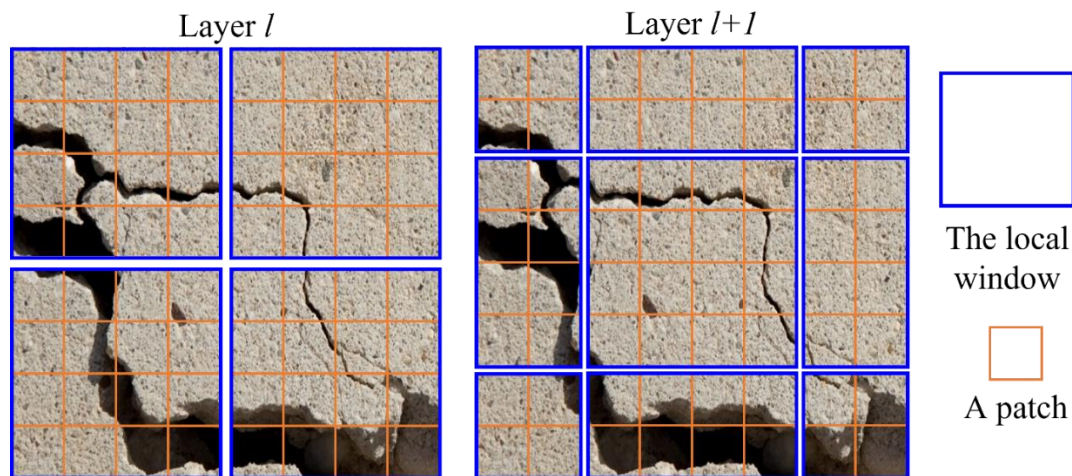


Figure 3. The illustration of shifted window operation.

2.1.2. Strip Refinement Module

Most existing crack feature extraction methods mainly rely on convolutional operators, multi-scale context aggregation, and Transformer-based architectures to improve the representation of crack regions. Nevertheless, these methods are generally designed for general visual structures and lack explicit modeling of the distinctive morphology of pavement cracks. Pavement cracks typically exhibit slender, elongated, and direction-sensitive structures. Since standard convolution kernels aggregate information within isotropic square neighborhoods, they are less effective for describing such anisotropic and topology-sensitive structures. As a result, thin crack branches and disconnected crack segments are often insufficiently represented, which limits the structural continuity of crack segmentation. Inspired by strip pooling [36], as shown in Figure 4, the SRM is proposed to enhance morphology-aware crack representation. The key idea of SRM is to introduce strip-shaped receptive fields to refine feature responses along specific spatial directions, thereby enhancing directional continuity and improving the representation of elongated crack structures.

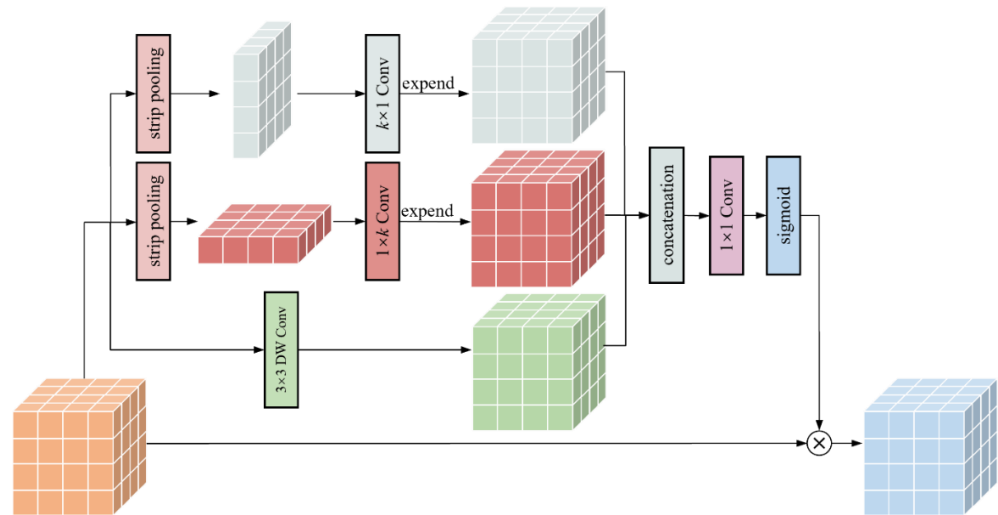


Figure 4. The structure of the SRM.

Specifically, the proposed SRM adopts a multi-branch architecture consisting of two directional strip branches and one local detail branch. Unlike the method in strip pooling, which focuses on directional feature modeling, SRM introduces an additional local branch to mitigate the potential loss of fine-grained details caused by strip pooling and to better preserve crack structures. Given an input feature map $F \in \mathbb{R}^{H \times W \times C}$, the horizontal branch applies a strip convolution with a kernel size of $1 \times k$ to extract horizontally distributed contextual feature maps, while the vertical branch employs a strip convolution with a kernel size of $k \times 1$ to capture directional feature maps along the vertical direction. In addition, a local branch with a 3×3 depthwise convolution is introduced to preserve local texture information. The outputs of the three branches can be formulated as:

$$F_h = \text{Conv}_{1 \times k}(F) \quad (5)$$

$$F_v = \text{Conv}_{k \times 1}(F) \quad (6)$$

$$F_l = \text{Conv}_{3 \times 3}(F) \quad (7)$$

where F_h , F_v , and F_l denote the horizontal, vertical, and local feature maps, respectively. These feature maps are concatenated along the channel dimension and then compressed by a 1×1 convolution to produce the fused feature, which can be expressed as:

$$F_s = \text{Conv}_{1 \times 1}([F_h; F_v; F_l]) \quad (8)$$

where $[\cdot; \cdot; \cdot]$ denotes channel-wise concatenation. Finally, the fused feature is further refined by a spatial attention mechanism. The final output F_{out} is computed as:

$$F_{\text{out}} = \text{Scale}(F, \sigma(F_s)) \quad (9)$$

where σ represents a sigmoid function, and $\text{Scale}(\cdot, \cdot)$ is element-wise multiplication. It should be noted that SRM is designed for directional feature extraction and refinement, without changing the spatial resolution or channel dimension of the input feature map. To ensure consistency with the feature size required by each stage of the Swin Transformer encoder, a 3×3 depthwise convolution is applied in advance to adjust the spatial resolution and channel dimension of the input feature map. In addition, the strip convolution kernels in the two directional branches are set to 1×5 and 5×1 , a stride of 1, and zero-padding sizes of (0, 2) and (2, 0), respectively. All convolutions in SRM are followed

by batch normalization and ReLU activation. The channel number is set to 24, 48, 96, and 192 in the four encoder stages, respectively.

2.1.3. Cascaded Atrous Spatial Pyramid Pooling

In semantic segmentation tasks, the effective extraction of multi-scale contextual information is crucial for accurate dense prediction. Since target structures may vary significantly in scale and morphology, relying on a single receptive field often makes it difficult to capture both local details and global semantic dependencies simultaneously. To address this issue, atrous spatial pyramid pooling (ASPP) was introduced as a representative multi-scale context modeling strategy. By utilizing parallel atrous convolutions with different dilation rates, ASPP enlarges the receptive field without reducing the spatial resolution, thereby enabling the network to aggregate contextual information under multiple scales. However, although ASPP is capable of capturing multi-scale context, its independent branches and sparse sampling at large dilation rates make it less effective at preserving the fine continuity and subtle details of thin, tortuous, and discontinuous crack patterns.

To overcome the above limitations, the CASPP module is introduced, as depicted in Figure 5. Unlike conventional ASPP, which employs multiple independent atrous branches in parallel, the proposed CASPP adopts a cascaded context modeling strategy. Specifically, several atrous convolutions with progressively increasing dilation rates are connected sequentially, allowing contextual features captured at smaller receptive fields to be further propagated across branches with larger receptive fields. In addition, residual summation is incorporated between adjacent atrous stages, so that the original input context can be continuously injected into deeper atrous branches.

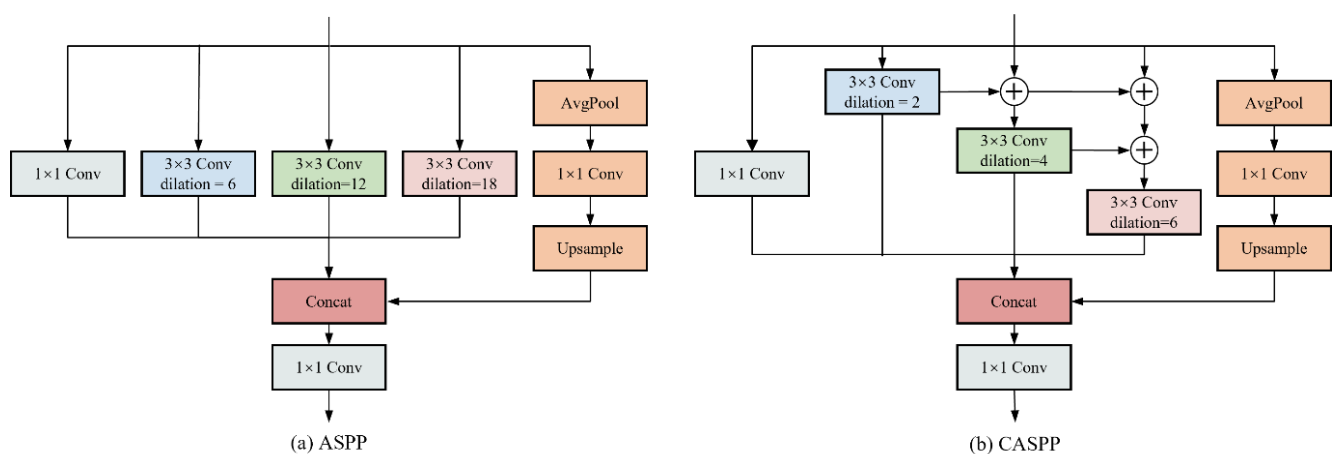


Figure 5. The overview of the conventional ASPP and the proposed CASPP.

Specifically, CASPP consists of four components: a 1×1 convolution branch for local feature preservation, a cascaded atrous convolution branch with dilation rates of 2, 4, and 6, a global average pooling branch for contextual encoding, and a final fusion layer for feature integration. The dilation rates of 2, 4, and 6 were selected according to the spatial resolution of the deepest encoder feature map and the complementary roles of the CASPP branches. CASPP operates on a feature map with a spatial resolution of 14×14 . For a 3×3 convolution, dilation rates of 2, 4, and 6 correspond to individual effective sampling spans of 5×5 , 9×9 , and 13×13 , respectively. This progressively increasing configuration provides multi-scale contextual information, while keeping the largest sampling span compatible with the spatial extent of the input feature map. Using substantially larger dilation rates at this resolution would cause an increasing proportion of the sampling positions to fall outside the valid feature region, thereby reducing effective contextual utilization. In addition,

because the 1×1 convolution and global average pooling branches already encode local and global information, respectively, the three atrous stages are mainly used to provide complementary intermediate-scale representations. Therefore, the rates of 2, 4, and 6 provide a balanced transition from local structural details to broader contextual dependencies. All convolutions in CASPP are followed by batch normalization and ReLU activation. The module receives a 14×14 feature map with 192 channels and produces an output with the same spatial resolution and channel dimension. The 1×1 convolution branch and each atrous convolution produce 192-channel features. The three atrous convolutions use a stride of 1 and padding sizes of 2, 4, and 6, respectively, thereby preserving the spatial resolution. The global feature is also projected to 192 channels. Finally, the five 192-channel features are concatenated into 960 channels and projected back to 192 channels through the final 1×1 convolution. Given an input feature map $F \in \mathbb{R}^{H \times W \times C}$, the output of the 1×1 convolution branch is defined as follows:

$$F_{1 \times 1} = \text{Conv}_{1 \times 1}(F) \quad (10)$$

For the cascaded atrous convolution branch, three 3×3 atrous convolutions with dilation rates of 2, 4, and 6 are sequentially employed. The first feature map is obtained as:

$$F_1 = \text{Conv}_{d=2}(F) \quad (11)$$

where $\text{Conv}_{d=2}(\cdot)$ denotes a 3×3 atrous convolution with dilation rate 2. Then, the first feature map F_1 is added to the original input feature F through residual summation, and the resulting feature map is fed into the second atrous convolution to obtain F_2 . The process can be expressed as follows:

$$\hat{F}_1 = F_1 + F \quad (12)$$

$$F_2 = \text{Conv}_{d=4}(F) \quad (13)$$

where $\text{Conv}_{d=4}(\cdot)$ denotes a 3×3 atrous convolution with dilation rate 4. Similarly, the detailed formulation of the third feature map is described as follows:

$$\hat{F}_2 = F_2 + F \quad (14)$$

$$F_3 = \text{Conv}_{d=6}(F) \quad (15)$$

where $\text{Conv}_{d=6}(\cdot)$ denotes a 3×3 atrous convolution with dilation rate 6. In this way, the cascaded atrous convolution branch progressively enlarges the receptive field while repeatedly injecting the original input feature map into each stage. This helps preserve structural details and facilitates more stable multi-scale feature propagation.

The output of the global average pooling branch is formulated as follows:

$$F_{\text{AvgPool}} = \text{Up}(\text{Conv}_{1 \times 1}(\text{Avg}(F))) \quad (16)$$

where $\text{Avg}(\cdot)$ denotes global average pooling and $\text{Up}(\cdot)$ denotes bilinear upsampling. Finally, the feature maps of all branches are concatenated and fused by a 1×1 convolution, which can be expressed as:

$$F_{\text{CASPP}} = \text{Conv}_{1 \times 1}([F_{1 \times 1}; F_1; F_2; F_3; F_{\text{AvgPool}}]) \quad (17)$$

where $F_{\text{CASPP}} \in \mathbb{R}^{H \times W \times C}$.

The proposed CASPP module not only preserves the multi-scale contextual modeling capability of ASPP, but also enhances inter-scale information interaction through

cascaded atrous convolutions, thereby providing more detailed representations for fine-grained, discontinuous and complex crack structures.

2.2. Overall Structure of Student Network

In knowledge distillation, the structural relationship between the teacher network and the student network has an important influence on the learning efficiency and final performance of the student network. Therefore, when designing the teacher network and student network, it is necessary to ensure that they have a certain structural similarity to facilitate effective knowledge transfer. Meanwhile, an appropriate structural difference is also necessary to preserve the diversity of knowledge distillation. For this reason, a Swin Transformer-based encoder–decoder network, RTCS-T, is adopted as the teacher model, whereas a lightweight convolutional encoder–decoder network, RTCS-S, is designed as the student model.

The overall framework of RTCS-S is shown in Figure 6. RTCS-S adopts a lightweight hierarchical encoder–decoder architecture for crack segmentation. Specifically, the encoder consists of four stages, each formed by a feature extraction module composed of a depthwise separable convolution (3×3 depthwise + 1×1 pointwise) followed by an SRM. Through progressive downsampling, the feature maps are encoded into four levels with resolutions and channel dimensions of (112, 112, 24), (56, 56, 48), (28, 28, 96), and (14, 14, 192), respectively. Then, a CASPP module is employed to enhance multi-scale contextual representation on the deepest feature map. Subsequently, the decoder progressively restores the spatial resolution in a stage-wise manner by fusing the high-level semantic features with the corresponding encoder features via skip connections. Finally, the refined feature map is fed into the segmentation head to produce the crack prediction mask.

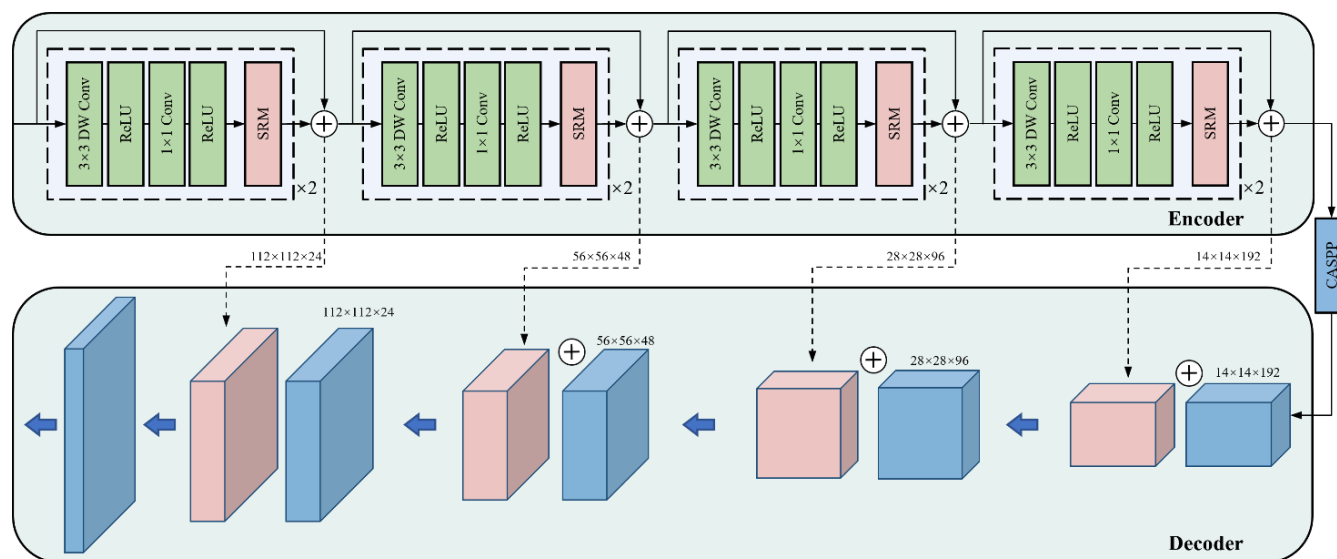


Figure 6. The overall structure of the student model RTCS-S.

2.3. Knowledge Distillation Method

To effectively transfer discriminative crack representations from the teacher network to the lightweight student network, we propose a foreground-aware and boundary-aware knowledge distillation (FBKD) framework. Inspired by instance-aware hybrid distillation (IHD) [33], which integrates feature-based and relation-based distillation, our FBKD framework is specifically streamlined for crack segmentation by focusing on critical foreground and boundary information.

As illustrated in Figure 7, RTCS-T is employed as the teacher network, and RTCS-S serves as the lightweight student network. Knowledge transfer is performed on the output

feature maps of the four feature extraction stages, the CASPP output feature, and the final predicted crack mask. Let F_i^T and F_i^S ($i=1,2,3,4$) denote the stage-wise feature maps of RTCS-T and RTCS-S, respectively, F_c^T and F_c^S represent the output feature maps of CASPP, and P^T and P^S denote the final crack probability maps. To emphasize crack-relevant regions during knowledge transfer, a foreground-aware and boundary-aware weighting map is constructed as follows:

$$W = \lambda_1 Y + \lambda_2 B \quad (18)$$

where Y denotes the ground-truth crack mask, B represents the corresponding boundary map extracted from Y , and λ_1 and λ_2 are weighting coefficients that control the contributions of the foreground and boundary regions, respectively. Since the boundary region usually contains finer structural details and is more difficult to segment accurately, λ_2 can be set larger than λ_1 . The boundary map B is obtained by the morphological gradient of Y , and the process can be expressed as follows:

$$B = \text{Dilate}(Y) - \text{Erode}(Y) \quad (19)$$

where $\text{Dilate}(\cdot)$ and $\text{Erode}(\cdot)$ denote the morphological dilation and erosion operations, respectively. The weighting map is subjected to a simple min-max normalization to obtain the final weighting mask $\hat{W} \in [0,1]^{H \times W}$, where H and W denote the spatial dimensions of the ground-truth mask. For each distilled feature map, $\hat{W}$ is further resized to the corresponding spatial resolution, yielding $\hat{W}_i \in [0,1]^{H_i \times W_i}$. In this way, knowledge transfer is concentrated on crack body regions and boundary details rather than on irrelevant background. The feature distillation loss is imposed on the four stage-wise feature maps and the CASPP feature map as follows:

$$L_{\text{feat}} = \sum_{i=1}^4 \omega_i \left\| (F_i^S - F_i^T) \odot \hat{W}_i \right\|_{\text{F}}^2 + \omega_c \left\| (F_c^S - F_c^T) \odot \hat{W}_c \right\|_{\text{F}}^2 \quad (20)$$

where ω_i and ω_c are balancing coefficients. To reflect the increasing semantic importance of deeper representations, ω_i and ω_c are empirically set as $\omega_1 = 0.10$, $\omega_2 = 0.15$, $\omega_3 = 0.20$, $\omega_4 = 0.25$, and $\omega_c = 0.30$ in our experiments.

In addition to local feature imitation, spatial relation distillation is introduced to preserve the structural continuity of cracks. Spatial relation distillation is applied exclusively to the CASPP feature, as it captures the most comprehensive multi-scale contextual information while maintaining a relatively low spatial resolution. The weighted teacher and student feature maps are obtained as follows:

$$\hat{F}_c^T = F_c^T \odot W_c, \quad \hat{F}_c^S = F_c^S \odot W_c \quad (21)$$

After reshaping them into $N \times C$ matrices, where $N = H \times W$, the spatial relation matrices are computed as follows:

$$R^T = \text{Norm}(\hat{F}_c^T (\hat{F}_c^T)^\top), \quad R^S = \text{Norm}(\hat{F}_c^S (\hat{F}_c^S)^\top) \quad (22)$$

and the relation distillation loss is defined as:

$$L_{\text{rel}} = \left\| R^S - R^T \right\|_{\text{F}}^2 \quad (23)$$

This term enables RTCS-S to mimic the structural dependency encoded by RTCS-T, thereby improving the continuity and integrity of segmented crack patterns.

To further transfer the refined prediction behavior of the teacher, an output distillation loss is introduced on the final probability maps:

$$L_{\text{out}} = \left\| (P^S - P^T) \odot \hat{W} \right\|_F^2 \quad (24)$$

The feature, spatial relation, and output distillation losses complement the hard supervision from the ground-truth labels and are particularly beneficial for ambiguous crack boundaries and low-contrast crack regions. The hard supervised segmentation loss of RTCS-S is defined as:

$$L_{\text{seg}} = \lambda L_{\text{bce}} + (1 - \lambda) L_{\text{dice}} \quad (25)$$

where L_{bce} and L_{dice} denote the binary cross-entropy loss and Dice loss, respectively, and λ represents the weight assigned to the two loss functions, with a range between 0 and 1. The segmentation loss L_{seg} is also used as the loss function for training RTCS-T. Finally, the overall loss function of FBKD is formulated as:

$$L_{\text{total}} = L_{\text{seg}} + \alpha L_{\text{feat}} + \beta L_{\text{rel}} + \gamma L_{\text{out}} \quad (26)$$

where α , β and γ are trade-off coefficients. Gou et al. [37] categorized the knowledge transferred in knowledge distillation as response-based, feature-based, and relation-based, which provide complementary supervision to the student network. Consistent with this categorization, the proposed FBKD framework uses feature distillation as the principal knowledge-transfer component because it directly aligns the multi-scale intermediate representations of RTCS-T and RTCS-S. Therefore, α is set to 1. Spatial relation distillation and output distillation provide complementary constraints on crack-structure dependency and final prediction behavior, respectively. β and γ are both set to 0.5 to preserve these complementary forms of knowledge while preventing the auxiliary objectives from excessively constraining the optimization of the lightweight student network. The same coefficients are fixed across all datasets and experiments to maintain a consistent training configuration and avoid dataset-specific adjustment.

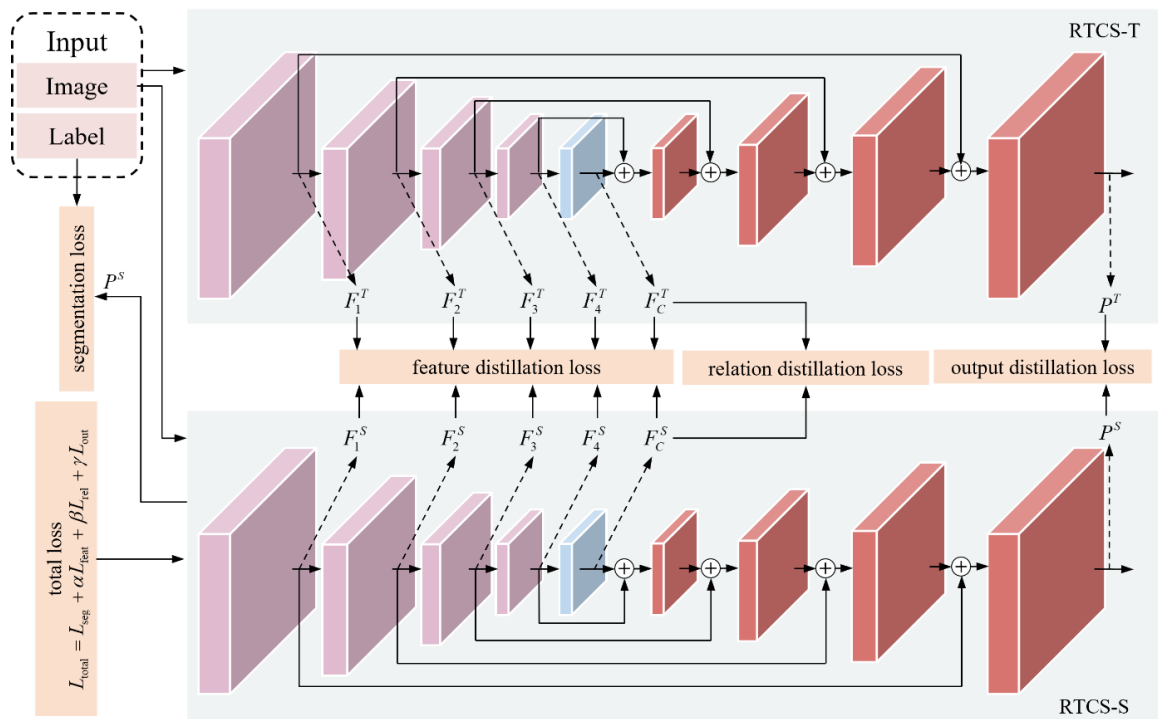


Figure 7. The overall structure of the knowledge distillation method.

3. Experiments

3.1. Experimental Setting

Hardware equipment: The proposed methods were evaluated on a desktop workstation equipped with an NVIDIA GeForce RTX 4090 GPU with 24 GB of memory, running Ubuntu 18.04, CUDA 11.2, and cuDNN 8.7.0. For a fair comparison, all segmentation frameworks involved in this study were reimplemented in PyTorch 2.0.1 under the same experimental settings.

Hyperparameters: All models were trained under the same training conditions to ensure that the performance differences were entirely attributed to the model architecture rather than external factors. Specifically, the batch size was set to 2, the initial learning rate was set to 1×10^{-3} , and the maximum number of training epochs was fixed at 100. The optimizer was configured as Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 2×10^{-4} .

3.2. Datasets

To evaluate the effectiveness of the proposed models, experiments were performed on three publicly available pavement crack segmentation datasets, including Crack500, [38] DeepCrack, [39] and CFD [40]. For Crack500 and DeepCrack, the original dataset settings were adopted, while CFD was randomly divided into training, validation, and test sets at a ratio of 7:1:2, respectively. Moreover, for ablation analysis, 15% of the images were randomly sampled from each dataset to construct a dedicated ablation dataset for comparative experiments. Considering the limited availability of annotated samples, data augmentation was adopted in accordance with common practice in existing studies. Specifically, flipping, shifting, and rotation were applied to enlarge the training set, and both the original and augmented images were used during training. Furthermore, all input images were resized to a uniform resolution of 224×224 pixels, with pixel intensities normalized to the range of $[0, 1]$, thereby improving training stability and facilitating faster convergence.

3.3. Evaluation Metrics

To comprehensively and objectively assess the performance of different models, four widely used evaluation metrics in crack segmentation are adopted in this study: Precision, Recall, F1 Score, and binary mean intersection over union (mIoU). These metrics provide a quantitative evaluation of the agreement between the predicted results and the ground truth, with higher values indicating better segmentation performance. The mathematical formulations are expressed as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (27)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (28)$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (29)$$

$$\text{mIoU} = \frac{1}{2} \left(\frac{TP}{TP + FP + FN} + \frac{TN}{TN + FP + FN} \right) \quad (30)$$

3.4. Compared Methods

To further evaluate the performance of RTCS-T and RTCS-S, ten representative segmentation models were selected as comparison baselines, covering different architectural paradigms and levels of computational complexity. The five non-lightweight models

included U-Net, CT-CrackSeg [41], DTrC-Net [42], HoVer-UNet [43], and Hybrid-Seg [44], representing conventional encoder–decoder, convolution–Transformer, hybrid, and knowledge-distillation-based segmentation architectures. The five lightweight models included EfficientNet, ShuffleNetV2, MobileNetV3, CarNet [45], and Crackmer [46]. EfficientNet, ShuffleNetV2, and MobileNetV3 were selected as widely used lightweight visual backbones, whereas CarNet and Crackmer were included as compact models specifically developed for crack analysis. This combination enables a comprehensive comparison of segmentation performance and computational efficiency across general-purpose, crack-specific, and lightweight architectures.

Although HoVer-UNet was originally developed for multi-class cell-nuclei segmentation rather than pavement crack segmentation, it was included as a cross-domain dense-segmentation baseline because its U-Net-based knowledge-distillation framework performs fine-grained pixel-level prediction and boundary delineation. These characteristics are also relevant to pavement crack segmentation, which requires accurate localization of thin structures and complex boundaries. All comparison models were trained and evaluated using identical dataset partitions and training conditions. Each model was independently trained ten times, and the results are reported as the mean $\pm$ standard deviation across the ten runs.

4. Results

4.1. Ablation Study

4.1.1. Ablation Study of the Segmentation Loss Function

The proportion of crack pixels to background pixels is usually very small. Assigning equal weights during loss calculation may cause the model to focus more on predicting background pixels while neglecting crack pixels, ultimately reducing segmentation accuracy. In this study, a hybrid loss function is used to effectively address the class imbalance problem and enhance the model's ability to capture fine-grained crack features. As shown in Figure 8, compared to using a single loss function, the hybrid loss function demonstrates superior performance in segmentation tasks with imbalanced data.

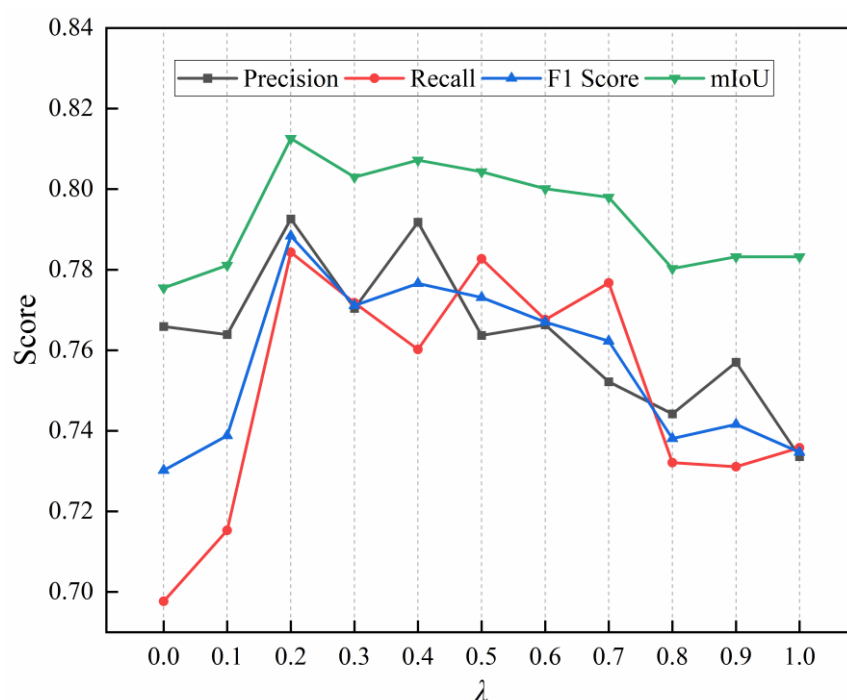


Figure 8. Model performance with different λ .

To further investigate the effectiveness of the combined binary cross-entropy and Dice loss for crack segmentation, the model performance was systematically evaluated under different weight ratios by varying the contribution of the two loss functions. The results show that when λ was set to 0.2, the model achieved the best performance across all evaluation metrics. The hybrid loss function enabled the model to mitigate the severe class imbalance between crack and background pixels more effectively, thereby improving the completeness and continuity of crack segmentation tasks.

4.1.2. Ablation Study of the Modules

The ablation study was conducted to evaluate the validity of the proposed modules, i.e., SRM and CASPP. Four variants of RTCS-T were compared on the ablation dataset:

- (1) Base, which removes both SRM and CASPP;
- (2) RTCS-T w/o SRM;
- (3) RTCS-T w/o CASPP;
- (4) The complete RTCS-T, in which both SRM and CASPP are incorporated.

The testing results under the evaluation metrics are shown in Table 1, all of which can be used to demonstrate the effectiveness of the proposed modules. The upward arrow ($\uparrow$) indicates an improvement relative to the baseline, and the numerical value following the symbol denotes the magnitude of the increase in the corresponding metric. Specifically, when both modules were incorporated, the segmentation performance was significantly improved, with Precision, Recall, F1 Score, and mIoU increasing by 4.19%, 6.20%, 5.17%, and 3.43% on the ablation dataset. These results demonstrate that both SRM and CASPP are effective, and their combination yields the most favorable segmentation performance. When the two modules are jointly employed, the network is able to simultaneously exploit local structural details and contextual feature maps, resulting in more accurate and complete crack segmentation. Therefore, the ablation results fully validate the rationality and effectiveness of the proposed modules.

Table 1. Ablation study for the effectiveness of the proposed modules on ablation dataset.

Methods	Precision	Recall	F1 Score	mIoU
Base	0.7148	0.7257	0.7202	0.7684
RTCS-T w/o SRM	0.7303 $\uparrow$ _{0.0155}	0.7560 $\uparrow$ _{0.0303}	0.7429 $\uparrow$ _{0.0227}	0.7736 $\uparrow$ _{0.0052}
RTCS-T w/o CASPP,	0.7547 $\uparrow$ _{0.0399}	0.7719 $\uparrow$ _{0.0462}	0.7632 $\uparrow$ _{0.0430}	0.7973 $\uparrow$ _{0.0289}
the complete RTCS-T	0.7567 $\uparrow$ _{0.0419}	0.7877 $\uparrow$ _{0.0620}	0.7719 $\uparrow$ _{0.0517}	0.8027 $\uparrow$ _{0.0343}

To further verify the effectiveness of the proposed SRM and CASPP modules, Grad-CAM was used to qualitatively analyze the spatial attention distributions across different ablation variants. As shown in Figure 9, the activation map of the base model without SRM and CASPP is widely dispersed over the background region, indicating limited discriminative capability in complex pavement textures. When only CASPP is introduced, the activation distribution becomes cleaner, suggesting that multi-scale contextual aggregation helps suppress irrelevant background interference; however, the crack region is still not sufficiently emphasized. In contrast, the model with only SRM exhibits stronger responses to the crack structure, but also produces redundant activations in texture-rich regions, reflecting its high sensitivity to local directional patterns. By jointly introducing SRM and CASPP, the activation map becomes more concentrated on the crack region while background responses are effectively suppressed. This demonstrates that SRM and CASPP play complementary roles in enhancing local structural representation and global contextual discrimination, respectively.

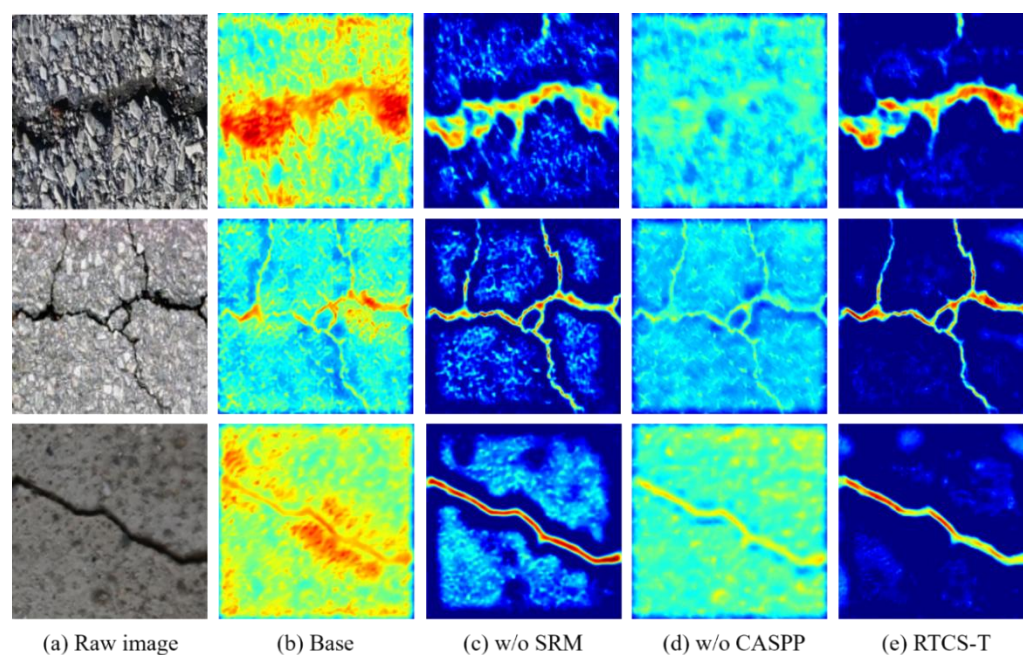


Figure 9. Grad-CAM visualization of the ablation variants. From left to right: input image, Base, Base + CASPP, Base + SRM, and the complete RTCS-T. Warmer colors indicate stronger activation responses, whereas cooler colors indicate weaker responses.

4.1.3. Ablation Study of the Knowledge Distillation Method

To isolate the contribution of the proposed knowledge distillation strategy, four variants of RTCS-S were evaluated on the same ablation dataset: (1) RTCS-S without knowledge distillation (w/o KD), which was trained using only the supervised segmentation loss; (2) RTCS-S with conventional knowledge distillation [32], in which the output probabilities of RTCS-T were used to supervise the student network; (3) RTCS-S with IHD; and (4) RTCS-S with the proposed FBKD. All variants employed identical teacher and student architectures, data partitions, training schedules, and optimization settings, with only the distillation objective being changed to ensure a fair comparison.

As shown in Table 2, conventional KD improved the segmentation performance of RTCS-S compared with training without knowledge distillation, indicating that the predictions of RTCS-T provided useful supplementary supervision for the lightweight student network. IHD further improved the performance by transferring both feature-level and relational information. The proposed FBKD achieved a Precision, Recall, F1 Score, and mIoU of 0.7364, 0.7649, 0.7504, and 0.7734, respectively. Compared with RTCS-S without KD, FBKD increased these four metrics by 6.08%, 7.12%, 6.59%, and 3.89%, respectively. Moreover, FBKD improved the F1 Score and mIoU by 1.01% and 0.47%, respectively, compared with IHD, demonstrating its advantage over the existing hybrid distillation strategy.

These results are consistent with the design motivation of FBKD. Conventional KD transfers teacher knowledge without explicitly distinguishing crack pixels from the dominant background, whereas IHD does not specifically account for the sparse foreground and fine boundaries of pavement cracks. By introducing foreground- and boundary-aware weighting, FBKD concentrates knowledge transfer on crack regions and boundary details. Meanwhile, spatial relation distillation encourages RTCS-S to preserve the structural dependencies between crack pixels. The improvement over conventional KD and IHD therefore indicates that task-specific foreground, boundary, and structural supervision is more effective for transferring crack representations from RTCS-T to the lightweight RTCS-S.

Table 2. Ablation study of different knowledge distillation strategies on the ablation dataset.

Methods	Precision	Recall	F1 Score	mIoU
RTCS-S w/o KD	0.6756	0.6937	0.6845	0.7345
RTCS-S + KD	0.7074 _{↑0.0318}	0.7345 _{↑0.0408}	0.7208 _{↑0.0363}	0.7477 _{↑0.0132}
RTCS-S + IHD	0.7248 _{↑0.0492}	0.7566 _{↑0.0629}	0.7403 _{↑0.0558}	0.7687 _{↑0.0342}
Ours	0.7364 _{↑0.0608}	0.7649 _{↑0.0712}	0.7504 _{↑0.0659}	0.7734 _{↑0.0389}

4.2. Comparison Experiments

To comprehensively verify the performance of the proposed RTCS-T and RTCS-S, several widely adopted segmentation networks were selected for comparative experiments. Experiments were conducted on three public datasets, including Crack500, DeepCrack, and CFD, and the performance of the proposed models was evaluated using four evaluation metrics.

4.2.1. Comparison Experiments on Crack500

As shown in Table 3, RTCS-T achieved the best overall performance in terms of F1 Score and mIoU, with values of 0.7477 and 0.7940, respectively. Meanwhile, it also achieved relatively high Precision (0.7343) and Recall (0.7616), indicating a favorable balance between crack localization accuracy and segmentation completeness. Although Hybrid-Seg achieved the highest Precision (0.7387), its Recall was lower than that of RTCS-T, which limited its overall segmentation performance. By contrast, RTCS-T maintained more balanced results across all evaluation metrics, suggesting that the proposed architecture can more effectively capture crack structures while suppressing background interference. In addition, RTCS-S also demonstrated competitive performance, with an F1 Score of 0.7213 and an mIoU of 0.7775.

Table 3. Comparison of results for different segmentation methods on Crack500.

Method	Precision	Recall	F1 Score	mIoU
U-Net	0.6515 ± 0.0044	0.7092 ± 0.0056	0.6791 ± 0.0019	0.7405 ± 0.0006
CT-CrackSeg	0.6663 ± 0.0026	0.7831 ± 0.0032	0.7199 ± 0.0029	0.7652 ± 0.0008
DTrC-Net	0.6590 ± 0.0029	0.7303 ± 0.0031	0.6928 ± 0.0018	0.7482 ± 0.0005
HoVer-UNet	0.6707 ± 0.0102	0.6803 ± 0.0104	0.6828 ± 0.0015	0.7450 ± 0.0008
Hybrid-Seg	0.7387 ± 0.0023	0.7296 ± 0.0061	0.7341 ± 0.0013	0.7769 ± 0.0006
EfficientNet	0.7169 ± 0.0043	0.7287 ± 0.0107	0.7228 ± 0.0017	0.7707 ± 0.0012
ShuffleNetV2	0.6399 ± 0.0027	0.7056 ± 0.0036	0.6711 ± 0.0018	0.7327 ± 0.0010
MobileNetV3	0.7032 ± 0.0132	0.6621 ± 0.0126	0.6820 ± 0.0027	0.7503 ± 0.0014
CarNet	0.6863 ± 0.0035	0.7002 ± 0.0062	0.6932 ± 0.0019	0.7492 ± 0.0011
Crackmer	0.6551 ± 0.0041	0.7134 ± 0.0074	0.6830 ± 0.0021	0.7286 ± 0.0012
RTCS-T	0.7343 ± 0.0029	0.7616 ± 0.0038	0.7477 ± 0.0017	0.7940 ± 0.0008
RTCS-S	0.7031 ± 0.0027	0.7405 ± 0.0033	0.7213 ± 0.0013	0.7775 ± 0.0007

As illustrated in Figure 10, the proposed RTCS-T and RTCS-S maintain relatively smooth growth trends throughout the training process. RTCS-T achieved consistently strong performance with superior overall convergence. In Figure 11, the first and second rows depict the qualitative comparison results of samples selected from the Crack500 dataset. In the first row, RTCS-T generated segmentation results that were more consistent with the ground truth, particularly in preserving crack continuity and local structures. It further showed stronger capability in detecting thin and elongated crack regions, producing clearer crack responses with fewer missing details. RTCS-S also exhibited competitive visual performance, with complete and stable segmentation results.

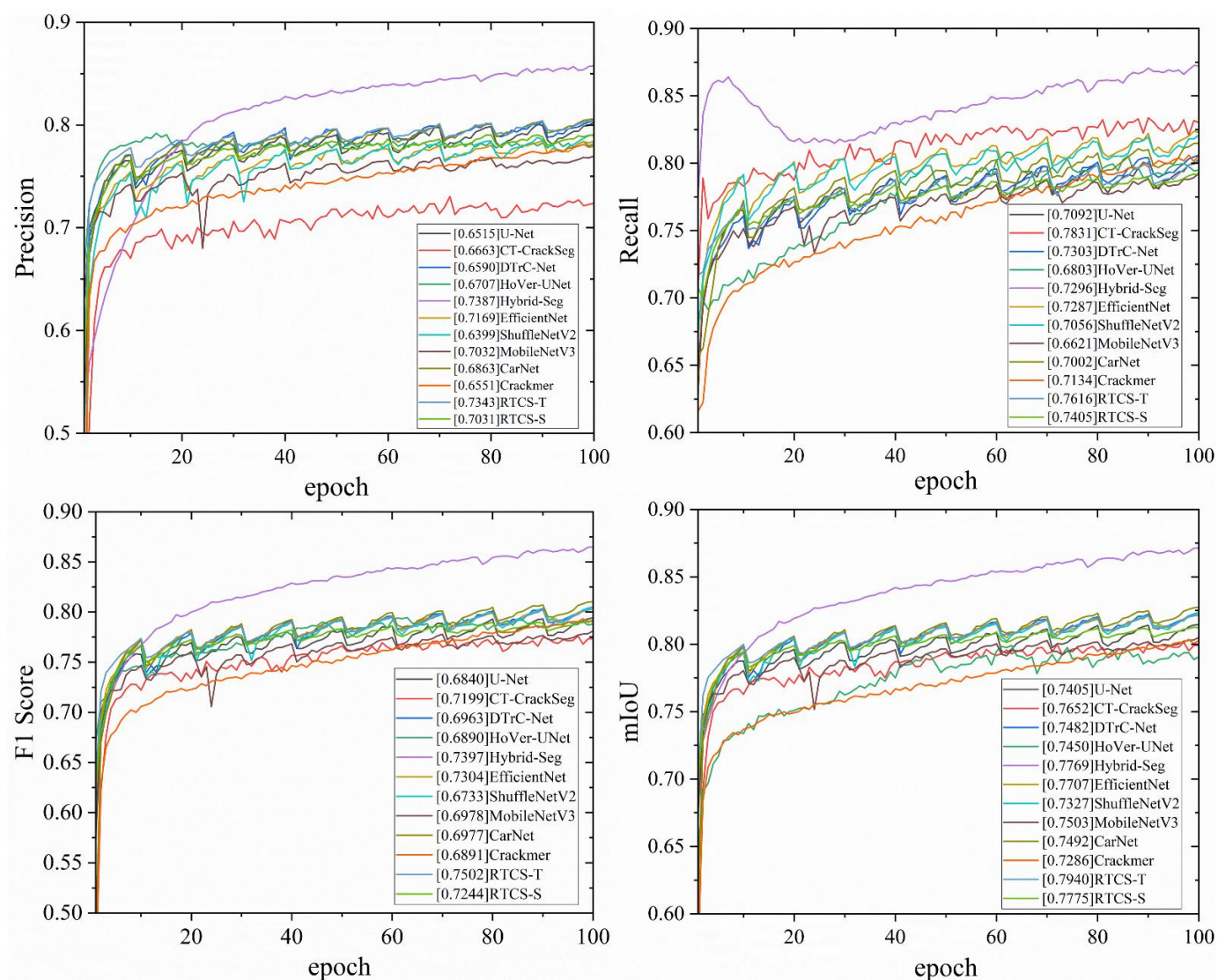


Figure 10. Training curves on the Crack500 dataset.

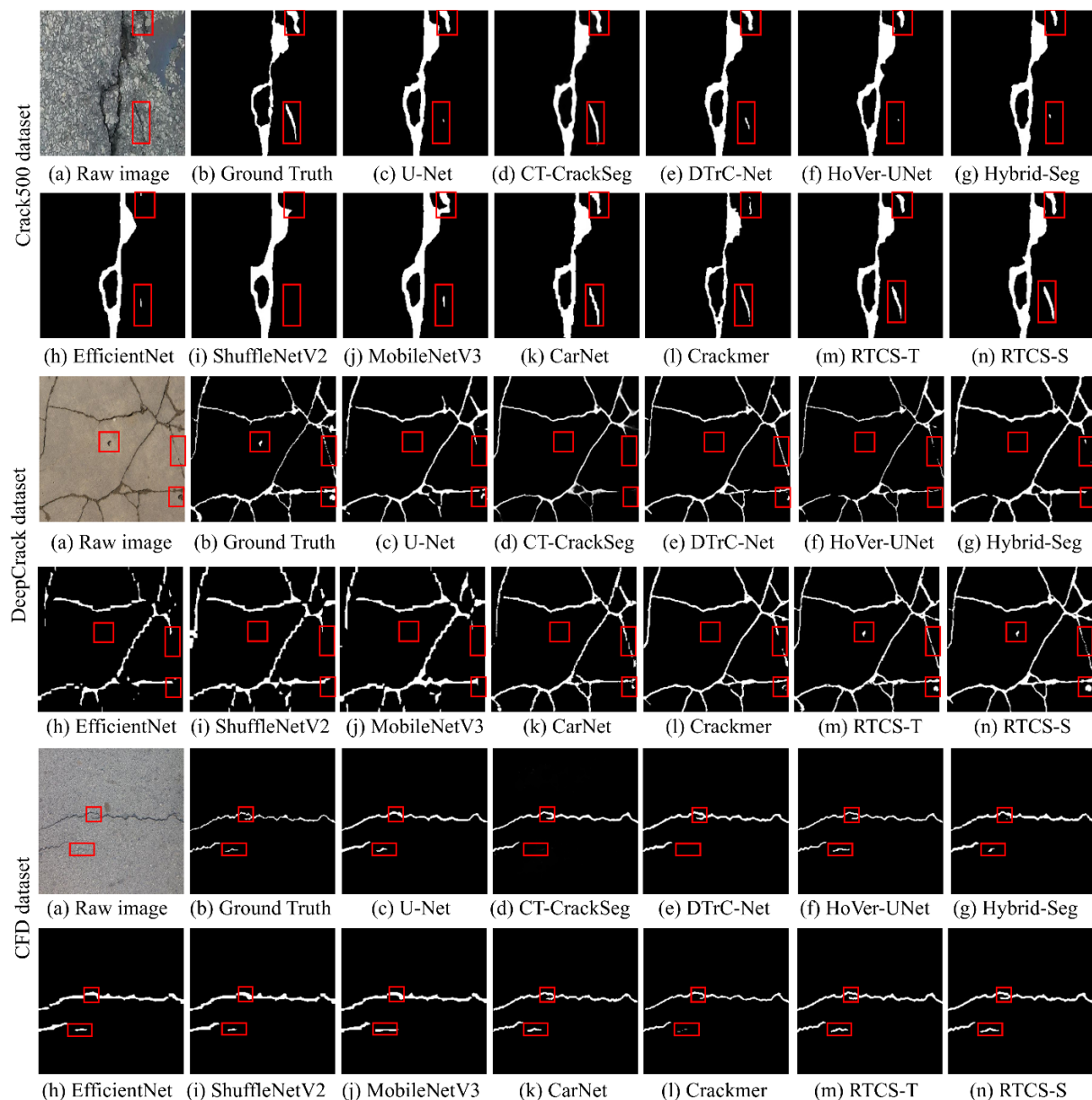


Figure 11. Visualization of segmentation results on experimental datasets.

4.2.2. Comparison Experiments on DeepCrack

As shown in Table 4, it is evident that RTCS-T achieved the best comprehensive performance among all compared methods. Specifically, it obtained the highest Precision of 0.8884, F1 Score of 0.8491, and mIoU of 0.8678, while maintaining a high Recall of 0.8131. Although Crackmer achieved the highest Recall of 0.8361, its relatively lower Precision reduced its overall F1 Score and mIoU, suggesting that it tended to generate more false positive predictions. CarNet also showed strong performance on this dataset, ranking second in F1 Score with 0.8369 and mIoU with 0.8582. Moreover, RTCS-S achieved the second-highest Precision of 0.8826 and maintained competitive results on the other evaluation metrics, indicating that the student model can effectively inherit discriminative knowledge from the teacher network.

As illustrated in Figure 12, RTCS-T converges rapidly and reaches higher final values than most compared methods, especially in F1 Score and mIoU, demonstrating its strong ability to capture discriminative crack features. RTCS-S also shows competitive convergence trends, indicating that the lightweight student model is able to effectively inherit useful knowledge from the teacher network. The third and fourth rows of Figure 11 show

the qualitative comparison results on the DeepCrack dataset. In this example, where the crack network is more complex and contains multiple intersections, RTCs-S captured fine crack patterns more accurately and reduced incomplete predictions in the marked areas. RTCs-T also demonstrated competitive visual performance, producing coherent and stable segmentation results on the DeepCrack dataset.

Table 4. Comparison of results for different segmentation methods on DeepCrack.

Method	Precision	Recall	F1 Score	mIoU
U-Net	0.8663 $\pm$ 0.0064	0.7952 $\pm$ 0.0051	0.8292 $\pm$ 0.0019	0.8492 $\pm$ 0.0009
CT-CrackSeg	0.8250 $\pm$ 0.0046	0.7958 $\pm$ 0.0034	0.8101 $\pm$ 0.0024	0.8369 $\pm$ 0.0011
DTrC-Net	0.8654 $\pm$ 0.0067	0.7917 $\pm$ 0.0053	0.8269 $\pm$ 0.0027	0.8437 $\pm$ 0.0015
HoVer-UNet	0.8768 $\pm$ 0.0062	0.7891 $\pm$ 0.0043	0.8306 $\pm$ 0.0028	0.8489 $\pm$ 0.0017
Hybrid-Seg	0.8642 $\pm$ 0.0049	0.7668 $\pm$ 0.0037	0.8126 $\pm$ 0.0019	0.8365 $\pm$ 0.0016
EfficientNet	0.8106 $\pm$ 0.0054	0.7871 $\pm$ 0.0047	0.7987 $\pm$ 0.0022	0.8270 $\pm$ 0.0014
ShuffleNetV2	0.8112 $\pm$ 0.0106	0.7959 $\pm$ 0.0086	0.8035 $\pm$ 0.0033	0.8266 $\pm$ 0.0029
MobileNetV3	0.8001 $\pm$ 0.0043	0.7742 $\pm$ 0.0052	0.7869 $\pm$ 0.0017	0.8189 $\pm$ 0.0013
CarNet	0.8672 $\pm$ 0.0053	0.8087 $\pm$ 0.0045	0.8369 $\pm$ 0.0019	0.8582 $\pm$ 0.0015
Crackmer	0.8111 $\pm$ 0.0067	0.8361 $\pm$ 0.0048	0.8234 $\pm$ 0.0023	0.8347 $\pm$ 0.0017
RTCS-T	0.8884 $\pm$ 0.0053	0.8131 $\pm$ 0.0046	0.8491 $\pm$ 0.0017	0.8678 $\pm$ 0.0008
RTCS-S	0.8826 $\pm$ 0.0044	0.7951 $\pm$ 0.0036	0.8366 $\pm$ 0.0015	0.8567 $\pm$ 0.0009

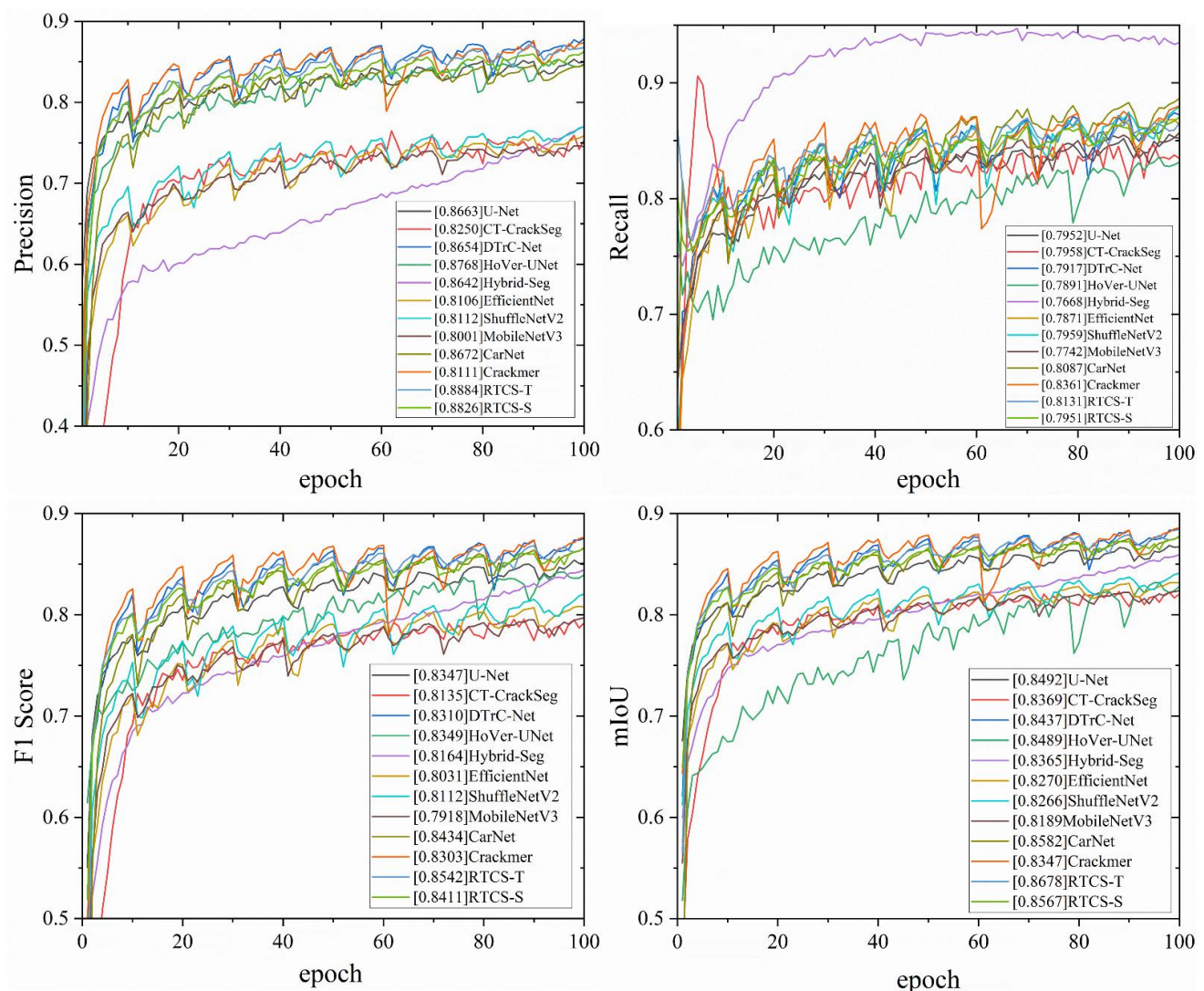


Figure 12. Training curves on the DeepCrack dataset.

4.2.3. Comparison Experiments on CFD

As shown in Table 5, the proposed RTCS-S achieved the best performance in Recall, F1 Score, and mIoU, with values of 0.7811, 0.7514, and 0.7962, respectively. This result indicates that RTCS-S was more effective in detecting complete crack regions under relatively complex background conditions. Although CarNet achieved the highest Precision of 0.7257, its Recall and overall segmentation performance were lower than those of RTCS-S, indicating that it was more conservative in crack prediction. In comparison, RTCS-S provided a better balance between segmentation accuracy and crack completeness, leading to superior overall results. RTCS-T also achieved stable performance on this dataset, with a Precision, Recall, F1 Score, and mIoU of 0.7228, 0.7443, 0.7334, and 0.7866, respectively. These results further demonstrate the robustness of the proposed frameworks across different crack datasets.

Table 5. Comparison of results for different segmentation methods on CFD.

Method	Precision	Recall	F1 Score	mIoU
U-Net	0.7181 ± 0.0053	0.7567 ± 0.0043	0.7369 ± 0.0013	0.7929 ± 0.0008
CT-CrackSeg	0.6627 ± 0.0075	0.7272 ± 0.0065	0.6935 ± 0.0024	0.7798 ± 0.0014
DTrC-Net	0.6933 ± 0.0089	0.7598 ± 0.0086	0.7250 ± 0.0019	0.7843 ± 0.0017
HoVer-UNet	0.6979 ± 0.0063	0.6995 ± 0.0031	0.6987 ± 0.0014	0.7668 ± 0.0009
Hybrid-Seg	0.6846 ± 0.0130	0.6469 ± 0.0096	0.6652 ± 0.0055	0.7471 ± 0.0018
EfficientNet	0.6040 ± 0.0051	0.6475 ± 0.0028	0.6250 ± 0.0022	0.7206 ± 0.0011
ShuffleNetV2	0.5968 ± 0.0090	0.7061 ± 0.0064	0.6469 ± 0.0076	0.7326 ± 0.0024
MobileNetV3	0.6006 ± 0.0054	0.6602 ± 0.0049	0.6290 ± 0.0062	0.7203 ± 0.0019
CarNet	0.7257 ± 0.0051	0.7760 ± 0.0034	0.7499 ± 0.0026	0.7943 ± 0.0015
Crackmer	0.5967 ± 0.0033	0.5939 ± 0.0027	0.5953 ± 0.0031	0.7127 ± 0.0017
RTCS-T	0.7228 ± 0.0051	0.7443 ± 0.0042	0.7334 ± 0.0029	0.7866 ± 0.0011
RTCS-S	0.7239 ± 0.0047	0.7811 ± 0.0035	0.7514 ± 0.0023	0.7962 ± 0.0012

As illustrated in Figure 13, although the training curves exhibit relatively larger fluctuations due to the limited dataset size, RTCS-S gradually surpasses most compared methods in the later stages of training and achieves the best overall convergence performance. The fifth and sixth rows of Figure 11 present the qualitative comparison results on the CFD dataset. RTCS-S generated predictions that more closely matched the ground truth, particularly in identifying thin and less salient crack traces and producing fewer missing predictions in the highlighted regions. RTCS-T also demonstrated competitive visual performance, yielding stable segmentation results on the CFD dataset.

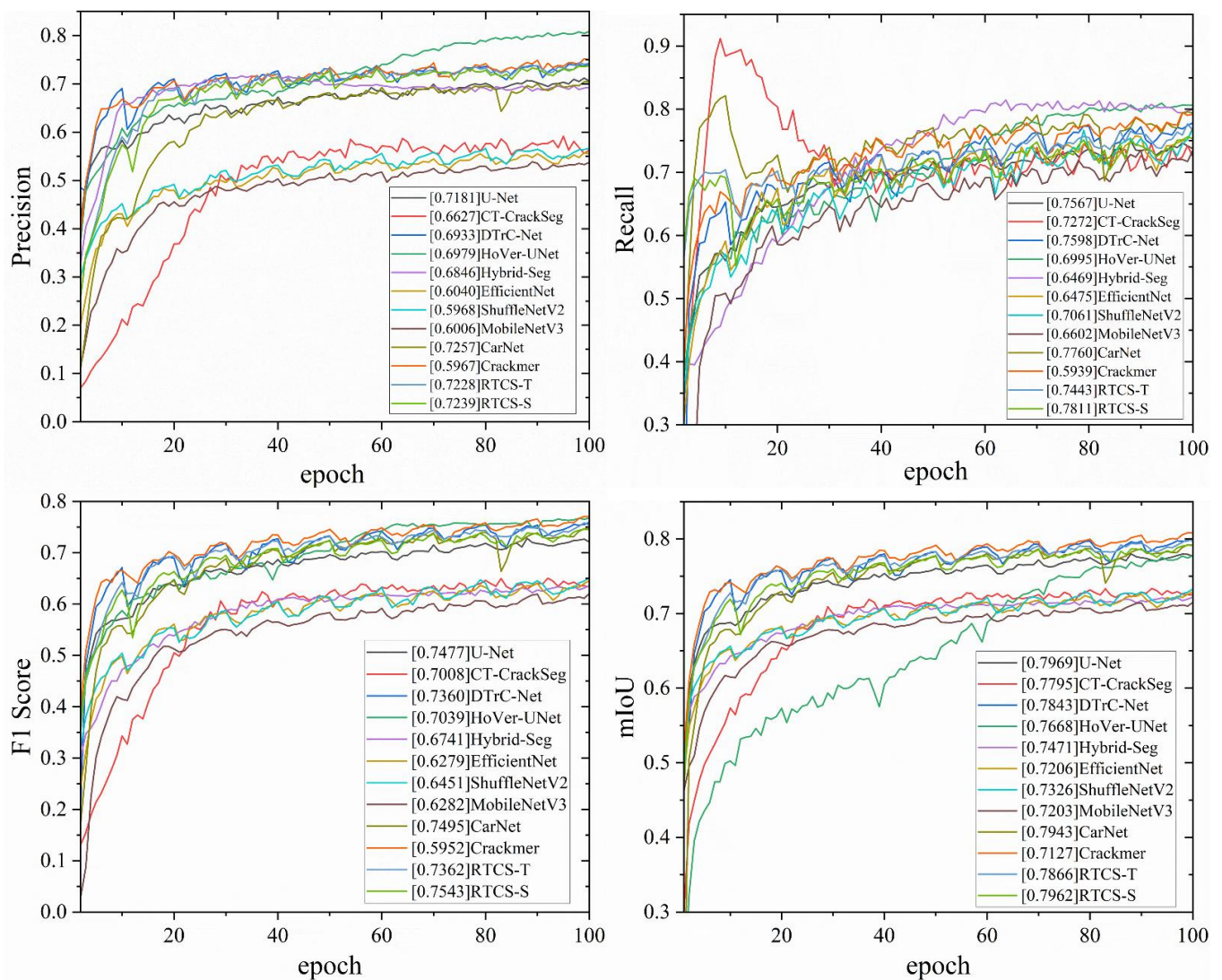


Figure 13. Training curves on the CFD dataset.

4.3. Limitation

Although the proposed models achieve satisfactory segmentation results in most cases, they exhibit limitations in some extremely challenging scenarios. In particular, no segmentation model can guarantee accurate crack segmentation for all samples. When the input images contain watermark text or similar background markings, the model may produce false positive predictions. As illustrated in Figure 14, both RTCS-T and RTCS-S mistakenly identified the watermark in the highlighted areas as cracks, although the main crack regions were still segmented well. This suggests that text-like background interferences may provide misleading structural patterns, thereby reducing the model's ability to distinguish true cracks from non-crack regions. Nevertheless, such failure cases occur only in a limited number of samples, and the proposed methods still demonstrate strong robustness and competitive segmentation performance in most situations.

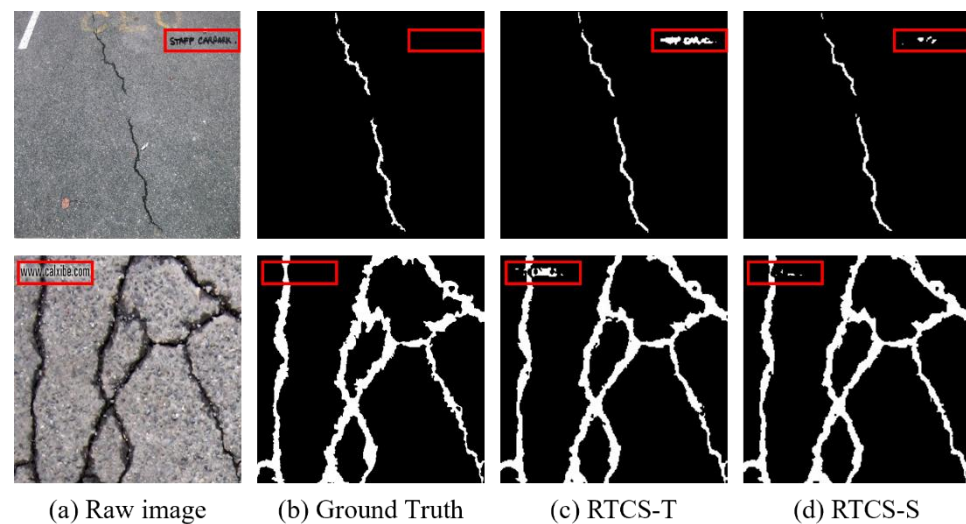


Figure 14. Failure cases of the proposed models. The red boxes indicate regions in which water-mark text is incorrectly segmented as cracks.

4.4. Comparison of Computational Complexity

In this subsection, the models analyzed in the previous sections are further evaluated in terms of parameter count and computational efficiency. The parameter count, reported in millions (M), refers to the number of trainable parameters in a model and is commonly used to reflect model capacity. A larger parameter count generally indicates stronger feature representation ability, but it also increases model complexity and resource consumption during both training and inference. To assess computational efficiency, frames per second (FPS), inference time, and floating-point operations (FLOPs) were adopted. FPS denotes the number of images that can be processed per second, with higher values indicating faster inference speed. Inference time is reported in milliseconds (ms) and represents the average time required to process a single image; lower values indicate faster inference. FLOPs represent the amount of floating-point computation required by a model and serve as an important indicator of computational complexity. In general, models with higher FLOPs require more computational resources and incur greater inference cost.

Figure 15 visually presents the parameter counts and computational complexity of various models. In this figure, the horizontal axis denotes FPS, while the vertical axis represents FLOPs. In addition, the size of each circle corresponds to the parameter count of a model, where a larger circle indicates a larger number of parameters. It is evident that different models exhibit distinct trade-offs among inference speed, computational complexity, and parameter count. Some models achieve stronger representation capability at the expense of higher computational burden and larger parameter counts, whereas lightweight models generally show lower FLOPs and faster inference speed. Notably, the proposed RTCS-T and RTCS-S are both located in regions with relatively low computational cost, and RTCS-S in particular achieves very high inference speed while maintaining a very small parameter count.

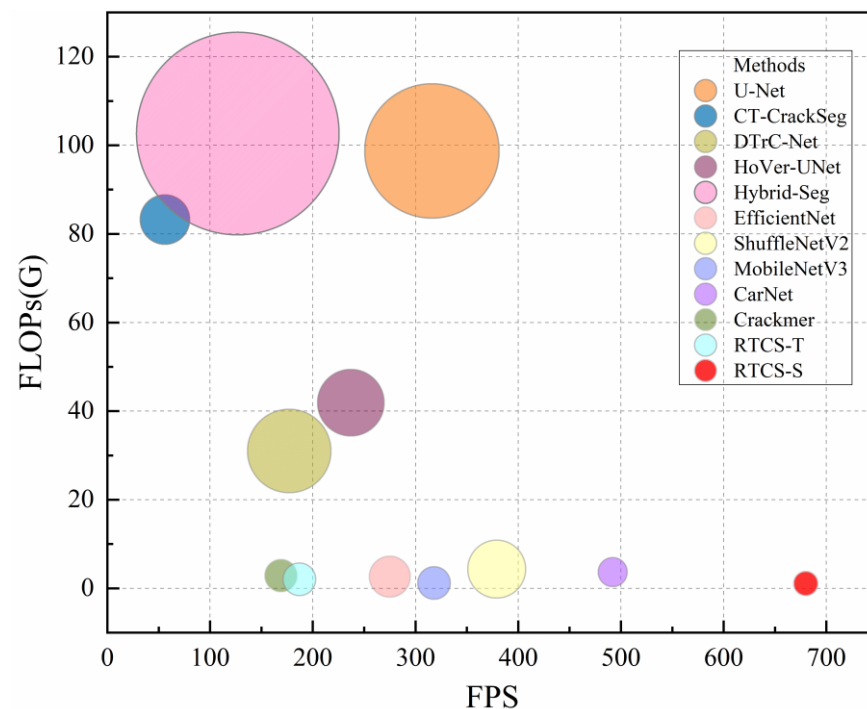


Figure 15. Comparison of parameter counts and computational complexity of each model.

Table 6 provides the parameter counts, FPS values, and FLOPs of all compared models. It can be observed that RTCS-S achieved the lowest model complexity among all compared methods, with only 1.82 M parameters and 1.13 GFLOPs. At the same time, it attained the highest inference speed of 680 FPS, demonstrating a significant advantage in computational efficiency. Although MobileNetV3 also exhibited relatively low computational cost, with 5.38 M parameters and 1.23 GFLOPs, its overall efficiency was still inferior to that of RTCS-S. CarNet achieved the second highest inference speed (492 FPS) and maintained a lightweight structure, but its parameter counts and FLOPs remained higher than those of RTCS-S. By contrast, traditional or heavier models such as U-Net and Hybrid-Seg required substantially larger parameter counts and computational costs, indicating lower efficiency in practical deployment scenarios. In addition, RTCS-T also showed favorable lightweight characteristics, with 4.83 M parameters and 2.03 GFLOPs, which were markedly lower than those of most compared methods. These results demonstrate that the proposed RTCS-S not only achieves strong segmentation performance, but also maintains excellent efficiency, making it more suitable for real-time crack segmentation tasks.

Table 6. Computational complexity with compared methods.

Method	Parameters (M)	FLOPs (G)	FPS	Inference Time (ms)
U-Net	134.19	98.73	316	3.2
CT-CrackSeg	22.88	83.24	56	17.9
DTrC-Net	37.86	30.99	177	5.7
HoVer-UNet	31.04	41.91	237	4.2
Hybrid-Seg	237.88	102.65	127	7.9
EfficientNet	8.64	2.63	275	3.6
ShuffleNetV2	9.54	4.35	379	2.6
MobileNetV3	5.38	1.23	318	3.1
CarNet	4.89	3.68	492	2.0
Crackmer	5.63	2.87	169	5.4
RTCS-T	4.83	2.03	187	5.3
RTCS-S	1.82	1.13	680	1.5

4.5. Results of Real-Time Deployment

To evaluate the practical applicability of RTCS-S, the model was deployed on an RDK X5 edge-computing development board, as shown in Figure 16. The board integrates an eight-core 1.5 GHz Arm Cortex-A55 CPU, an INT8-capable BPU with a computing capacity of 10 TOPS, and 4 GB of LPDDR4 memory. An 8-megapixel IMX219 camera was used to acquire pavement images at a resolution of 1280×720 pixels and a frame rate of 60 fps. The captured frames were processed online without offline preprocessing. Each frame was resized to the input resolution of RTCS-S and normalized before inference, after which the predicted crack mask was overlaid on the corresponding camera image for visualization.

The complete RTCS-S deployment pipeline achieved an end-to-end throughput of 34 FPS on the RDK X5 platform. This measurement covered image acquisition, online resizing and normalization, model inference, post-processing, and result visualization. The corresponding average end-to-end latency was approximately 29.4 ms, while the peak memory consumption during embedded execution was 1.8 GB. The measured throughput showed that RTCS-S could perform real-time pavement crack segmentation on the embedded device under the evaluated configuration.

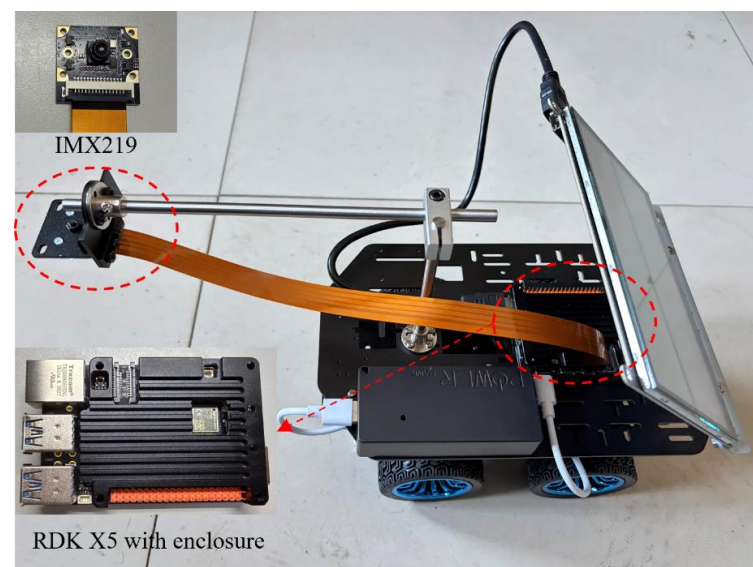


Figure 16. Real-time deployment platform.

Figure 17 presents several representative results obtained under different real-world pavement conditions. The first row contains an elongated crack. The second row shows a curved crack affected by irregular water stains, whose appearance may be confused with crack pixels. The third row presents a crack crossing a painted road marking. The fourth row contains several intersecting crack branches, whereas the fifth row presents an irregular crack network in a severely deteriorated region adjacent to a utility-cover boundary.

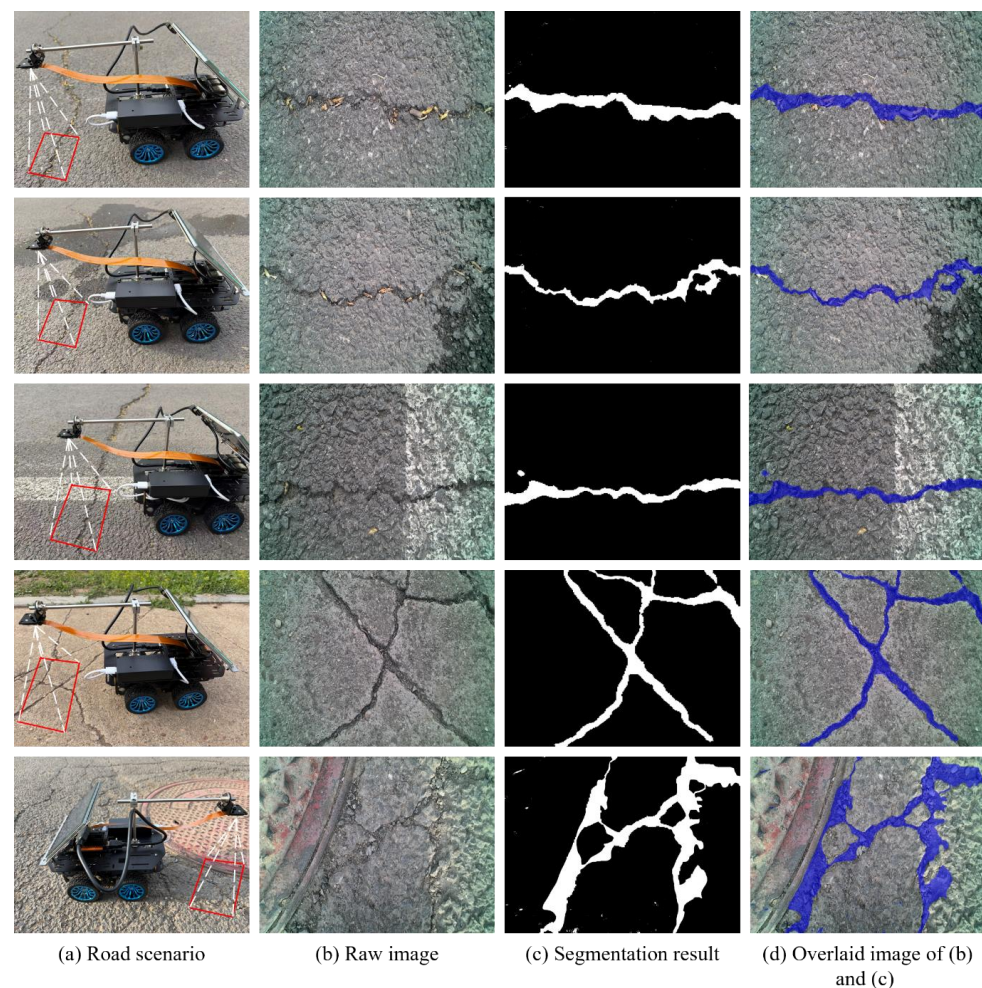


Figure 17. Qualitative results of real-time crack segmentation in different road scenarios.

In the first row, the predicted mask followed the elongated crack and retained its overall continuity. In the second row, the response remained concentrated along the visible crack trace rather than spreading across the surrounding water-stained region. In the road-marking scenario, RTCSS followed the crack across the marked pavement while producing only limited responses to the painted area. The principal branches and junctions were retained in the intersecting-crack scenario. In the final scene, the model detected the main crack network but produced broader responses in the severely deteriorated, highly textured region. These field observations provide qualitative evidence that RTCSS preserved crack continuity under the pavement conditions evaluated in the deployment experiment.

The complete RTCSS pipeline achieved an end-to-end throughput of 34 FPS on the RDK X5 platform, showing that the system operated in real time under the evaluated configuration. The representative field results further showed that the model retained the principal crack structures under interference from water stains, painted road markings, intersecting branches, and severely deteriorated pavement. Overall, the deployment results indicate that the proposed RTCSS model achieves a favorable balance between segmentation performance and computational efficiency, demonstrating its potential for real-time pavement distress detection and embedded intelligent inspection systems.

5. Conclusions and Future Work

In this study, a novel teacher–student crack segmentation framework, namely RTCSS-T and RTCSS-S, was proposed to address the challenging trade-off between segmentation

accuracy and computational efficiency in pavement crack detection tasks. The teacher network, RTCS-T, was constructed based on the Swin Transformer to effectively capture long-range dependencies and multi-scale contextual information, while the introduced strip refinement module and cascaded atrous spatial pyramid pooling module further enhanced the representation of elongated crack structures and complex contextual features. The lightweight student network, RTCS-S, was designed using depthwise separable convolutions and a hierarchical encoder–decoder architecture, enabling efficient inference with significantly reduced computational cost. In addition, a foreground-aware and boundary-aware knowledge distillation strategy was developed to guide the transfer of discriminative structural knowledge from the teacher to the student model.

Extensive experiments conducted on three public datasets, including Crack500, DeepCrack, and CFD, demonstrated that the proposed framework achieves competitive or superior performance compared with state-of-the-art methods. Specifically, RTCS-T achieved the best overall performance on Crack500, with an F1 Score of 0.7477 and an mIoU of 0.7940, and also obtained the best comprehensive results on DeepCrack, with a Precision of 0.8884, an F1 Score of 0.8491, and an mIoU of 0.8678. On CFD, RTCS-S achieved the best overall performance, with a Recall of 0.7811, an F1 Score of 0.7514, and an mIoU of 0.7962. In addition, RTCS-S exhibited high computational efficiency, requiring only 1.82 M parameters and 1.13 GFLOPs while achieving a model inference speed of 680 FPS on the RTX 4090 GPU. When deployed on the RDK X5 edge-computing platform, the complete end-to-end pipeline achieved a throughput of 34 FPS, with an average latency of approximately 29.4 ms and peak memory consumption of 1.8 GB. These results indicate that RTCS-S can perform real-time crack segmentation under the evaluated embedded configuration, supporting its potential for on-device pavement inspection and structural health monitoring applications.

Future work will focus on improving the discrimination ability of the model against visually similar background interference, so that crack patterns can be more accurately distinguished from non-crack structures. In addition, the integration of temporal information from video streams and multi-modal sensing frameworks will be further investigated to provide richer contextual cues and enhance the reliability and stability of real-time crack detection systems.

Author Contributions: N.X.: conceptualization, methodology, software, experiments, data analysis, and writing the original draft; J.Q.: supervision, project administration, funding acquisition, and manuscript review and editing. Y.T.: manuscript review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Postgraduate Education Reform Project of Liaoning Province [grant number LNYJG2022073], Postgraduate Education Reform Project of Shenyang University of Technology [grant number SYJG20222002], and Educational Department of Liaoning Province [grant number LJ212410142020]. The authors are grateful for their financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments and feedback, which were extremely helpful in improving the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fan, Q.; Xie, J.; Xia, Z.; Tan, Y.; Zhu, S.; Wang, J. A novel high-precision lightweight concrete bridge crack image recognition method based on knowledge distillation. *Struct. Health Monit.* **2025**, *24*, 3441–3453. <https://doi.org/10.1177/14759217241274277>.
2. Oliveira, H.; Correia, P.L. Automatic road crack segmentation using entropy and image dynamic thresholding. In *Proceedings of the 2009 17th European Signal Processing Conference, Glasgow, UK*; IEEE: Piscataway, NJ, USA, 2009; pp. 622–626.
3. Abdel-Qader, L.; Abudayyeh, O.; Kelly, M.E. Analysis of edge-detection techniques for crack identification in bridges. *J. Comput. Civ. Eng.* **2003**, *17*, 255–263. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2003\)17:4\(255\)](https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)).
4. Sobel, I.; Feldman, G. A 3x3 Isotropic Gradient Operator for Image Processing. In *Pattern Classification and Scene Analysis*; Wiley: New York, NY, USA, 1973; pp. 271–272.
5. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>.
6. Yamaguchi, T.; Nakamura, S.; Saegusa, R.; Hashimoto, S. Image-Based Crack Detection for Real Concrete Surfaces. *IEEJ Trans. Electr. Electron. Eng.* **2008**, *3*, 128–135. <https://doi.org/10.1002/tee.20244>.
7. Serra, J. Introduction to mathematical morphology. *Comput. Vis. Graph. Image Process.* **1986**, *35*, 283–305. [https://doi.org/10.1016/0734-189X\(86\)90002-2](https://doi.org/10.1016/0734-189X(86)90002-2).
8. Yu, Y.; Rashidi, M.; Samali, B.; Mohammadi, M.; Nguyen, T.N.; Zhou, X. Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm. *Struct. Health Monit.* **2022**, *21*, 2244–2263. <https://doi.org/10.1177/14759217211053546>.
9. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. <https://doi.org/10.1109/CVPR.2015.7298965>.
10. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, 5–9 October 2015; pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.
11. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. <https://doi.org/10.1109/TPAMI.2016.2644615>.
12. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890. <https://doi.org/10.1109/CVPR.2017.660>.
13. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. <https://doi.org/10.1109/TPAMI.2017.2699184>.
14. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 8–14 September 2018; pp. 833–851. https://doi.org/10.1007/978-3-030-01234-2_49.
15. Qin, S.; Yang, Y.; Chen, M. CrackMambaNet: A road crack segmentation network based on dual-branch encoder ensemble with adaptive feature fusion. *J. Infrastruct. Intell. Resil.* **2026**, *5*, 100194. <https://doi.org/10.1016/j.jintel.2025.100194>.
16. Sun, H.; Zhang, T.; Yu, M.; Ren, S.; Wang, D. Cross-layer context boundary guided network for crack segmentation. *Constr. Build. Mater.* **2025**, *498*, 143975. <https://doi.org/10.1016/j.conbuildmat.2025.143975>.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 4–9 December 2017.
18. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
19. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022. <https://doi.org/10.1109/ICCV48922.2021.00986>.
20. Guo, F.; Qian, Y.; Liu, J.; Yu, H. Pavement crack detection based on transformer network. *Autom. Constr.* **2023**, *145*, 104646. <https://doi.org/10.1016/j.autcon.2022.104646>.
21. Wang, C.; Liu, H.; An, X.; Gong, Z.; Deng, F. SwinCrack: Pavement crack detection using convolutional swin-transformer network. *Digit. Signal Process.* **2024**, *145*, 104297. <https://doi.org/10.1016/j.dsp.2023.104297>.

22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
23. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018; pp. 6848–6856. <https://doi.org/10.1109/CVPR.2018.00716>.
24. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
25. Liang, J.; Zhang, Q.; Gu, X. Lightweight convolutional neural network driven by small data for asphalt pavement crack segmentation. *Autom. Constr.* **2024**, *158*, 105214. <https://doi.org/10.1016/j.autcon.2023.105214>.
26. Dong, S.; Wang, Y.; Cao, J.; Yu, S.; Ma, J. Pavement crack segmentation with attention and progressive fusion in a lightweight framework. *Measurement* **2026**, *261*, 119891. <https://doi.org/10.1016/j.measurement.2025.119891>.
27. Zim, A.H.; Iqbal, A.; Al-Huda, Z.; Malik, A.; Kuribayashi, M. EfficientCrackNet: A Lightweight Model for Crack Segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Tucson, AZ, USA, 26 February–6 March 2025; pp. 6279–6289. <https://doi.org/10.1109/WACV61041.2025.00612>.
28. Mehta, S.; Rastegari, M. MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer. In Proceedings of the International Conference on Learning Representations (ICLR); Virtual Event, 25–29 April 2022.
29. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.-Y.; et al. Segment Anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 4015–4026. <https://doi.org/10.1109/ICCV51070.2023.00371>.
30. Zhang, C.; Han, D.; Qiao, Y.; Kim, J.U.; Bae, S.H.; Lee, S.; Hong, C.S. Faster Segment Anything: Towards Lightweight SAM for Mobile Applications. *arXiv* **2023**, arXiv:2306.14289.
31. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
32. Liu, Y.; Chen, K.; Liu, C.; Qin, Z.; Luo, Z.; Wang, J. Structured Knowledge Distillation for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 2604–2613. <https://doi.org/10.1109/CVPR.2019.00271>.
33. Zhang, J.; Ding, L.; Wang, W.; Wang, H.; Brilakis, I.; Davletshina, D.; Heikkilä, R.; Yang, X. Crack segmentation-guided measurement with lightweight distillation network on edge device. *Comput.-Aided Civ. Infrastruct. Eng.* **2025**, *40*, 2269–2286. <https://doi.org/10.1111/mice.13446>.
34. Wang, X.; Tang, Z.; Guo, J.; Meng, T.; Wang, C.; Wang, T.; Jia, W. Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models. *ACM Comput. Surv.* **2025**, *57*, 228. <https://doi.org/10.1145/3724420>.
35. Geng, X.; Wang, Z.; Chen, C.; Xu, Q.; Xu, K.; Jin, C.; Gupta, M.; Yang, X.; Chen, Z.; Aly, M.M.S.; et al. From Algorithm to Hardware: A Survey on Efficient and Safe Deployment of Deep Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 5837–5857. <https://doi.org/10.1109/TNNLS.2024.3394494>.
36. Hou, Q.; Zhang, L.; Cheng, M.; Feng, J. Strip pooling: Rethinking spatial pooling for scene parsing. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4003–4012. <https://doi.org/10.1109/CVPR42600.2020.00406>.
37. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819. <https://doi.org/10.1007/s11263-021-01453-z>.
38. 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. Transp. Syst.* **2020**, *21*, 1525–1535. <https://doi.org/10.1109/TITS.2019.2910595>.
39. Liu, Y.; Yao, J.; Lu, X.; Xie, R.; Li, L. DeepCrack: A Deep Hierarchical Feature Learning Architecture for Crack Segmentation. *Neurocomputing* **2019**, *338*, 139–153. <https://doi.org/10.1016/j.neucom.2019.01.036>.
40. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. <https://doi.org/10.1109/TITS.2016.2552248>.
41. Tao, H.; Liu, B.; Cui, J.; Zhang, H. A Convolutional-Transformer Network for Crack Segmentation with Boundary Awareness. In Proceedings of the 2023 IEEE International Conference on Image Processing (ICIP), Kuala Lumpur, Malaysia, 8–11 October 2023; pp. 86–90. <https://doi.org/10.1109/ICIP49359.2023.10222276>.
42. Xiang, C.; Guo, J.; Cao, R.; Deng, L. A Crack-Segmentation Algorithm Fusing Transformers and Convolutional Neural Networks for Complex Detection Scenarios. *Autom. Constr.* **2023**, *152*, 104894. <https://doi.org/10.1016/j.autcon.2023.104894>.

43. Tommasino, C.; Russo, C.; Rinaldi, A.M.; Ciompi, F. HoVer-UNet: Accelerating HoVerNet with UNet-Based Multi-Class Nuclei Segmentation Via Knowledge Distillation. In Proceedings of the 2024 IEEE International Symposium on Biomedical Imaging (ISBI), Athens, Greece, 27–30 May 2024; pp. 1–4. <https://doi.org/10.1109/ISBI56570.2024.10635755>.
44. Goo, J.M.; Milidonis, X.; Artusi, A.; Boehm, J.; Ciliberto, C. Hybrid-Segmentor: Hybrid Approach for Automated Fine-Grained Crack Segmentation in Civil Infrastructure. *Autom. Constr.* **2025**, *170*, 105960. <https://doi.org/10.1016/j.autcon.2024.105960>.
45. Li, K.; Yang, J.; Ma, S.W.; Wang, B.; Wang, S.; Tian, Y.; Qi, Z. Rethinking Lightweight Convolutional Neural Networks for Efficient and High-Quality Pavement Crack Detection. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 237–250. <https://doi.org/10.1109/TITS.2023.3307286>.
46. Wang, J.; Zeng, Z.G.; Sharma, P.K.; Alfarraj, O.; Tolba, A.; Zhang, J.; Wang, L. Dual-path network combining CNN and transformer for pavement crack segmentation. *Autom. Constr.* **2024**, *158*, 105217. <https://doi.org/10.1016/j.autcon.2023.105217>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.