Automation in Construction 181 (2026) 106635

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Context- and refinement-driven convolutional architecture for robust crack segmentation under real-world and zero-shot conditions

Ammar M. Okran [a],*, Domenec Puig [a], Saif Khalid Musluh [a,c], Sylvie Chambon [b], Hatem A. Rashwan [a]

[a] Department of Computer Engineering and Mathematics, Rovira i Virgili University, 43007 Tarragona, Spain
[b] Institut de Recherche en Informatique de Toulouse (IRIT), 31000, Toulouse, France
[c] University of Al-Qadisiyah, Iraq

ARTICLE INFO

Dataset link: https://github.com/AmmarOkran/CrackRefineNet

Keywords:
Artificial intelligence
Crack segmentation
Deep learning
Context-aware features
Adaptive feature refinement
Selective feature aggregation
Infrastructure inspection

ABSTRACT

Crack segmentation is essential for infrastructure monitoring but remains challenging due to complex textures, lighting variations, and the thin, fragmented nature of cracks. This paper presents a deep learning architecture, CrackRefineNet, which integrates three complementary modules, Context-Aware Modeling (CAM), Adaptive Feature Refinement (AFR), and Selective Feature Aggregation (SFA), to improve crack localization and boundary precision. Built on the ConvNeXt backbone, the method captures both fine-grained structures and global context by injecting spatial information into early layers and refining deep semantic features at later stages. Aggregated intermediate predictions guide final mask generation through a unified decoder. Comprehensive experiments on four benchmark datasets (RCFD, Crack500, CFD, and Sylvie) show that CrackRefineNet outperforms state-of-the-art methods by up to 3.5% in IoU and 2.8% in F1-score, achieving 42.91% F1-score and 63.03% mIoU in zero-shot testing, confirming its robustness and readiness for real-world applications.

1. Introduction

Cracks are among the earliest indicators of structural deterioration in civil infrastructure. If not addressed promptly, they can escalate into serious safety hazards, lead to costly repairs, and in extreme cases, cause catastrophic failures. Maintaining the integrity of roads, bridges, tunnels, and buildings requires timely inspections to detect such issues at an early stage. Traditionally, these inspections have been conducted manually by trained engineers. However, manual inspection is time-consuming, labor-intensive, and inherently subjective—its reliability varies with the inspector's expertise and environmental conditions. Moreover, comprehensive manual inspection of large-scale infrastructure, such as highways or multi-story buildings, is often impractical and economically unsustainable [1].

To overcome these limitations, researchers have explored automated crack detection techniques. Early efforts focused on classical image processing methods, such as thresholding, edge detection, and morphological operations. While effective under controlled conditions, these techniques often fail in real-world scenarios characterized by cluttered backgrounds, varying lighting, and complex surface textures [2].

Machine learning (ML) methods, including Support Vector Machines [3] (SVMs) and Random Forests [4], improved robustness by leveraging hand-crafted features. However, their performance is limited by the quality and generalizability of these features. These models frequently struggle with faint, fragmented, or noisy crack patterns embedded in complex backgrounds [5].

With the progress of deep learning (DL), the field of crack detection has undergone a substantial transformation [6,7]. Convolutional neural networks (CNNs) enable end-to-end learning of hierarchical features directly from raw images, eliminating the need for manual feature engineering [8,9]. Architectures such as U-Net [10], DeepCrack [2], and CrackFormer [11] have achieved promising results. Our prior work, CrackMaster [12], introduced a multi-decoder architecture to enhance crack localization by fusing multi-scale features. However, key challenges remain, particularly in detecting thin and discontinuous cracks in cluttered scenes or under adverse lighting. Where, thin cracks can be defined as cracks with an average width < 10 pixels in 512×512 images, and discontinuous cracks as trajectories with visible gaps ≥ 3 pixels between adjacent crack segments in the binary mask.

* Corresponding author.
E-mail addresses: ammar.okran@urv.cat (A.M. Okran), domenec.puig@urv.cat (D. Puig), saif.khalid@qu.edu.iq, saifkhalidmusluh.al-khalidy@urv.cat (S.K. Musluh), sylvie.chambon@toulouse-inp.fr (S. Chambon), hatem.abdellatif@urv.cat (H.A. Rashwan).

https://doi.org/10.1016/j.autcon.2025.106635
Received 18 July 2025; Received in revised form 23 October 2025; Accepted 27 October 2025
Available online 11 November 2025
0926-5805/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

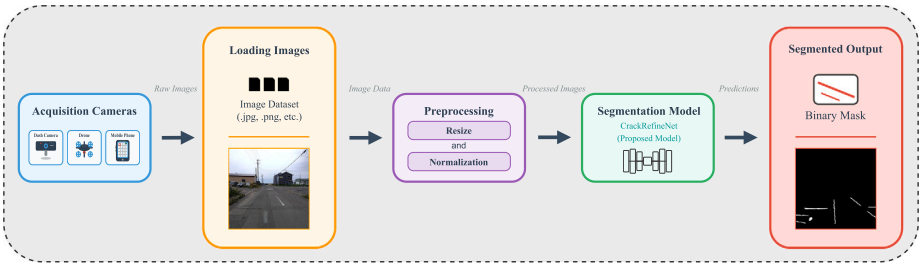


Fig. 1. Overview of the proposed CrackRefineNet segmentation framework.

Additionally, most existing crack segmentation approaches operate on cropped image patches where cracks are centered and occupy a significant portion of the input [2,12–15]. While this setup simplifies the task, it does not reflect the complexity of real-world scenarios, where cracks are often sparse, thin, and embedded in cluttered backgrounds. In practical applications, such as road or bridge inspection, cracks typically occupy only a small fraction (generally less than 5% of the total image area) of high-resolution images, making full-scene segmentation significantly more challenging. It introduces issues such as extreme class imbalance, low signal-to-noise ratio, and greater appearance variability [1,16].

To address these limitations, we propose CrackRefineNet, a multi-stage attention-based deep segmentation framework designed for realistic and complex crack detection scenarios. Unlike many existing approaches, CrackRefineNet is explicitly developed to process full-scene images. Here, full-scene images refer to un-cropped, high-resolution views of entire pavements or road surfaces that include diverse background contexts and irrelevant objects such as vehicles, trees, and people, enabling accurate localization of subtle and fine-grained crack structures without the need for pre-cropped regions of interest. As illustrated in Fig. 1, the overall pipeline includes stages for data acquisition, preprocessing, segmentation, and binary mask generation. The architecture, detailed in Fig. 2, is built upon a ConvNeXt encoder [17] and incorporates three dedicated modules: a context-aware module (CAM) to enhance spatial detail in early stages, an adaptive feature refinement (AFR) module to improve deep semantic features, and a selective feature aggregation (SFA) module to unify multi-level representations. These components work in synergy to enable robust crack segmentation across diverse conditions and acquisition settings. The key contributions of this work are as follows:

- CrackRefineNet is introduced as a segmentation framework designed to address real-world challenges in crack detection, including thin and fragmented cracks, cluttered textures, and varying lighting conditions. The framework combines multi-stage attention mechanisms and feature aggregation strategies to improve segmentation precision and robustness in full scene images.
- Three specialized modules are proposed, each aligned with specific engineering requirements in road crack segmentation:
  - CAM: Addresses varying lighting conditions and complex textures by enhancing shallow encoder features with spatial complexity information (e.g., local standard deviation and mean), enabling more reliable crack boundary detection in varying lighting and textured surfaces.
  - AFR: Targets thin and discontinuous cracks by refining deep semantic features with multi-scale convolutional cues and channel-wise attention, enabling the model to reliably detect crack regions even when they appear fragmented or low-contrast.
  - SFA: Facilitates full-scene crack analysis by aggregating outputs from all refinement stages with learnable weights, selectively integrating local structural details and global context to handle large, high-resolution road images.
- A new Road Crack Full-Scene Dataset (RCFD) of complete outdoor scenes is introduced, derived from RDD2022 [16], containing 1428 full-resolution RGB images captured under diverse environmental conditions and acquisition platforms. This dataset provides a realistic benchmark for evaluating segmentation performance in challenging field conditions.
- Extensive experiments and ablation studies are performed on four datasets: RCFD, Crack500, CFD, and Sylvie, demonstrating that each proposed module contributes meaningfully to overall performance and that CrackRefineNet consistently outperforms state-of-the-art methods in both supervised and zero-shot scenarios.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 details the CrackRefineNet architecture. Section 4 describes the datasets and training setup. Section 5 presents experimental results and analysis. Finally, Section 6 concludes the paper and discusses future directions.

2. Related work

This section reviews recent advances in automated crack detection and segmentation, highlighting the evolution from classical image processing techniques to deep learning-based models. The discussion is structured to cover traditional methods, machine learning approaches, and modern architectures that incorporate attention and multi-scale feature fusion. This contextual overview establishes the foundation for understanding how the proposed CrackRefineNet builds upon and advances existing research.

2.1. Traditional and machine learning approaches

Early crack detection methods relied on classical image processing techniques, such as thresholding, edge detection, and morphological operations [18]. While computationally efficient, these methods were highly sensitive to lighting variations, surface noise, and background textures. Later enhancements introduced region-growing [3], minimal-path tracing [19], and graph-based strategies such as CrackTree [20], improving robustness to some extent. However, their dependency on hand-crafted rules and predefined parameters limited their ability to generalize to complex or irregular crack patterns.

To overcome these constraints, ML-based techniques were introduced. Early ML-based methods combined Support Vector Machines (SVMs) with preprocessing pipelines [21], while more recent approaches leveraged ensemble classifiers such as LightGBM [22] and Random Structured Forests [23]. Transfer learning has also been explored, using CNN-extracted features fed into classifiers [24]. Despite improved performance, most ML methods still depend on externally defined features, limiting adaptability to diverse crack morphologies and cluttered environments.

These limitations have motivated a shift toward end-to-end deep learning models that directly learn discriminative features from raw images, enabling more robust crack detection under real-world conditions.

2.2. Deep learning-based approaches

DL models have become the dominant paradigm for crack segmentation, enabling models to automatically learn hierarchical and task-relevant features directly from raw images. This end-to-end capability eliminates the need for manual feature engineering and allows for greater adaptability.

U-Net [10] was a seminal model in this field, introducing skip connections to preserve spatial resolution during downsampling and upsampling. HED [25], although originally developed for edge detection, became widely used in crack detection due to its multi-scale feature fusion.

Recent DL models such as DeepCrack [2] and CrackFormer [11] improved segmentation of fine and discontinuous crack patterns using hierarchical CNN architectures and transformer modules, respectively. Lightweight designs like LMM [15] and EfficientCrackNet [26] emphasized computational efficiency, while DCMA-Net [27] and Hybrid-Segmentor [28] incorporated specialized attention and hybrid features to handle varying crack appearances. Wang et al. [29] developed MSDCrack, a dual-encoder network that applies multi-stage supervision to improve crack segmentation. The two encoders extract complementary local and global features, while staged supervision helps refine intermediate representations and enhance overall accuracy.

Recent studies have further advanced this line of research. Chen et al. [30] proposed a multi-task network for crack segmentation and quantification in reinforced concrete structures, highlighting the importance of joint optimization for structural assessment. Liang et al. [31] developed a CNN-based model that integrates multi-scale context features with attention to enhance robustness against background noise. Ni et al. [32] introduced RepCrack, which leverages structural re-parameterization for efficiency while preserving accuracy. Zhang et al. [33] combined Mamba modules with frequency-domain feature extraction to better capture tiny cracks. Liu et al. [34] proposed a cross-multi-feature fusion approach for interactive crack segmentation, demonstrating improvements in handling ambiguous textures. Khan et al. [35] presented a lightweight feature fusion framework designed for real-time crack segmentation in resource-constrained environments. Finally, Liu et al. [36] introduced SCSegamba, a Vision-Mamba-based architecture with only 2.8M parameters, showing that structure-aware hybrids can achieve competitive performance with minimal computational cost.

Our prior work, CrackMaster [12], introduced a multi-decoder design that improved feature utilization across scales. However, most deep learning models still struggle in two areas: effectively capturing both local detail and global context in full-scene images, and dealing with thin, low-contrast cracks embedded in complex backgrounds. These issues highlight the need for more targeted architectural components, such as attention modules and multi-scale feature fusion mechanisms, to enhance the ability of the deep segmentation models to isolate crack structures under real-world conditions.

2.3. Attention-based and feature aggregation strategies

Recent advances in semantic segmentation highlight the benefits of attention mechanisms and multi-scale feature aggregation for improving both spatial precision and contextual understanding. SENet [37] introduced channel-wise recalibration to focus on informative features, while FPN [38] demonstrated the effectiveness of fusing hierarchical features to preserve details at different scales. Hybrid attention modules like CBAM [39] integrated spatial and channel attention, helping networks suppress irrelevant regions and emphasize thin or fragmented structures—a crucial requirement for crack segmentation.

Transformer-based models such as CrackFormer [11] and dual-attention frameworks like DCMA-Net [27] further leveraged global interactions and cross-scale feature learning. Recent hybrids such as SCSegamba [36] and the ultra-lightweight Vision-Mamba design by Zhang et al. [33] confirm the growing trend toward combining efficient attention with context-aware feature extraction for high-accuracy, low-cost segmentation. These innovations have significantly improved the segmentation of challenging crack patterns.

Building on these advances, our proposed method, CrackRefineNet, introduces a tightly integrated architecture that addresses the key challenges identified in previous works. The CAM enhances spatial detail in early decoder stages using local complexity cues. The AFR module refines deep semantic features using multi-scale convolution and channel-wise modulation. Finally, the SFA module combines outputs from intermediate stages using learnable weights, enabling the network to integrate detailed and contextual features for improved segmentation accuracy. Together, these components form a unified framework that is well-suited for full-scene crack detection across varied environmental conditions.

To summarize, recent models differ significantly in terms of architectures, fusion/attention strategies, and dataset performance. A consolidated overview is provided in Table 1, which compares leading approaches from the last 3–5 years with CrackRefineNet across these dimensions. Metrics are reported as in the original papers to ensure fair comparison.

3. Methodology

This section details the architecture and components of the proposed CrackRefineNet framework. It begins with the problem formulation, followed by a description of each core module, AFR, CAM, and SFA and concludes with implementation and training details.

3.1. Problem formulation

Let the dataset be denoted as $\mathcal{D} = \{(I_i, M_i)\}_{i=1}^{N}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$ represents an RGB image and $M_i \in \{0,1\}^{H\times W}$ is its corresponding binary segmentation mask, with 1 indicating crack pixels and 0 denoting background.

The objective is to learn a mapping function $f_\theta : \mathbb{R}^{H\times W\times 3} \rightarrow [0,1]^{H\times W}$, parameterized by θ, which predicts a dense crack probability map $\hat{M}_i = f_\theta(I_i)$. The final binary mask is obtained by thresholding $\hat{M}_i$ at a fixed value (e.g., 0.5).

In this work, we propose a DL framework that enhances the segmentation capability through: (1) CAM: To better preserve spatial detail and enrich local representations. (2) AFR: To deep semantic layers that integrate multi-scale spatial cues and channel-wise attention to highlight crack-relevant features while suppressing noise. (3) SFA: A lightweight fusion strategy that selectively weights and aggregates multi-level features from different stages, enabling the final prediction to benefit from both detailed shallow cues and high-level semantic understanding.

The network is trained in a fully supervised setting using the provided pixel-level masks and optimized using a hybrid loss that balances classification and overlap quality.

3.2. Model components

The architecture of CrackRefineNet is designed to handle the subtle, thin, and fragmented nature of cracks in complex scenes. Fig. 2 provides an overview of the proposed CrackRefineNet architecture. The network follows a multi-stage design, beginning with a ConvNeXt encoder that extracts hierarchical features from the input image. These features are progressively refined through a combination of the AFR modules, which enhance deeper semantic representations, and CAM, which focuses on preserving fine spatial details. The decoder reconstructs high-resolution segmentation maps by integrating information from multiple encoder stages. Finally, the SFA module fuses outputs from intermediate stages using learnable weights to generate a unified and robust crack prediction mask. This architecture is designed to handle the subtle, thin, and fragmented nature of cracks in complex scenes.

Table 1
Comparison of recent crack segmentation methods.

Method	Architecture type	Fusion/Attention strategy	Performance
CrackFormer II [11]	Transformer–CNN hybrid	Self-attention across scales	mIoU 89.6% (CrackTree260); mIoU 89.5% (CrackLS315); mIoU 89.7% (Stone331); mIoU 87.9% (DeepCrack537).
LMM [15]	Lightweight CNN	Multi-scale lightweight fusion	F1 74.9%, mIoU 74.8% (Crack500); F1 86.4%, mIoU 86.4% (DeepCrack537); F1 52.7%, mIoU 65.9% (GAPS384); F1 70.9%, mIoU 76.7% (AigleRN-TRIMM).
EfficientCrackNet [26]	Efficient CNN backbone	Spatial-channel fusion	F1 79.1%, mIoU 81.3% (Crack500); F1 85.9%, mIoU 87.1% (DeepCrack537); F1 66.4%, mIoU 71.9% (GAPs384).
DCMA-Net [27]	CNN backbone with dual-attention	Channel + spatial dual-attention	F1 73.41 ± 0.18 (DeepCrack537); F1 75.64 ± 0.15 (Crack500); F1 81.74 ± 0.26 (CrackTree).
MSDCrack [29]	Dual-encoder CNN–Transformer	Multi-stage supervision + dual feature fusion	F1 78.5%, IoU 64.5% (DeepCrack); F1 72.7%, IoU 57.11% (Crack500).
Hybrid-Segmentor [28]	CNN–Transformer hybrid	Cross-scale hybrid fusion	F1 0.774 ± 0.0036 (Refined Dataset).
RepCrack [32]	CNN with re-param blocks	Structural re-param fusion	F1 78.1% (CFD); F1 74.7% (GAPs384); F1 70.5% (CrackTree200).
Vision-Mamba [33]	Vision-Mamba backbone	Frequency-domain + efficient attention	F1 59.1% (BJN260); F1 86.0% (DeepCrack537).
SCSegamba [36]	Vision-Mamba hybrid	Structure-aware attention fusion	F1 75.5%, mIoU 77.8% (Crack500); F1 91.1%, mIoU 90.2% (DeepCrack537); F1 76.8%, mIoU 80.9% (CrackMap); F1 83.9%, mIoU 84.8% (TUT).
CrackMaster [12]	Multi-decoder CNN	Parallel decoders with cross-fusion	Dice 92.5%, mIoU 88.3% (RCD); Dice 93.5%, mIoU 93.5% (DeepCrack537); Dice 86.9%, mIoU 87.6% (YCD).
CrackRefineNet (Ours)	CNN encoder–decoder + modules	AFR + CAM + SFA (learnable fusion)	F1 79.1%, mIoU 81.9% (RCFD); F1 76.6%, mIoU 79.6% (Crack500); F1 70.6%, mIoU 76.8% (CFD).

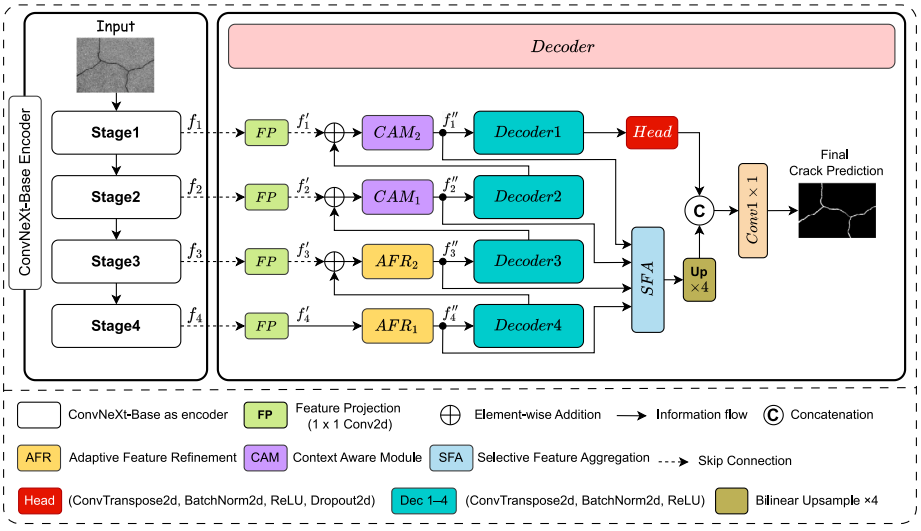


Fig. 2. Overview of the proposed CrackRefineNet architecture showing encoder, AFR, CAM, decoder, and SFA.

3.2.1. Encoder: ConvNeXt-Base

The encoder extracts hierarchical representations from the input image, capturing both low-level textures and high-level semantic features required for accurate crack segmentation.

ConvNeXt [17] is a modern convolutional architecture that combines the strengths of CNNs with transformer-inspired design patterns. It provides a favorable trade-off between accuracy and efficiency and has shown strong performance on dense prediction tasks, making it suitable for thin and discontinuous crack structures.

We adopt ConvNeXt-Base as the encoder backbone. The input image $I \in \mathbb{R}^{H\times W\times 3}$ is processed through four sequential stages, each producing a feature map:

$$f_l = \text{Stage}_l(I_l), \quad l = 1, 2, 3, 4, \tag{1}$$

where $I_1 = I$, and $I_l = f_{l-1}$ for $l > 1$. Each $\text{Stage}_l(\cdot)$ consists of convolutional downsampling followed by a series of ConvNeXt blocks.

The output feature map $f_l \in \mathbb{R}^{H_l\times W_l\times C_l}$ is projected into a unified 256-channel space using a $\text{Conv}_{1\times1}$:

$$f'_l = \text{Conv}_{1\times1}(f_l), \quad f'_l \in \mathbb{R}^{H_l\times W_l\times 256}, \tag{2}$$

where $\text{Conv}_{1\times1}$ denotes a convolution with kernel size 1×1 and fixed 256 output channels.

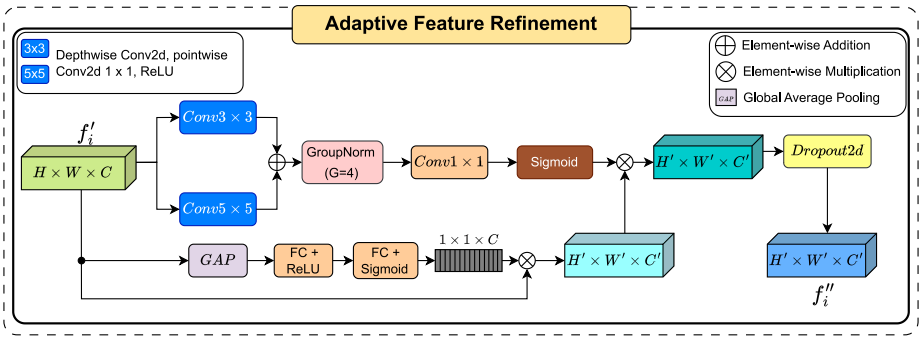


Fig. 3. Architecture of the Adaptive Feature Refinement Module.

The set of projected feature maps $\{f_1', f_2', f_3', f_4'\}$ serves as input to the refinement modules and decoder, as illustrated in Fig. 2.

For clarity, stages 1–2 are referred to as shallow stages (high spatial resolution, low-level features such as edges and textures), whereas stages 3–4 are deep stages (low spatial resolution, high-level semantic representations).

3.2.2. Adaptive feature refinement (AFR)

The AFR module refines deep features to improve detection of thin, fragmented, and low-contrast cracks by combining multi-scale spatial context with channel-wise modulation. Fig. 3 illustrates its structure, which consists of four components: spatial refinement, group normalization, channel attention, and spatial gating.

Given an input feature map $f_i' \in \mathbb{R}^{C\times H\times W}$, AFR applies two parallel depthwise convolutional branches with kernel sizes 3×3 and 5×5. Each is followed by a pointwise $\text{Conv}_{1\times1}$ and a ReLU activation:

$$S_3 = \text{ReLU}(\text{Conv}^{(3)}_{1\times1}(\text{Conv}_{3\times3}(f_i'))), \quad S_5 = \text{ReLU}(\text{Conv}^{(5)}_{1\times1}(\text{Conv}_{5\times5}(f_i'))). \tag{3}$$

The outputs are fused by summation:

$$S = S_3 + S_5. \tag{4}$$

Group Normalization [40] with four groups stabilizes the feature distribution:

$$\tilde{S} = \text{GroupNorm}(S). \tag{5}$$

To emphasize informative channels, channel attention is applied using the squeeze-and-excitation strategy [37]. Global average pooling generates a channel descriptor:

$$z_c = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{c,i,j}, \quad \forall c \in \{1, \ldots, C\}, \tag{6}$$

resulting in $z \in \mathbb{R}^C$. This vector passes through two fully connected layers with ReLU and sigmoid activations:

$$\alpha = \sigma\left(FC_2(\text{ReLU}(FC_1(z)))\right), \quad \alpha \in \mathbb{R}^C, \tag{7}$$

where $FC_1 \in \mathbb{R}^{\frac{C}{16}\times C}$, $FC_2 \in \mathbb{R}^{C\times\frac{C}{16}}$, and σ is the sigmoid function.

A spatial gating map is generated by applying a $\text{Conv}_{1\times1}$ to the normalized features:

$$g = \sigma(\text{Conv}_{1\times1}(\tilde{S})), \quad g \in \mathbb{R}^{C\times H\times W}. \tag{8}$$

The refined output is obtained by applying both channel attention and spatial gating to the input feature map:

$$f''(f_{\text{refined}}) = f_i' \odot g \odot \alpha, \tag{9}$$

where $\odot$ denotes element-wise multiplication. Here, α provides per-channel scaling factors that are applied uniformly across all spatial locations, while g supplies spatially varying weights. A dropout layer with probability $p = 0.3$ is applied for regularization. No residual connection is used in this configuration.

In CrackRefineNet, two AFR modules are used: one refines the encoder feature f_4' before Decoder 4, and the other refines the sum of Decoder 4's output and encoder feature f_3', guiding Decoder 3.

3.2.3. Context-aware module (CAM)

The CAM module enriches shallow feature maps by combining spatial and channel attention, helping the network focus on relevant structures under uneven illumination and cluttered textures. This refinement is particularly valuable for crack segmentation, where fine localization depends on edge- and texture-level cues in the early decoder stages.

As shown in Fig. 4, CAM integrates two branches: (1) spatial attention derived from local complexity, and (2) channel attention using squeeze-and-excitation.

Given an input feature map $f' \in \mathbb{R}^{C\times H\times W}$, the spatial attention branch starts by computing two statistical descriptors: a local standard deviation map and a local mean map across the channel dimension:

$$M_{\text{std}} = \text{Std}(f'), \quad M_{\text{mean}} = \text{Mean}(f'), \tag{10}$$

where $M_{\text{std}}, M_{\text{mean}} \in \mathbb{R}^{1\times H\times W}$. These are concatenated to form a 2-channel complexity map:

$$M_{\text{comp}} = \text{Concat}(M_{\text{std}}, M_{\text{mean}}) \in \mathbb{R}^{2\times H\times W}. \tag{11}$$

This complexity map is normalized channel-wise to stabilize the distribution:

$$\hat{M}_{\text{comp}} = \frac{M_{\text{comp}} - \mu}{\sigma + \epsilon}, \quad where \mu = \text{Mean}(M_{\text{comp}}),\ \sigma = \text{Std}(M_{\text{comp}}),\ \epsilon = 1e-6. \tag{12}$$

Next, spatial attention is computed in three steps. First, the normalized complexity map $\hat{M}_{\text{comp}}$ is passed through a 3×3 convolution followed by a 1×1 convolution and a ReLU activation:

$$f = \text{ReLU}(\text{Conv}_{1\times1}(\text{Conv}_{3\times3}(\hat{M}_{\text{comp}}))). \tag{13}$$

Then, parallel convolutions with larger receptive fields are applied:

$$f_1 = \text{Conv}_{5\times5}(f), \quad f_2 = \text{Conv}_{7\times7}(f). \tag{14}$$

The outputs, f_1 and f_2 are summed and passed through a sigmoid to produce the spatial attention map:

$$A_{\text{spatial}} = \sigma(f_1 + f_2). \tag{15}$$

For channel attention, global average pooling is applied:

$$z = \text{GAP}(f') \in \mathbb{R}^{C\times1\times1}, \tag{16}$$

and the descriptor passes through a two-layer bottleneck with ReLU and sigmoid activations:

$$\alpha = \sigma\left(\text{FC}_2(\text{ReLU}(\text{FC}_1(z)))\right), \quad \alpha \in \mathbb{R}^{C\times1\times1}. \tag{17}$$

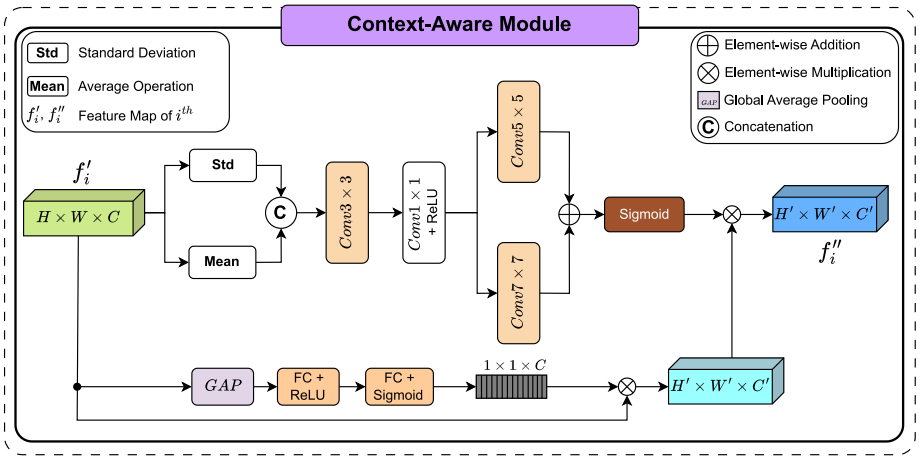


Fig. 4. Architecture of the Context Aware Module.

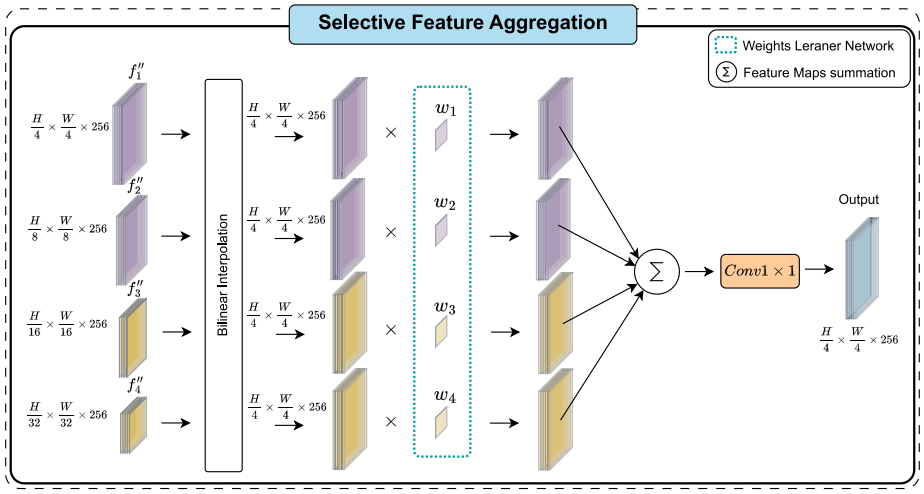


Fig. 5. Architecture of the Selective Feature Aggregation Module.

Finally, the refined output is obtained by applying both attention maps to the input:

$$f'' = f' \odot A_{\text{spatial}} \odot \alpha, \tag{18}$$

where $\odot$ denotes element-wise multiplication with broadcasting. No residual connection is used in this configuration.

In CrackRefineNet, two CAM modules are used: one processes the sum of Decoder 3's output and encoder feature f_2', and the other processes the sum of Decoder 2's output and encoder feature f_1'. These refinements enhance shallow decoder features and preserve fine crack boundaries.

3.2.4. Decoder, fusion, and selective feature aggregation

The decoder reconstructs high-resolution feature maps by progressively upsampling and refining encoded representations. It integrates deep semantic cues with shallow structural details, enabling precise crack boundary localization.

CrackRefineNet includes four decoder layers, each paired with a refinement module. Decoder 4 processes the output of $AFR_1(f_4')$. Its result is added element-wise to f_3', then refined by AFR_2 before entering Decoder 3. Decoder 2 receives the output of CAM_1, applied to the sum of Decoder 3 and f_2'. Finally, Decoder 1 processes the output of CAM_2, applied on top of Decoder 2 and f_1'.

Each decoder layer consists of a transposed convolution followed by batch normalization and ReLU activation, performing resolution upsampling and non-linear transformation. The output of Decoder 1 undergoes a second upsampling and dropout, yielding the final decoded representation F_{dec}.

To fuse complementary cues from AFR and CAM, a Selective Feature Aggregation (SFA) module is employed (Fig. 5). The outputs of the two AFR modules are denoted as f_3'', f_4'', and those of the two CAM modules as f_1'', f_2''. These four feature maps are resized to the highest spatial resolution by bilinear interpolation:

$$f_i'' = \text{Interpolate}(f_i'', \text{size} = H \times W), \quad i \in \{1, 2, 3, 4\}. \tag{19}$$

Each refined feature is weighted by a learnable scalar $w_i \in \mathbb{R}$, and the weighted maps are summed:

$$F_{\text{agg}} = \sum_{i=1}^{4} w_i \cdot f_i''. \tag{20}$$

A $\text{Conv}_{1\times1}$ is then applied to adjust channel representation:

$$F_{\text{agg}}^{\text{final}} = \text{Conv}_{1\times1}(F_{\text{agg}}). \tag{21}$$

To obtain the final segmentation output, $F_{\text{agg}}^{\text{final}}$ is concatenated with the output of the last decoder layer F_{dec}, and passed through a lightweight segmentation head consisting of a 1×1 convolution:

$$F_{\text{out}} = \text{Concat}(F_{\text{dec}}, F_{\text{agg}}^{\text{final}}), \quad \hat{M} = \text{Conv}_{1\times1}(F_{\text{out}}). \tag{22}$$

The decoder and fusion design ensures that both deep semantic cues and early structural information are retained and harmonized in the final prediction, improving crack boundary localization and robustness.

Table 2
Details of the crack segmentation datasets used in this study.

Dataset	Image resolution	Training set	Validation set	Test set
RCFD (in-house)	512 × 512	973	170	285
Crack500 [42]	512 × 512	1896	348	1124
CFD [4]	512 × 512	101	–	18
Sylvie [19]	512 × 512	52	–	13
TRCD (Pilot)	512 × 512	–	–	310

3.3. Training

CrackRefineNet is trained in a supervised fashion using paired RGB images and binary crack masks. The model is optimized to produce accurate segmentation outputs by minimizing a composite loss that balances both local pixel accuracy and structural consistency.

To select the most appropriate segmentation loss, we conducted a thorough evaluation of several candidates. These included standard functions such as Binary Cross-Entropy (BCE), Cross-Entropy (CE), Dice Loss, Focal Loss, and Tversky-based formulations, as well as composite versions like BCE + Dice and BCE + IoU.

Among them, the most effective was the combination of BCE and Dice losses. This choice is theoretically motivated by the characteristics of crack segmentation: the foreground (cracks) typically occupies a very small fraction of the pixels, leading to extreme class imbalance, and the targets are often thin, elongated, and fragmented. BCE ensures pixel-wise discrimination by penalizing misclassifications across all pixels, but may be biased toward the majority background. Dice loss, in contrast, directly optimizes region-level overlap, enforcing continuity and penalizing false negatives in thin structures [41]. By combining them, BCE stabilizes optimization under imbalance while Dice emphasizes structural alignment, making the hybrid formulation particularly suited to crack segmentation [32].

The composite loss is defined as:

$$L_{\text{Total}} = 0.5 \cdot L_{\text{BCE}} + 0.5 \cdot L_{\text{Dice}}, \tag{23}$$

where the Binary Cross-Entropy loss is:

$$L_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right], \tag{24}$$

with y_i denoting the ground truth and $\hat{y}_i$ the estimated output.

In turn, the Dice loss is:

$$L_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i + \epsilon}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + \epsilon}, \tag{25}$$

where $\epsilon = 10^{-6}$ ensures numerical stability.

This formulation simultaneously encourages correct pixel classification and global structural coherence, helping the model detect cracks even when they are faint or discontinuous.

The model is optimized using the AdamW optimizer. Further implementation details, including augmentation strategies and learning rate scheduling, are described in the Experimental Setup section.

Through experimentation, we found that while CrackRefineNet performed reliably across all tested loss functions, the combined BCE + Dice loss consistently yielded the best segmentation accuracy. Therefore, all reported experiments adopt it as the default training objective.

4. Datasets and experimental setup

This section presents the datasets used to evaluate CrackRefineNet, the data augmentation strategies applied during training, the implementation settings, and the metrics adopted for performance evaluation.

4.1. Datasets

We evaluate CrackRefineNet using five diverse datasets: RCFD, Crack500, CFD, Sylvie, and TRCD. These datasets vary in acquisition settings, crack morphologies, background complexity, and image quality. This diversity ensures a comprehensive and realistic evaluation of segmentation robustness and generalizability. All images are resized to 512 × 512 pixels to ensure consistency during training and testing. Fig. 6 shows representative samples from each dataset, with each column displaying two samples that highlight the variability in surface texture, lighting, and crack structure. Dataset statistics are summarized in Table 2.

In-house Road Crack Full-Scene (RCFD) Dataset: Our RCFD dataset is curated from the RDD2022 [16] benchmark, which contains road images collected across China, India, the Czech Republic, and Japan using drones, motorcycles, and dashboard cameras. While our earlier dataset [7] was derived from RDD2020 [43] and annotated with four crack categories (Longitudinal, Transverse, Alligator, and Pothole), the RCFD dataset consolidates these into a binary setting (crack vs. background) to better suit the segmentation task. Following the same annotation protocol described in our prior work, cracks were manually segmented at the pixel level. Bounding boxes were used as auxiliary guides to ensure complete coverage, and all annotations underwent quality control by expert reviewers to guarantee precision. Compared to our previous release of 506 images, RCFD expands to 1,428 full-scene RGB images with varied acquisition devices and environments, providing richer diversity and stronger evaluation benchmarks for crack segmentation. The dataset is split into 973 training, 170 validation, and 285 test images.

Crack500 [42]: Crack500 is a large-scale benchmark dataset comprising 3,368 RGB pavement images at a resolution of 640 × 360 pixels. Each image is annotated with a detailed binary mask that precisely outlines crack regions. The dataset is divided into 1,896 training images, 348 for validation, and 1,124 for testing. With its wide range of surface textures, lighting conditions, and crack patterns, Crack500 serves as a strong benchmark for evaluating the robustness and generalization capability of crack segmentation models.

CFD [4]: The CrackForest Dataset (CFD) comprises 119 road surface images with a resolution of 480 × 320 pixels. Each image is annotated with precise pixel-level crack labels. The dataset is split into 101 training images and 18 test images. Despite its relatively small size, CFD presents a variety of pavement textures and crack appearances, making it a useful benchmark for evaluating the effectiveness of crack segmentation models, especially under limited-data conditions.

Sylvie Dataset [19]: This dataset combines pavement images from five inspection systems (AIGLE_RN, ESAR, LCMS, LRIS, and TEMPEST2), offering diverse imaging conditions. We use a subset of 65 RGB images with pixel-level crack annotations, split into 52 for training and 13 for testing. The dataset presents significant challenges due to its thin, fragmented, and low-contrast cracks, making it well-suited for testing model robustness under real-world and low-supervision scenarios.

Tarragona Road Crack Dataset (TRCD): TRCD was collected in Tarragona, Spain, and consists of 310 images captured using both smartphone and GoPro cameras. The dataset includes a variety of road types, crack patterns, lighting conditions (day/night), and weather



Fig. 6. Sample crack images from the five datasets: RCFD, Crack500, CFD, Sylvie, and TRCD.

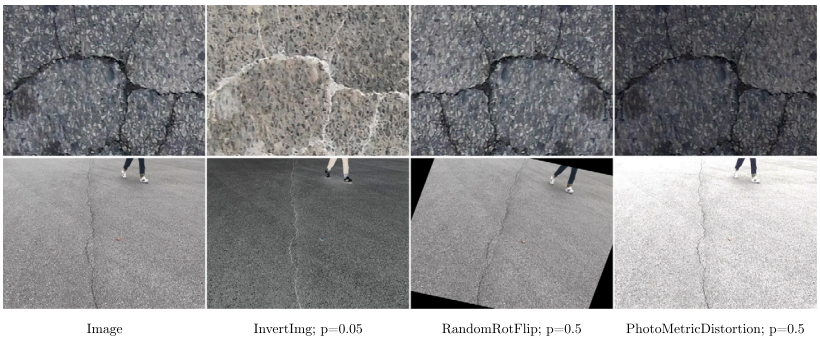


Fig. 7. Examples of data augmentation transformations applied during training.

scenarios (sunny/cloudy), making it well-suited for real-world evaluation. Its diversity in resolution and perspective helps benchmark model performance under practical, uncontrolled conditions.

4.2. Data augmentation

To increase model robustness and reduce overfitting, we employ several augmentation strategies from MMSegmentation [44]. These transformations simulate realistic visual distortions and increase diversity in the training data:

Color inversion: With a probability of 0.05, the image colors were inverted using the `InvertImg` transformation. This encourages the model to learn invariant features under significant pixel-level alterations.

Random rotations and flipping: The `RandomRotFlip` module was used to randomly rotate images by 90°, 180°, or 270°, and apply horizontal or vertical flips, with a probability of 0.5.

Photometric distortion: This included a sequence of visual perturbations such as random brightness adjustment, contrast changes, conversion from BGR to HSV color space, random saturation and hue shifts, and conversion back to BGR. These were also applied with a 0.5 probability.

Fig. 7 shows examples of the augmented images. These transformations improve generalization under variable lighting and imaging conditions.

4.3. Implementation details

All models were implemented using the MMSegmentation framework [44]. Experiments were conducted on a system running Windows 10 Pro, equipped with an Intel Core i7-9700K CPU (64-bit, 3.60 GHz), 64 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB. Training was performed using the AdamW optimizer [45], with an initial learning rate of 1×10^{-4} for all datasets. To ensure stable convergence, we employed a two-stage learning rate schedule: a `LinearLR` warm-up phase over the first 3,000 iterations with a starting factor of 1×10^{-6}, followed by a `PolyLR` policy from iteration 3000 to 20,000, using a power of 1.0 and an eta_min of 0.0. For training on the RCFD, Crack500, and CFD datasets, we used a batch size of 4, determined based on the GPU memory constraints and the input resolution of 512 × 512 pixels. Each training session was conducted over 20,000 iterations.

4.4. Evaluation metrics

We evaluate the performance of CrackRefineNet using five standard metrics: Intersection over Union (IoU), Precision, Recall, F1-score, and mIoU. These metrics offer a balanced assessment of segmentation quality. IoU and mIoU measure the overlap between predicted and ground truth regions, Precision and Recall capture correctness and completeness, while F1-score provides a harmonic mean of Precision and Recall, particularly useful in handling class imbalance.

5. Experimental results and discussion

This section presents an in-depth evaluation of the proposed CrackRefineNet framework, beginning with an ablation study to understand the contributions of each module. Performance is then analyzed across four datasets: RCFD, Crack500, CFD, and Sylvie, using the evaluation metrics illustrated in the previous section: IoU, mIoU, F1-score, Precision, and Recall.

5.1. Ablation study

To validate the effectiveness of each component in the proposed framework, ablation studies were performed on three datasets: RCFD, Crack500, and CFD. These experiments quantify the contribution of each module (AFR, CAM, and SFA) to segmentation performance. A baseline consisting of only an encoder and decoder was established, followed by incremental addition of each module to analyze its impact on key metrics: IoU, Precision, Recall, F1-score, and mIoU. The full model includes the concatenation of decoder and SFA outputs.

Table 3 summarizes the ablation study on the RCFD dataset by reporting the relative performance gains introduced by each module.

Table 3
Ablation study results on the RCFD dataset.

Model variant	IoU (%)	Precision (%)	Recall (%)	F1-score (%)	mIoU‘(%)
Baseline (Encoder + Decoder)	58.0	70.10	74.90	72.40	75.60
+ AFR	61.50	72.90	77.40	75.20	78.20
+ CAM	63.20	75.60	78.60	77.10	80.0
+ SFA (decoder bypassed)	63.90	77.0	78.60	77.80	80.90
+ SFA Fusion ‖ Decoder (full model)	**65.41**	**79.20**	**78.98**	**79.09**	**81.97**

Table 4
Ablation study results on the Crack500 dataset.

Model variant	IoU (%)	Precision (%)	Recall (%)	F1-score (%)	mIoU (%)
Baseline (Encoder + Decoder)	56.94	70.86	72.42	71.63	76.27
+ AFR	59.13	72.08	74.60	73.32	77.55
+ CAM	60.45	73.51	75.12	74.30	78.61
+ SFA (decoder bypassed)	61.72	73.98	76.36	75.15	79.33
+ SFA Fusion ‖ Decoder (full model)	**62.02**	**74.57**	**78.65**	**76.57**	**79.56**

Table 5
Ablation study results on the CFD dataset.

Model variant	IoU (%)	Precision (%)	Recall (%)	F1-score (%)	mIoU (%)
Baseline (Encoder + Decoder)	51.13	60.44	76.26	67.46	74.69
+ AFR	52.48	62.30	77.16	69.10	75.66
+ CAM	53.06	62.80	78.39	69.76	76.05
+ SFA (decoder bypassed)	54.09	63.45	78.49	70.13	76.57
+ SFA Fusion ‖ Decoder (full model)	**54.57**	**63.80**	**79.05**	**70.61**	**76.81**

Adding the AFR module to the baseline results in a 3.5% increase in IoU and a 2.8% improvement in F1-score, highlighting its contribution to refining high-level semantic features and recovering more complete crack regions. When CAM is introduced, IoU rises by an additional 1.7%, and the F1-score improves by 1.9%, demonstrating its role in enhancing spatial detail and boundary sensitivity. Replacing the decoder with the SFA module yields further improvements, particularly in Precision (+1.4%), suggesting its strength in suppressing false positives. The full model, which combines the outputs of the decoder and SFA in parallel, achieves the most significant overall gains: a 7.4% improvement in IoU and a 6.7% increase in F1-score over the baseline. These cumulative improvements confirm that each module introduces a meaningful enhancement, and their integration leads to a substantial boost in segmentation performance.

As shown in Table 4, the ablation study on the Crack500 dataset illustrates the incremental benefits of each proposed module. Incorporating AFR into the baseline yields a 2.2% relative improvement in IoU and a 1.7% boost in F1-score, validating its role in enhancing deep semantic representations. The addition of CAM provides a further 1.3% increase in IoU and a 1.0% gain in F1-score, demonstrating the advantage of incorporating contextual spatial cues. Replacing the decoder with SFA leads to an overall 3.5% improvement in F1-score and 4.8% in IoU compared to the baseline, emphasizing the effectiveness of selective multi-level aggregation. The full model, which combines decoder and SFA outputs in parallel, achieves the most notable gains: a 5.1% increase in IoU and a 4.9% rise in F1-score over the baseline. These consistent improvements across all metrics underscore the complementary contributions of each module and the robustness of the integrated architecture in handling complex crack segmentation scenarios.

Table 5 summarizes the ablation study results on the CFD dataset, highlighting the contribution of each component in CrackRefineNet. The addition of AFR to the baseline architecture led to a relative improvement of 1.4% in IoU and 1.6% in F1-score, confirming its utility in enhancing semantic representations and improving crack pixel recovery. Introducing the CAM module resulted in a further 0.6% IoU gain and 0.7% F1-score increase, reflecting the value of context-aware modeling in refining spatial boundaries. Replacing the decoder with the SFA module brought a 2.7% F1-score improvement over the baseline, with a total 3.0% rise in IoU, highlighting the importance of multi-level feature fusion. Finally, the full model that combines SFA and decoder outputs achieved the highest performance across all metrics, with a 3.2% improvement in F1-score and a 3.4% rise in IoU compared to the baseline. These results underscore the complementary contributions of AFR, CAM, and SFA, and demonstrate the robustness of the proposed design even under limited-data conditions present in the CFD dataset.

As illustrated in Fig. 8, the ablation on the RCFD dataset demonstrates the incremental and complementary gains of each architectural enhancement within CrackRefineNet. The plot compares five configurations: Baseline, +AFR, +CAM, +SFA (decoder bypass), and the Full model across five metrics (IoU, F1-score, mIoU, Precision, and Recall) that correspond to Table 3. The AFR module improves crack region recall by refining semantic features; CAM reinforces spatial awareness through contextual encoding; and the SFA module boosts segmentation accuracy by aggregating multi-level features. Most notably, the full model that concatenates SFA with the decoder output consistently outperforms all intermediate variants, achieving the highest scores across all metrics. These results confirm the effectiveness of each module and highlight that CrackRefineNet is modular and effective for crack segmentation.

5.2. Comparative experiments to the state of the art

To comprehensively assess CrackRefineNet performance, comparative experiments were conducted on four benchmark datasets: RCFD, Crack500, CFD, and Sylvie. The model was evaluated against a broad spectrum of state-of-the-art segmentation approaches, including classical CNN-based networks (U-Net [10], DeepCrack [2]), lightweight architectures (MobileNetV3 [46], LMM [15]), and transformer-based models (Swin Transformer [47], MobileViT [48], MSDCrack [29], CrackFormer [11]). Results are summarized in Tables 6, 7, 8, and 9.

Comparative Experiments on RCFD Dataset: As shown in Table 6, CrackRefineNet achieves the best overall performance on the RCFD dataset. Compared to the classic U-Net, it significantly improves IoU by +16.6% and F1-score by +13.5%, confirming its superiority in localizing and recovering thin crack patterns. Against the widely used DeepCrack model, the gains are even more pronounced: +22.2% in IoU, +18.7% in F1-score, and +12.2% in mIoU. While CrackMaster

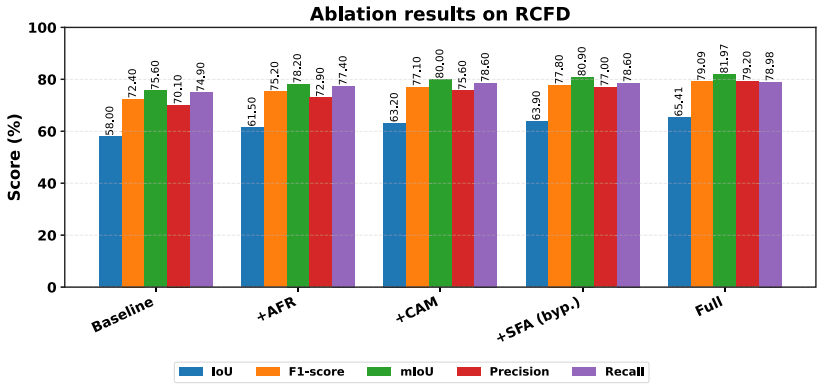


Fig. 8. Bar plot summarizing the incremental contributions of CAM, AFR, and SFA on the RCFD dataset.

Table 6
Comparative results across various methods on RCFD dataset. The **red bold** text indicates the best results, while the **blue bold** text represents the second-best results.

Model	Type	IoU (%)	Precision (%)	Recall (%)	F1-score (%)	mIoU (%)
U-Net [10]		48.82	58.42	74.82	65.61	73.03
HED [25]		24.86	56.66	30.69	39.82	60.82
ERFNet [52]		37.64	41.16	***81.47***	54.69	66.44
DeepCrack [2]	CNN	43.23	48.78	79.16	60.36	69.79
MobileNetv3 [46]		51.69	71.57	65.04	68.15	74.78
LMM [15]		28.05	29.92	***81.81***	43.81	60.33
CrackMaster [12]		***63.53***	***79.37***	76.09	***77.7***	***81.0***
SwinT [47]		53.94	73.84	66.68	70.08	75.97
MobileViT [48]		54.44	73.69	67.58	70.5	76.23
Crackformer II [11]		33.42	36.0	82.37	50.1	63.82
CT-CrackSeg [53]	Transformer	29.58	43.06	48.59	45.66	62.78
EfficientCrackNet [26]		35.47	39.42	77.93	52.36	65.24
MSDCrack [29]		33.94	43.68	60.35	50.68	64.92
Hybrid-Segmentor [28]		47.42	72.57	57.78	64.34	72.59
Proposed	CNN	***65.41***	***79.2***	78.98	***79.09***	***81.97***

Table 7
Comparative results across various methods on Crack500 dataset. The **red bold** text indicates the best results, while the **blue bold** text represents the second-best results.

Model	Type	IoU (%)	Precision (%)	Recall (%)	F1-score (%)	mIoU (%)
U-Net [10]		57.8	***74.77***	71.81	73.26	77.33
HED [25]		54.69	70.99	70.42	70.71	75.60
ERFNet [52]		59.03	73.41	75.08	74.24	77.95
DeepCrack [2]	CNN	58.62	72.34	75.55	73.91	77.71
MobileNetv3 [46]		57.26	73.99	71.69	72.82	77.03
LMM [15]		56.42	73.08	71.22	72.14	76.56
CrackMaster [12]		***60.96***	73.77	77.83	***75.74***	***78.98***
SwinT [47]		57.16	73.24	72.26	72.75	76.96
MobileViT [48]		54.75	72.4	69.2	70.76	75.67
Crackformer II [11]		57.01	71.61	73.66	72.62	76.84
CT-CrackSeg [53]	Transformer	59.5	70.48	***79.25***	74.61	78.13
EfficientCrackNet [26]		53.93	***75.71***	65.22	70.07	75.31
MSDCrack [29]		57.11	69.37	76.38	72.7	76.88
Hybrid-Segmentor [28]		58.16	72.22	74.92	73.55	77.51
Proposed	CNN	***62.02***	74.57	***78.65***	***76.57***	***79.56***

shows competitive Precision (only 0.17% higher), it underperforms in Recall and F1-score, resulting in 1.4% lower overall segmentation accuracy. Transformer-based models such as MobileViT and SwinT perform reasonably well but still lag behind by over 10% in IoU and F1 compared to CrackRefineNet, likely due to difficulty in modeling low-texture structures without strong local inductive biases. These results validate the benefits of the modular architecture and its ability to generalize to complex real-world scenes.

Comparative Experiments on Crack500 Dataset: As illustrated in Table 7, CrackRefineNet outperforms all competing models across most metrics. Compared to CrackMaster, the closest competitor, CrackRefineNet delivers a +0.8% gain in F1-score and a +0.6% improvement in mIoU, indicating stronger consistency and generalization. Relative to classical CNNs like U-Net and DeepCrack, it boosts IoU by over +3.4% and F1-score by more than +2.6%, confirming superior ability to capture crack boundaries and fine structures. While CT-CrackSeg achieves the highest Recall, it suffers from low Precision, resulting in a significantly lower F1-score. Transformer-based methods such as MobileViT and SwinT also fall short by over 2.6% in F1 and mIoU, highlighting that lightweight convolutional designs with adaptive refinement modules are more effective for this domain. These results further validate the strength of the modular design in balancing segmentation sensitivity and precision.

Comparative Experiments on CFD Dataset: Table 8 reports results on the CFD dataset, known for illumination variation and complex backgrounds. CrackRefineNet consistently outperforms all baselines,

Table 8
Comparative results across various methods on CFD dataset. The **red bold** text indicates the best results, while the **blue bold** text represents the second-best results.

Model	Type	IoU (%)	Precision (%)	Recall (%)	F1-score (%)	mIoU (%)
U-Net [10]		52.46	62.99	75.84	68.82	75.73
HED [25]		51.74	64.7	72.1	68.2	75.38
ERFNet [52]		49.73	60.61	73.48	66.43	74.32
DeepCrack [2]	CNN	52.26	62.05	76.81	68.65	75.62
MobileNetv3 [46]		50.92	60.01	77.08	67.48	74.92
LMM [15]		***53.65***	***65.05***	75.38	***69.83***	***76.35***
CrackMaster [12]		51.28	58.29	***81.0***	67.79	75.08
SwinT [47]		52.76	62.84	76.68	69.08	75.88
MobileViT [48]		53.26	***70.07***	68.95	69.5	76.19
Crackformer II [11]		50.43	57.26	80.88	67.05	74.63
CT-CrackSeg [53]	Transformer	49.85	57.87	78.23	66.53	74.35
EfficientCrackNet [26]		50.17	56.84	***81.05***	66.82	74.50
MSDCrack [29]		49.01	55.66	80.41	65.78	73.90
Hybrid-Segmentor [28]		47.04	62.21	65.86	63.98	72.98
Proposed	CNN	***54.57***	63.8	79.05	***70.61***	***76.81***

Table 9
Zero-shot evaluation results on Sylvie dataset. All models were trained solely on the CFD dataset without fine-tuning. **Red bold** indicates the best performance, while **blue bold** highlights the second-best.

Model	Type	IoU (%)	Precision (%)	Recall (%)	F1-score (%)	mIoU (%)
U-Net [10]		17.99	20.24	61.81	30.49	58.14
HED [25]		21.86	24.8	64.86	35.88	60.23
ERFNet [52]		19.79	***31.53***	34.71	33.05	59.47
DeepCrack [2]	CNN	9.92	26.94	13.58	18.05	54.59
MobileNetv3 [46]		15.0	23.23	29.72	26.08	56.99
LMM [15]		***24.6***	29.2	60.97	***39.48***	***61.73***
CrackMaster [12]		20.96	22.07	***80.56***	34.65	59.56
SwinT [47]		18.44	29.57	32.88	31.14	58.78
MobileViT [48]		22.48	***41.26***	33.05	36.7	60.89
Crackformer II [11]		17.51	19.64	61.81	29.81	57.88
CT-CrackSeg [53]	Transformer	17.52	24.44	38.21	29.81	58.21
EfficientCrackNet [26]		16.79	26.76	31.06	28.75	57.93
MSDCrack [29]		17.37	19.93	57.47	29.6	57.86
Hybrid-Segmentor [28]		17.2	30.71	28.11	29.35	58.19
Proposed	CNN	***27.31***	29.71	***77.21***	***42.91***	***63.03***

achieving a +0.9% improvement in IoU and nearly +0.8% in F1-score compared to the second-best model (LMM), confirming robust generalization in visually challenging conditions. While transformer-based models like CrackFormer and EfficientCrackNet attain high Recall (up to +2% higher), they suffer from imbalanced Precision, leading to over −3% lower F1 compared to CrackRefineNet. Classical CNNs such as U-Net and DeepCrack trail behind by more than −1.7% in F1 and −2.0% in IoU. These results demonstrate that CrackRefineNet effectively balances detection sensitivity and precision, outperforming both compact and attention-augmented models across diverse scenarios.

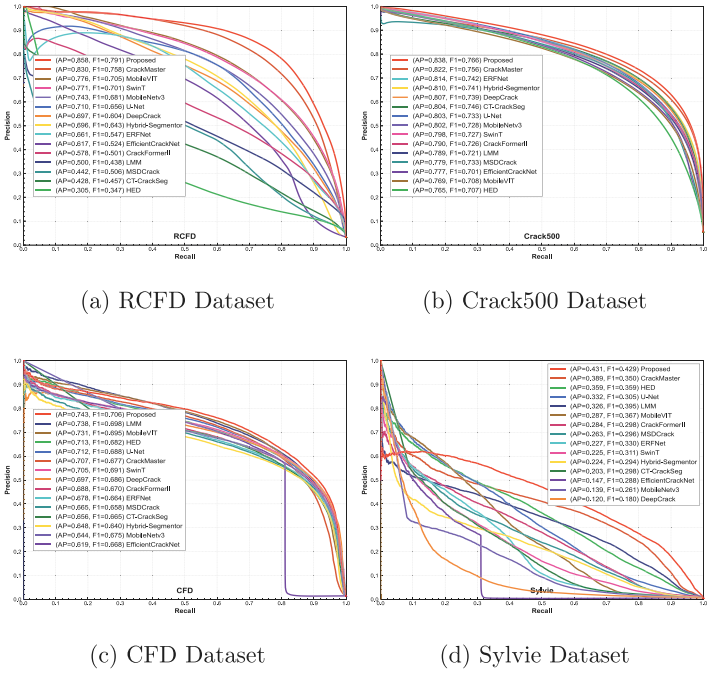
Comparative Experiments on Sylvie Dataset (Zero-shot): Table 9 presents the zero-shot generalization performance, where models trained on CFD were evaluated directly on Sylvie without fine-tuning. Unlike CFD, which primarily contains crack images captured under relatively consistent urban road conditions, the Sylvie dataset introduces stronger domain shift through its diverse road types, acquisition settings, and environmental factors. It features higher variability in surface textures, illumination, and crack morphology, making it a more challenging benchmark for evaluating generalization.

CrackRefineNet exhibits strong transferability, achieving a relative improvement of +3.4% in F1-score and +1.3% in mIoU over the next best model (LMM). While LMM shows solid overall consistency, the proposed model benefits from superior structural awareness and modular design, leading to better domain robustness. Notably, CrackMaster obtains the highest Recall (over +3% compared to the proposed approach) but suffers from low Precision, resulting in a weaker F1-score. Transformer-based methods like MobileViT and SwinT show competitive precision, yet their limited recall constrains overall performance.

These findings highlight how CrackRefineNet's architecture facilitates domain generalization: CAM improves robustness against complex textures and uneven illumination; AFR enhances sensitivity to thin or fragmented cracks that may differ across domains; and SFA adaptively integrates local and global cues, stabilizing predictions in unfamiliar environments. Together, these modules enable a more balanced trade-off between detection sensitivity and precision under domain shift, supporting the model's adaptability to unseen infrastructure conditions.

Overall, the proposed CrackRefineNet consistently surpasses both convolutional and transformer-based baselines across all evaluation scenarios, including standard supervised training and zero-shot inference, highlighting its robustness, adaptability, and suitability for real-world deployment.

Beyond the quantitative scores presented in Tables 6–9, Fig. 9 illustrates the Precision–Recall (PR) curves across all datasets. In these plots, higher curves indicate better precision–recall trade-offs. The proposed method (red) maintains dominant PR trajectories in all cases. On RCFD 9(a) and Crack500 9(b), it exhibits consistently high precision across the entire recall spectrum, with near-perfect values on Crack500. Although performance narrows on the more complex CFD dataset 9(c), the proposed model still secures the top PR curve. Under domain shift in the Sylvie zero-shot setting 9(d), it achieves the highest area under the curve, especially excelling in the 0.0–0.6 recall range, where other methods degrade significantly. These results collectively affirm the



(a) RCFD Dataset (b) Crack500 Dataset

(c) CFD Dataset (d) Sylvie Dataset

Fig. 9. Precision–Recall (PR) curves for different segmentation models on four datasets.

model's strong generalization capacity and resilience across diverse and challenging crack segmentation scenarios.

5.3. Qualitative analysis of results

To complement the quantitative evaluation, a qualitative comparison of segmentation outputs across three representative datasets: RCFD, Crack500, and CFD is presented in Fig. 10. From left to right, each column of Fig. 10 shows the input image, ground truth (GT), and predicted overlays from multiple models. Blue regions indicate true positives, red regions are false positives, and green regions are false negatives, with best IoU scores highlighted in yellow. The visualized predictions include various state-of-the-art models: CNN-based architectures (U-Net, HED, DeepCrack, MobileNetV3, ERFNet, LMM, EfficientCrackNet), transformer-based models (SwinT, MobileViT, CrackFormer II, CT-CrackSeg), the recent CrackMaster, and the proposed CrackRefineNet.

On the RCFD dataset (top rows), traditional CNN models like U-Net and HED exhibit both under-segmentation (false negatives) and over-segmentation (false positives) by introducing excessive background noise from shadows, textures, and road markings. While transformer-based models such as MobileViT and CrackFormer II improve contextual understanding, they still miss fine crack edges. In contrast, CrackRefineNet produces cleaner, more precise boundaries with minimal over- or under-segmentation, showing robustness to illumination changes and cluttered road scenes.

For Crack500 (middle rows), which features thin and highly fragmented cracks, most baseline models struggle to preserve crack continuity and often generate noisy predictions. While DeepCrack and CrackMaster partially improve connectivity, they still suffer from artifacts. CrackRefineNet, however, better maintains the structural integrity of fine cracks while suppressing irrelevant textures, resulting in more complete and accurate predictions.

On the CFD dataset (bottom rows), lightweight models like EfficientCrackNet and LMM fail to capture micro-level details, leaving large missing areas, while transformer models provide smoother but less detailed outputs. Again, CrackRefineNet achieves the most balanced results, capturing both thin and wider cracks while reducing false detections, demonstrating strong adaptability to varying background patterns.

Overall, the qualitative comparison highlights three core strengths of CrackRefineNet: (1) improved boundary precision, (2) better preservation of thin and fragmented cracks, and (3) superior generalization across diverse domains.

These visual findings are consistent with the quantitative metrics, further confirming the effectiveness and robustness of the proposed framework.

5.4. Visual insights into model interpretability

Fig. 11 provides an intuitive look into how different design choices affect the model's focus during crack segmentation. From left to right, each row of Fig. 11 displays the input image, ground truth (GT), baseline model, baseline + AFR module, baseline + AFR and CAM modules, baseline + AFR, CAM, and SFA modules (decoder bypassed), the full proposed model, and feature focus heatmaps from the encoder. This progression highlights how each module progressively refines the model's ability to attend to crack regions while suppressing background noise. Specifically, columns (c)–(f) present Grad-CAM [49] heatmaps generated from the variations of the model, while column (g) shows the full CrackRefineNet model.

The baseline model shown in Fig. 11(c) produces fragmented and unsmooth activations, often missing thin or faint cracks, while also showing slight attention to background regions. Adding the AFR module shown in Fig. 11(d) enhances focus around thin crack structures, demonstrating its role in refining feature representation. The CAM module (e) encourages more context-aware activations, improving the model's ability to localize cracks in cluttered regions. Incorporating the SFA module shown in Fig. 11(f), even in a bypassed setting, produces sharper attention around fine-scale details. Finally, the complete model

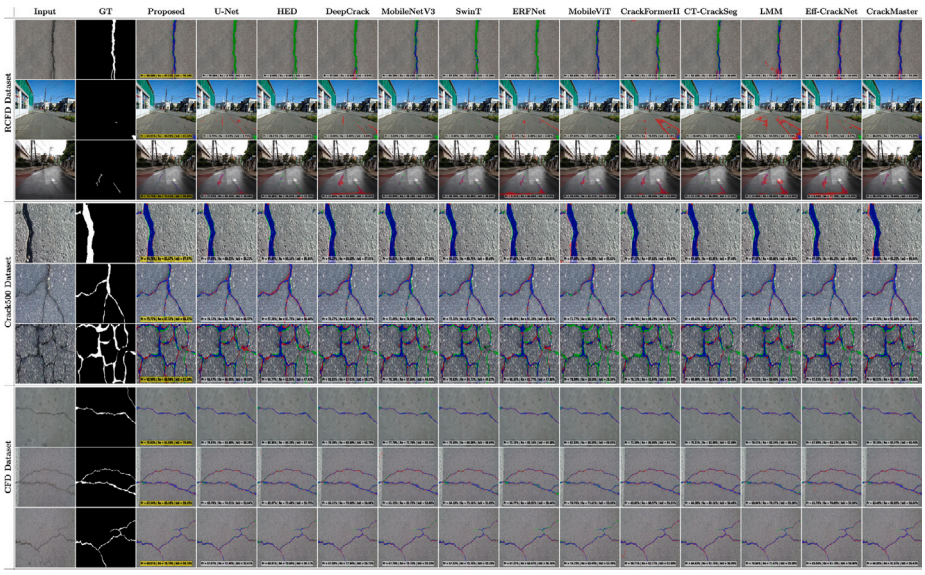


Fig. 10. Qualitative comparison of crack segmentation methods and CrackRefineNet using three datasets (RCFD, Crack500 and CFD).

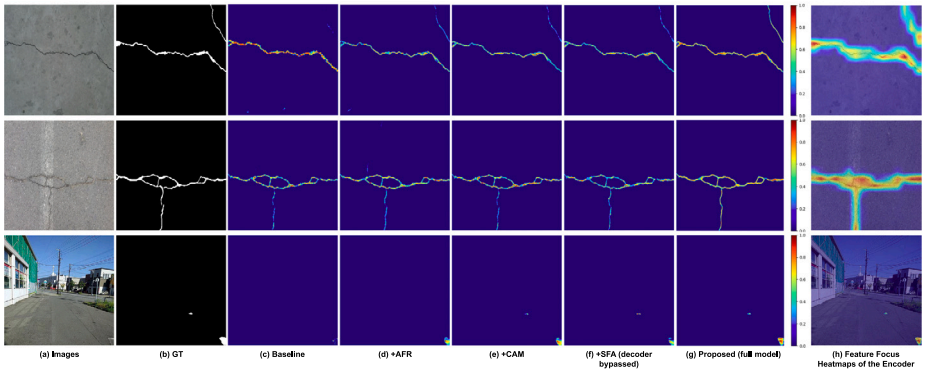


Fig. 11. Visualization of heatmaps for the proposed model and its variations.

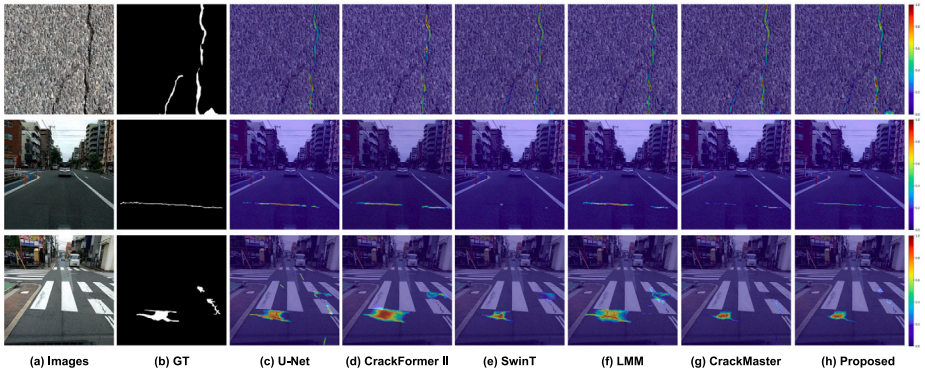


Fig. 12. Qualitative heatmap comparisons between different models and the proposed one with three representative challenges.

shown in Fig. 11(g) integrates these contributions, yielding the most precise and concentrated activations aligned with ground truth cracks.

For note, the column (h) of Fig. 11 displays heatmaps extracted from the encoder stage, highlighting how high-level features evolve before reaching the decoder. The consistent alignment between the highlighted regions and annotated cracks validates the interpretability of CrackRefineNet, showing that its modules not only improve accuracy but also guide the model toward meaningful structural patterns rather than irrelevant background noise.

To further highlight the superiority of CrackRefineNet under real-world challenges, Fig. 12 compares activation heatmaps across baseline models (U-Net, CrackFormer II, SwinT, LMM, CrackMaster) and the proposed framework. The figure displays three representative scenarios in rows: (top) thin/fragmented cracks, (middle) illumination variations, and (bottom) complex textures/road markings. From left to right, columns of Fig. 12 show: (a) input image, (b) ground truth (GT), (c) U-Net, (d) CrackFormer II, (e) SwinT, (f) LMM, (g) CrackMaster, and (h) the proposed model. All heatmaps share the same color scale

Table 10
Trustworthiness and reliability comparison of different models using RGA and RPBT metrics.

Model	RCFD Dataset				Crack500 Dataset					CFD Dataset			
	RGA% ↑	RPBT% ↓			RGA% ↑	RPBT% ↓				RGA% ↑	RPBT% ↓		
		t = 0.95	t = 0.96	t = 0.97		t = 0.95	t = 0.96	t = 0.97	t = 0.98		t = 0.95	t = 0.96	t = 0.97
U-Net	0.4746	**0.0001**	0.0095	0.2533	0.6912	**0.0000**	0.0038	0.4965	0.9995	0.6720	0.0438	0.0651	0.0989
HED	0.1202	0.0017	0.0731	0.6072	0.6586	**0.0000**	0.0413	0.8338	1.0000	0.6646	0.0426	0.0634	0.0964
ERFNet	0.4196	0.2705	0.8280	0.9973	0.7017	**0.0000**	0.0033	0.4758	0.9994	0.6551	0.0483	0.0716	0.1082
DeepCrack	0.4133	0.0112	0.2154	0.8337	0.6943	**0.0000**	0.0058	0.5572	0.9997	0.6714	0.0449	0.0668	0.1013
MobileNetv3	0.3822	**0.0000**	0.0002	0.0290	0.6817	**0.0000**	0.0063	0.5688	0.9998	0.6686	0.0483	0.0716	0.1082
LMM	0.3933	0.9986	1.0000	1.0000	0.6835	**0.0000**	0.0120	0.6638	0.9999	**0.6853**	**0.0409**	**0.0610**	**0.0929**
CrackMaster	**0.5475**	**0.0000**	**0.0000**	**0.0016**	**0.7219**	**0.0000**	**0.0011**	**0.3351**	**0.9979**	0.6457	0.0525	0.0775	0.1167
SwinT	0.4129	**0.0000**	**0.0001**	0.0146	0.6861	**0.0000**	0.0083	0.6099	0.9999	0.6798	0.0437	0.0650	0.0988
MobileViT	0.4608	**0.0000**	**0.0001**	0.0136	0.6535	**0.0000**	0.0276	0.7820	1.0000	0.6799	0.0414	0.0618	0.0941
Crackformer II	0.4184	0.8023	0.9930	1.0000	0.6851	**0.0000**	0.0149	0.6949	0.9999	0.6630	0.0527	0.0778	0.1171
CT-CrackSeg	0.2967	0.0509	0.4539	0.9528	0.7129	**0.0000**	0.0076	0.5965	0.9998	0.6539	0.0520	0.0769	0.1158
EfficientCrackNet	0.3740	0.3973	0.9049	0.9992	0.6648	**0.0000**	0.0159	0.7045	1.0000	0.6634	0.0535	0.0790	0.1188
MSDCrack	0.3889	0.0627	0.4965	0.9628	0.6872	**0.0000**	0.0151	0.7167	1.0000	0.6531	0.0563	0.0829	0.1243
Hybrid-Segmentor	0.3377	**0.0000**	0.0005	0.0484	0.6824	**0.0000**	0.0034	0.5013	0.9997	0.6172	0.0482	0.0714	0.1080
Proposed	**0.6282**	**0.0000**	**0.0000**	**0.0004**	**0.7276**	**0.0000**	**0.0004**	**0.2315**	**0.9944**	**0.6997**	**0.0373**	**0.0558**	**0.0854**

for direct comparison. Baseline CNNs such as U-Net often fail to capture thin structures, while transformer-based models (CrackFormer II, SwinT) produce blurred or incomplete activations. Multi-scale CNNs (LMM, CrackMaster) achieve stronger localization but remain noisy or excessively thick. In contrast, compared with baselines (which miss fine structures, blur responses, or activate on background clutter), CrackRefineNet generates sharper, more continuous responses closely aligned with the ground truth, confirming the benefits of CAM, AFR, and SFA in handling these challenging conditions.

5.5. Trustworthy and reliability analysis

The outcomes presented in Table 10 provide a comprehensive evaluation of the models' reliability and safety by incorporating the Rank Graduation Box Accuracy (RGA) [50] and Risk Probability Below Threshold (RPBT) [51] metrics. The table reports RGA and RPBT values at four confidence thresholds: t = 0.95, 0.96, 0.97, and 0.98, where red bold indicates the best performance and blue bold highlights the second-best. These indicators assess not only segmentation performance but also the consistency and trustworthiness of predictions under varying confidence thresholds.

Across the RCFD, Crack500, and CFD datasets, the proposed model achieves the highest RGA scores of 62.82%, 72.76%, and 69.97%, respectively, significantly surpassing all competing methods. These high RGA values reflect the model's superior capability in ranking segmentation outputs accurately in alignment with ground truth relevance. The second-best RGA results were observed with CrackMaster on RCFD (54.75%) and Crack500 (72.19%), and LMM on CFD (68.53%), yet none approached the consistency of the proposed model.

In terms of RPBT, which quantifies the likelihood of high-risk predictions under threshold constraints, CrackRefineNet demonstrated the most reliable behavior. It achieved perfect or near-zero RPBT scores across all thresholds: $t = 0.95$, 0.96, and 0.97 on the RCFD and CFD datasets, and $t = 0.95$, 0.96, 0.97, and 0.98 on the Crack500 dataset. Specifically, CrackRefineNet obtained particularly low values of 0.0000, 0.0000, and 0.0004 on RCFD; 0.0000, 0.0004, 0.2315, and 0.9944 on Crack500; and 0.0373, 0.0558, and 0.0854 on CFD. These results demonstrate that the model makes few mistakes when using strict confidence thresholds, confirming its stability and reliability.

While other models, such as MobileNetv3, SwinT, and CrackMaster, occasionally recorded competitive RPBT values at individual thresholds, none matched the proposed model's consistently low-risk profile across all scenarios. Notably, many models achieved a risk score of 0.0000 at $t = 0.95$, but only the proposed approach maintained this advantage as the threshold increased highlighting its robustness in high-stakes applications.

By incorporating the SAFE AI framework [50] and Bayesian reliability principles [51], this analysis offers a holistic view of trust and uncertainty. These metrics are critical for real-world deployment, particularly in safety-critical infrastructure monitoring. In summary, the proposed model's dominant performance across both RGA and RPBT establishes it as the most trustworthy and reliable solution among contemporary crack segmentation architectures.

5.6. Effect of CAM, AFR, and SFA modules

In addition to the quantitative ablation results, the individual effect of the three proposed modules is qualitatively analyzed.

Effect of CAM. Fig. 13 demonstrates the influence of CAM on boundary preservation. From left to right, the figure shows: the input image, ground truth (GT) boundaries, predicted boundaries without CAM, predicted boundaries with CAM, and the CAM attention map. For visualization, crack boundaries are extracted with green contours representing the GT boundaries and red contours denoting the predicted boundaries. Without CAM, the predicted crack edges appear less sharp and partially fragmented. With CAM, the boundaries are noticeably sharper and more continuous, confirming its ability to leverage shallow features to emphasize fine crack contours while suppressing background noise. The CAM attention map (rightmost column) illustrates how the module emphasizes crack contours while suppressing background noise, thereby improving fine-grained boundary preservation.

Effect of AFR. Fig. 14 highlights AFR's role in improving the detection of low-contrast and faint cracks. From left to right, each row displays: the input image, ground truth (GT), prediction without AFR, prediction with AFR, and the AFR attention map. Predictions without AFR often miss or fragment continuous cracks due to weak responses in deeper layers. With AFR, these regions are more accurately recovered and connected, producing more complete crack structures. Importantly, AFR also enhances the detection of naturally disconnected cracks, ensuring that such cracks are still correctly identified rather than being suppressed as noise. Thus, AFR reduces false discontinuities while also capturing genuine disconnected cracks, as illustrated across the three representative cases.

Effect of SFA. While the effect of SFA is less visually evident, its contribution is consistently reflected in the ablation study (Tables 3–5). By selectively weighting and aggregating features from multiple levels, SFA enhances structural consistency across scales. This leads to smoother and more coherent crack predictions, particularly in cluttered or large-scale scenes.

Together, these qualitative and quantitative results confirm the complementary roles of CAM, AFR, and SFA in enhancing CrackRefineNet's robustness for realistic crack segmentation.

5.7. Complexity analysis

The computational complexity of the proposed model is evaluated against diverse crack segmentation baselines considering three key factors: number of trainable parameters, inference time per image, and

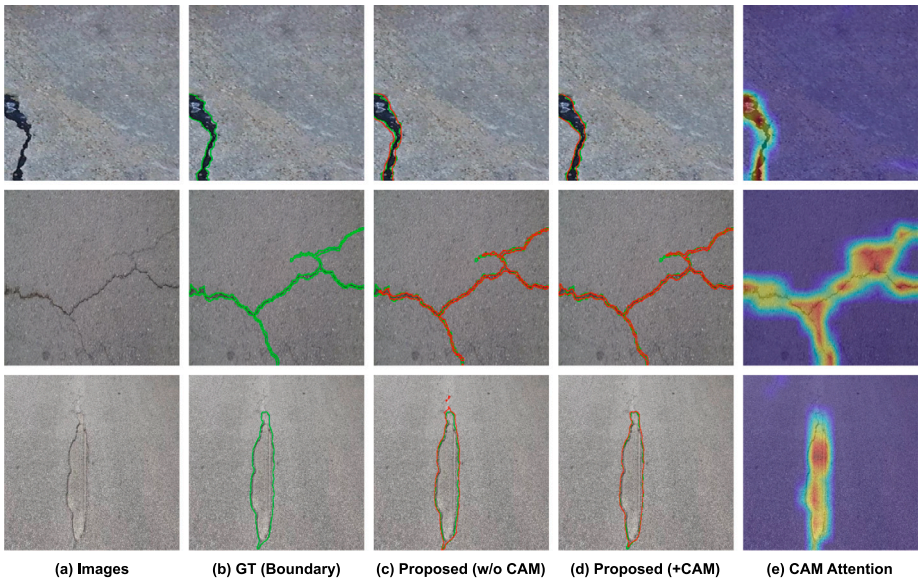


Fig. 13. Comparison of crack boundary predictions with and without CAM.

Table 11
Comparative analysis of trained parameters, inference time, FLOPs, and FPS of different segmentation models.

Model	Parameters (M)	Inference time (s)	FLOPs (T)	FPS
U-Net [10]	31.04	0.0192	0.193	52
HED [25]	14.72	0.0083	0.080	120
ERFNet [52]	2.08	0.0092	0.015	108
DeepCrack [2]	3.68	0.0045	0.020	222
MobileNetv3 [46]	3.28	0.0105	0.008	95
LMM [15]	0.84	0.0981	0.035	10
CrackMaster [12]	93.61	0.0192	0.087	52
SwinT [47]	58.94	0.0298	0.236	33
MobileViT [48]	31.85	0.0629	0.841	16
Crackformer II [11]	4.96	0.0715	0.091	14
CT-CrackSeg [53]	22.88	0.0304	0.195	33
EfficientCrackNet [26]	0.35	0.0532	0.045	19
MSDCrack [29]	15.39	0.0296	0.008	34
Hybrid-Segmentor [28]	347	0.0388	0.142	26
Proposed	99.66	0.0387	0.206	26

floating-point operations per second (FLOPs). Frames per second (FPS), computed as the reciprocal of inference time, is also reported to assess real-time feasibility.

Table 11 summarizes these statistics across 15 representative models, including both classical convolution-based architectures (e.g., U-Net [10], DeepCrack [2]), lightweight designs (e.g., MobileNetv3 [46], LMM [15], EfficientCrackNet [26]), and recent transformer-based networks (e.g., SwinT [47], MobileViT [48], Crackformer II [11], Hybrid-Segmentor [28]).

Despite having the highest number of parameters (99.66M) among all models, CrackRefineNet achieves a competitive inference time of 0.0387 s and moderate FLOPs of 0.206T, while sustaining ≈ 26 FPS. This is a direct result of the parallel decoder streams and refined feature enhancement modules, which slightly increase complexity but significantly improve segmentation accuracy. When compared to similarly sized models like CrackMaster [12] (93.61M) or SwinT (58.94M), CrackRefineNet shows only a minor increase in latency and computational cost while delivering higher performance across all evaluation metrics.

On the efficiency spectrum, extremely lightweight models such as EfficientCrackNet (0.35M, 0.0532s, 0.045T, 19 FPS) and LMM (0.84M, 0.0981s, 0.035T, 10 FPS) offer low complexity but often sacrifice segmentation quality. In contrast, transformer-based models like MobileViT and SwinT have relatively high FLOPs, particularly MobileViT (0.841T), making them less suitable for real-time or resource-constrained environments.

Overall, the proposed model achieves a good balance between performance and computational efficiency, maintaining near real-time capability at ≈ 26 FPS while achieving superior segmentation results. This makes it suitable for high-accuracy applications where a moderate increase in inference cost is acceptable in exchange for robust crack detection.

5.8. Analysis of failure cases

To better understand the limitations of the proposed method, representative failure cases from the RCFD, Crack500, and CFD datasets are analyzed, as shown in Fig. 15. The figure displays three rows showing input images (top), ground truth (middle), and predictions from the proposed method (bottom). Each column group presents three challenging samples from each dataset where the method struggles with ambiguous boundaries, low contrast, or complex textures. These examples highlight challenging conditions that affect the model's performance.

In the RCFD dataset, failures often occur in regions with low contrast caused by shadows or lighting variations. Additionally, full-scene images that include many irrelevant objects such as vehicles, buildings, and lane markings introduce distractions, leading to false positives or missed cracks. In the Crack500 dataset, highly textured surfaces sometimes resemble cracks, resulting in fragmented predictions or missed thin crack branches, particularly at texture intersections. In the CFD dataset, the model generally captures the correct crack topology but struggles to maintain consistent crack width. This occasionally leads to small gaps along the crack path, indicating difficulty with precise boundary refinement.

RCFD is primarily affected by cluttered scenes and lighting inconsistencies. Crack500 presents challenges due to texture ambiguity and the presence of fine hairline cracks, while CFD reveals limitations in preserving uniform crack thickness on smooth concrete surfaces.

These failure cases suggest the need for further improvements, such as adaptive contrast enhancement, texture-aware attention mechanisms, and multi-scale boundary refinement modules, to more effectively handle subtle and ambiguous crack patterns. Despite these

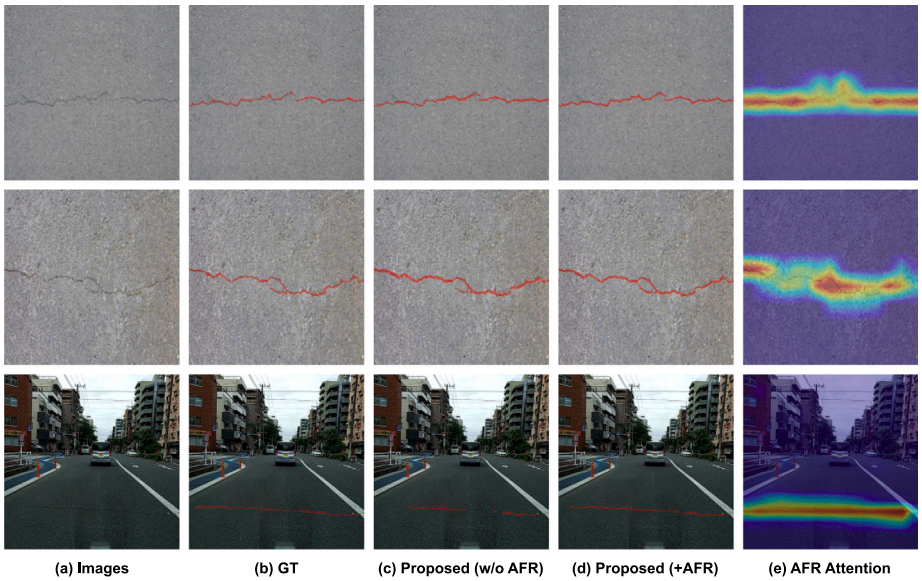


Fig. 14. Comparison of crack predictions with and without AFR.

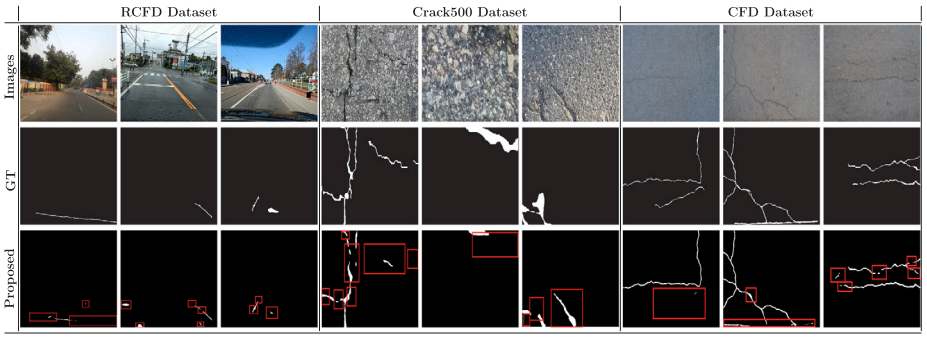


Fig. 15. Failure cases of the proposed method across three benchmark datasets (RCFD, Crack500 and CFD).

challenges, such failures are relatively rare and do not significantly affect the overall strong performance of the proposed model.

5.9. Limitations

While CrackRefineNet demonstrates strong performance across diverse datasets and conditions, several limitations remain. First, the model is sensitive to extreme conditions such as heavy occlusion (e.g., leaves, dirt, or vehicles covering cracks), severe contamination, and extreme weather effects like rain or snow. These cases reduce visibility and can confuse both spatial and contextual cues. Second, although the proposed modules improve robustness against texture ambiguity and illumination changes, performance may degrade when cracks are faintly visible under severe lighting inconsistencies. Finally, the model requires moderate computational resources, which could limit its direct deployment in low-power embedded systems. These limitations suggest potential research directions, including improved contrast modeling, adaptive attention, and lightweight design optimization.

5.10. Real-world case study

To further assess the practical applicability of the proposed CrackRefineNet framework, a pilot deployment was conducted using real-world road surface images collected from Tarragona City, Spain. Unlike benchmark datasets, these images exhibit highly diverse illumination conditions, varying asphalt textures, and significant noise from environmental factors such as dirt, shadows, and water stains. Such uncontrolled conditions present additional challenges for crack detection, making this case study a critical test of the model's robustness and adaptability.

Fig. 16 illustrates representative results from this real-world deployment. Each column pair consists of a raw road image and the corresponding predicted crack mask generated by CrackRefineNet. Despite the complexity of the scenes, CrackRefineNet demonstrates a strong ability to accurately localize cracks, preserving fine-grained structures while minimizing false detections on background textures. Notably, even thin and fragmented cracks are detected with high fidelity, validating strong generalization beyond curated training datasets.

This case study highlights the potential of the proposed approach for practical urban infrastructure monitoring. By achieving consistent and reliable predictions under real-world conditions, CrackRefineNet shows promise for integration into automated pavement inspection systems, reducing the need for manual surveys and improving maintenance planning for road networks.

6. Conclusion and future work

This paper presented CrackRefineNet, a context- and refinement-driven convolutional framework for robust crack segmentation in real-world and zero-shot conditions. By integrating three complementary modules—Context-Aware Modeling (CAM), Adaptive Feature Refinement (AFR), and Selective Feature Aggregation (SFA)—the model

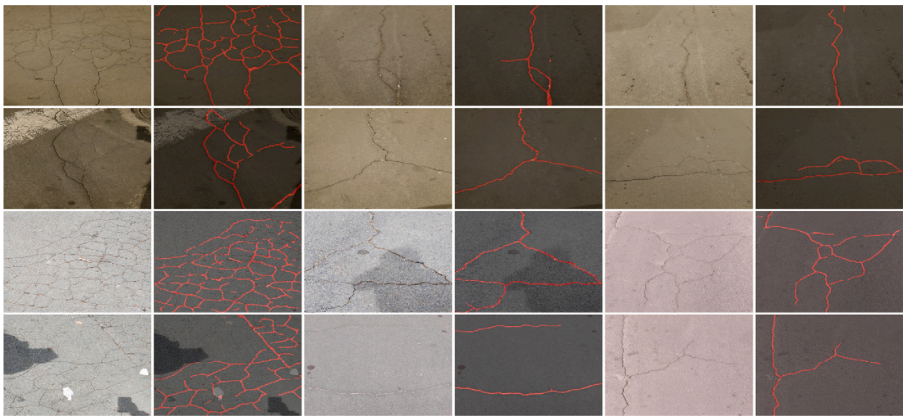


Fig. 16. Real-world case study results using the Tarragona City Road Crack Dataset. Col 1, 3 and 5: original images, and Col 2, 4 and 8 original images with crack segmentation in a red color.

achieves superior accuracy and generalization across diverse datasets. On the supervised benchmarks (i.e., RCFD, Crack500, and CFD), CrackRefineNet achieved up to +3.5% improvement in IoU and +2.8% in F1-score compared to the strongest state-of-the-art baselines, while maintaining the highest zero-shot F1-score of 42.91% on the Sylvie dataset. These results confirm the model's capability to preserve boundary precision, segmentation continuity, and resilience across heterogeneous acquisition domains.

Beyond numerical performance, this study contributes new methodological insights to the field. It demonstrates how multi-scale contextual encoding and adaptive refinement can be jointly optimized within convolutional frameworks to enhance structural coherence, improve robustness to domain shifts, and balance global context aggregation with local boundary precision. The synergy between CAM, AFR, and SFA thus advances the understanding of refinement-based architectures for infrastructure inspection.

Additionally, the introduction of the RCFD dataset further reinforces this contribution by providing a realistic full-scene benchmark with pixel-level annotations, enabling reproducible and context-aware evaluation in complex outdoor environments. While CrackRefineNet shows strong potential for practical deployment in road inspection, its modular design can be readily extended to bridges, tunnels, and airport pavements.

Future work will focus on improving generalization through self-supervised pretraining and domain adaptation, integrating uncertainty quantification for interpretability, and developing lightweight model variants for real-time applications in resource-constrained settings.

CRediT authorship contribution statement

Ammar M. Okran: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Domenec Puig:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Saif Khalid Musluh:** Writing – review & editing. **Sylvie Chambon:** Writing – review & editing, Methodology, Formal analysis, Conceptualization. **Hatem A. Rashwan:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is part of the PECT "Cuidem el que ens uneix" project, Operation 4, within the framework of the RIS3CAT and ERDF Catalonia Operational Programme 2014–2020. PECT, Spain is co-financed by the Catalan Government, the Provincial Council of Tarragona, Spain, "Diputació de Tarragona, Spain" and Universitat Rovira i Virgili, Spain.

Data availability

Code supporting the findings of this study are available at https://github.com/AmmarOkran/CrackRefineNet.

References

[1] V. Hoskere, Y. Narazaki, T. Hoang, B. Spencer, Vision-based structural inspection using multiscale deep convolutional neural networks, 2018, http://dx.doi.org/10.48550/ARXIV.1805.01055, URL https://arxiv.org/abs/1805.01055.

[2] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: A deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153, http://dx.doi.org/10.1016/j.neucom.2019.01.036.

[3] H. Oliveira, P.L. Correia, Automatic road crack detection and characterization, IEEE Trans. Intell. Transp. Syst. 14 (1) (2013) 155–168, http://dx.doi.org/10.1109/tits.2012.2208630.

[4] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transp. Syst. 17 (12) (2016) 3434–3445, http://dx.doi.org/10.1109/tits.2016.2552248.

[5] A. Ellenberg, A. Kontsos, F. Moon, I. Bartoli, Bridge related damage quantification using unmanned aerial vehicle imagery, Struct. Control. Health Monit. 23 (9) (2016) 1168–1179, http://dx.doi.org/10.1002/stc.1831.

[6] A.M. Okran, M. Abdel-Nasser, H.A. Rashwan, D. Puig, Effective deep learning-based ensemble model for road crack detection, in: 2022 IEEE International Conference on Big Data (Big Data), IEEE, 2022, pp. 6407–6415, http://dx.doi.org/10.1109/bigdata55660.2022.10020790.

[7] A.M. Okran, M. Abdel-Nasser, H.A. Rashwan, D. Puig, A curated dataset for crack image analysis: Experimental verification and future perspectives, in: Artificial Intelligence Research and Development, IOS Press, 2022, pp. 225–228, http://dx.doi.org/10.3233/faia220342.

[8] A.M. Okran, H.A. Rashwan, S. Abdulwahab, S. Chambon, D. Puig, Annotation-efficient crack segmentation in full scene images via bounding box-guided feature modulation, in: Artificial Intelligence Research and Development, IOS Press, 2025, p. 243, http://dx.doi.org/10.3233/faia250599.

[9] A.M. Okran, H.A. Rashwan, S. Chambon, D. Puig, Annotation-efficient and domain-general segmentation from weak labels: A bounding box-guided approach, Electronics 14 (19) (2025) 3917, http://dx.doi.org/10.3390/electronics14193917.

[10] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[11] H. Liu, J. Yang, X. Miao, C. Mertz, H. Kong, CrackFormer network for pavement crack segmentation, IEEE Trans. Intell. Transp. Syst. 24 (9) (2023) 9240–9252, http://dx.doi.org/10.1109/tits.2023.3266776.
[12] A.M. Okran, H.A. Rashwan, A. Saleh, D. Puig, Efficient crack segmentation with multi-decoder networks and enhanced feature fusion, Eng. Appl. Artif. Intell. 152 (2025) 110697, http://dx.doi.org/10.1016/j.engappai.2025.110697.
[13] A.M. Okran, H.A. Rashwan, D. Puig, Enhanced crack segmentation network: Leveraging multi-dimensional attention, in: Artificial Intelligence Research and Development, IOS Press, 2024, pp. 94–96, http://dx.doi.org/10.3233/faia240416.
[14] A.M. Okran, A. Saleh, D. Puig, H.A. Rashwan, Stacking up for success: A cascade network model for efficient road crack segmentation, in: Artificial Intelligence Research and Development, IOS Press, 2023, pp. 38–47, http://dx.doi.org/10.3233/faia230657.
[15] O. Al-maqtari, B. Peng, Z. Al-Huda, A. Al-Malahi, N. Maqtary, Lightweight yet effective: A modular approach to crack segmentation, IEEE Trans. Intell. Veh. 9 (12) (2024) 7961–7972, http://dx.doi.org/10.1109/tiv.2024.3405495.
[16] D. Arya, H. Maeda, S.K. Ghosh, D. Toshniwal, Y. Sekimoto, RDD2022: A multi-national image dataset for automatic road damage detection, Geosci. Data J. (2024) http://dx.doi.org/10.1002/gdj3.260.
[17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A ConvNet for the 2020s, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2022, pp. 11976–11986, http://dx.doi.org/10.1109/cvpr52688.2022.01167.
[18] S. Chambon, J.-M. Moliard, Automatic road pavement assessment with image processing: Review and comparison, Int. J. Geophys. 2011 (2011) 1–20, http://dx.doi.org/10.1155/2011/989354.
[19] R. Amhaz, S. Chambon, J. Idier, V. Baltazart, Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection, IEEE Trans. Intell. Transp. Syst. 17 (10) (2016) 2718–2729, http://dx.doi.org/10.1109/tits.2015.2477675.
[20] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: Automatic crack detection from pavement images, Pattern Recognit. Lett. 33 (3) (2012) 227–238, http://dx.doi.org/10.1016/j.patrec.2011.11.004.
[21] Y. Sari, P.B. Prakoso, A.R. Baskara, Road crack detection using support vector machine (SVM) and otsu algorithm, in: 2019 6th International Conference on Electric Vehicular Technology, ICEVT, IEEE, 2019, pp. 349–354, http://dx.doi.org/10.1109/icevt48285.2019.8993969.
[22] P.-j. Chun, S. Izumi, T. Yamane, Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine, Computer-Aided Civ. Infrastruct. Eng. 36 (1) (2020) 61–72, http://dx.doi.org/10.1111/mice.12564.
[23] X. Wang, X. Wang, J. Li, W. Liang, C. Bi, Research on pavement crack detection based on random structure forest and density clustering, Automation 5 (4) (2024) 467–483, http://dx.doi.org/10.3390/automation5040027.
[24] K.S. Bhalaji Kharthik, E.M. Onyema, S. Mallik, B.V.V. Siva Prasad, H. Qin, C. Selvi, O.K. Sikha, Transfer learned deep feature based crack detection using support vector machine: a comparative study, Sci. Rep. 14 (1) (2024) 14517, http://dx.doi.org/10.1038/s41598-024-63767-5.
[25] S. Xie, Z. Tu, Holistically-nested edge detection, in: 2015 IEEE International Conference on Computer Vision, ICCV, IEEE, 2015, pp. 1395–1403, http://dx.doi.org/10.1109/iccv.2015.164.
[26] A.H. Zim, A. Iqbal, Z. Al-Huda, A. Malik, M. Kuribayashi, EfficientCrackNet: A lightweight model for crack segmentation, in: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, IEEE, 2025, pp. 6279–6289, http://dx.doi.org/10.1109/wacv61041.2025.00612.
[27] Y. Yan, J. Sun, H. Zhang, C. Tang, X. Wu, S. Wang, Y. Zhang, DCMA-net: A dual channel multi-scale feature attention network for crack image segmentation, Eng. Appl. Artif. Intell. 148 (2025) 110411, http://dx.doi.org/10.1016/j.engappai.2025.110411.
[28] J.M. Goo, X. Milidonis, A. Artusi, J. Boehm, C. Ciliberto, Hybrid-segmentor: Hybrid approach for automated fine-grained crack segmentation in civil infrastructure, Autom. Constr. 170 (2025) 105960, http://dx.doi.org/10.1016/j.autcon.2024.105960.
[29] J. Wang, H. Yao, J. Hu, Y. Ma, J. Wang, Dual-encoder network for pavement concrete crack segmentation with multi-stage supervision, Autom. Constr. 169 (2025) 105884, http://dx.doi.org/10.1016/j.autcon.2024.105884.
[30] Y.-C. Chen, R.-T. Wu, A. Puranam, Multi-task deep learning for crack segmentation and quantification in RC structures, Autom. Constr. 166 (2024) 105599, http://dx.doi.org/10.1016/j.autcon.2024.105599.
[31] J. Liang, X. Gu, D. Jiang, Q. Zhang, CNN-based network with multi-scale context feature and attention mechanism for automatic pavement crack segmentation, Autom. Constr. 164 (2024) 105482, http://dx.doi.org/10.1016/j.autcon.2024.105482.
[32] M. Ni, L. Chen, P. Shi, R. Ren, RepCrack: An efficient pavement crack segmentation method based on structural re-parameterization, Eng. Appl. Artif. Intell. 141 (2025) 109791, http://dx.doi.org/10.1016/j.engappai.2024.109791.
[33] Z. Zhang, B. Peng, T. Zhao, An ultra-lightweight network combining mamba and frequency-domain feature extraction for pavement tiny-crack segmentation, Expert Syst. Appl. 264 (2025) 125941, http://dx.doi.org/10.1016/j.eswa.2024.125941.
[34] J. Liu, P. Niu, L. Kou, Y. Zhang, H. Chang, F. Guo, An interactive cross-multi-feature fusion approach for salient object detection in crack segmentation, Computer-Aided Civ. Infrastruct. Eng. 40 (8) (2025) 1080–1099, http://dx.doi.org/10.1111/mice.13437.
[35] A. Khan, S. Khan, M.A. Elhassan, H. Deng, Advancing road safety: A lightweight feature fusion model for robust road crack segmentation, Adv. Eng. Informatics 65 (2025) 103262, http://dx.doi.org/10.1016/j.aei.2025.103262.
[36] H. Liu, C. Jia, F. Shi, X. Cheng, S. Chen, SCSegamba: Lightweight structure-aware vision mamba for crack segmentation in structures, in: 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2025, pp. 29406–29416, http://dx.doi.org/10.1109/cvpr52734.2025.02738.
[37] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 7132–7141, http://dx.doi.org/10.1109/cvpr.2018.00745.
[38] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2017, pp. 2117–2125, http://dx.doi.org/10.1109/cvpr.2017.106.
[39] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, CBAM: Convolutional block attention module, in: Computer Vision – ECCV 2018, Springer International Publishing, 2018, pp. 3–19, http://dx.doi.org/10.1007/978-3-030-01234-2_1.
[40] Y. Wu, K. He, Group normalization, Int. J. Comput. Vis. 128 (3) (2019) 742–755, http://dx.doi.org/10.1007/s11263-019-01198-w.
[41] C.H. Sudre, W. Li, T. Vercauteren, S. Ourselin, M. Jorge Cardoso, Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer International Publishing, 2017, pp. 240–248, http://dx.doi.org/10.1007/978-3-319-67558-9_28.
[42] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, H. Ling, Feature pyramid and hierarchical boosting network for pavement crack detection, IEEE Trans. Intell. Transp. Syst. 21 (4) (2020) 1525–1535, http://dx.doi.org/10.1109/tits.2019.2910595.
[43] D. Arya, H. Maeda, S.K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, Y. Sekimoto, Deep learning-based road damage detection and classification for multiple countries, Autom. Constr. 132 (2021) 103935, http://dx.doi.org/10.1016/j.autcon.2021.103935.
[44] M. Contributors, MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark, 2020, https://github.com/open-mmlab/mmsegmentation.
[45] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017, http://dx.doi.org/10.48550/ARXIV.1711.05101, URL https://arxiv.org/abs/1711.05101.
[46] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, Q. Le, Searching for MobileNetV3, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, IEEE, 2019, pp. 1314–1324, http://dx.doi.org/10.1109/iccv.2019.00140.
[47] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, IEEE, 2021, pp. 9992–10002, http://dx.doi.org/10.1109/iccv48922.2021.00986.
[48] S. Mehta, M. Rastegari, MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer, 2021, http://dx.doi.org/10.48550/ARXIV.2110.02178, URL https://arxiv.org/abs/2110.02178.
[52] E. Romera, J.M. Alvarez, L.M. Bergasa, R. Arroyo, ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation, IEEE Trans. Intell. Transp. Syst. 19 (1) (2018) 263–272, http://dx.doi.org/10.1109/tits.2017.2750080.
[53] H. Tao, B. Liu, J. Cui, H. Zhang, A convolutional-transformer network for crack segmentation with boundary awareness, in: 2023 IEEE International Conference on Image Processing, ICIP, IEEE, 2023, pp. 86–90, http://dx.doi.org/10.1109/icip49359.2023.10222276.
[49] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision, ICCV, IEEE, 2017, pp. 618–626, http://dx.doi.org/10.1109/iccv.2017.74.
[50] G. Babaei, P. Giudici, E. Raffinetti, A rank graduation box for SAFE AI, Expert Syst. Appl. 259 (2025) 125239, http://dx.doi.org/10.1016/j.eswa.2024.125239.
[51] N.D. Singpurwalla, Reliability and risk: A Bayesian perspective, Wiley, 2006, http://dx.doi.org/10.1002/9780470060346,