Construction and Building Materials

Context-CrackNet: A Context-Aware Framework for Precise Segmentation of Tiny Cracks in Pavement images

--Manuscript Draft--

Manuscript Number:	CONBUILDMAT-D-25-01325R1
Article Type:	Research Paper
Keywords:	pavement distress; Segmentation; deep learning; Cracks; Context-CrackNet; region-focused enhancement; global context modeling
Corresponding Author:	Blessing Agyei Kyem North Dakota State University UNITED STATES
First Author:	Blessing Agyei Kyem
Order of Authors:	Blessing Agyei Kyem
	Joshua Kofi Asamoah
	Armstrong Aboah
Abstract:	The accurate detection and segmentation of pavement distresses, particularly tiny and small cracks, are critical for early intervention and preventive maintenance in transportation infrastructure. Traditional manual inspection methods are labor-intensive and inconsistent, while existing deep learning models struggle with fine-grained segmentation and computational efficiency. To address these challenges, this study proposes Context-CrackNet, a novel encoder-decoder architecture featuring the Region-Focused Enhancement Module (RFEM) and Context-Aware Global Module (CAGM). These innovations enhance the model's ability to capture fine-grained local details and global contextual dependencies, respectively. Context-CrackNet was rigorously evaluated on ten publicly available crack segmentation datasets, covering diverse pavement distress scenarios. The model consistently outperformed 9 state-of-the-art segmentation frameworks, achieving superior performance metrics such as mIoU and Dice score, while maintaining competitive inference efficiency. Ablation studies confirmed the complementary roles of RFEM and CAGM, with notable improvements in mIoU and Dice score when both modules were integrated. Additionally, the model's balance of precision and computational efficiency highlights its potential for real-time deployment in large-scale pavement monitoring systems.

Armstrong Aboah
North Dakota State University
Department of Civil, Construction, and Environmental Engineering

10th April, 2025

Editorial Office
Construction and Building Materials
Elsevier

Dear Editor,

We are pleased to submit our revised manuscript, **"Context-CrackNet: A Context-Aware Framework for Precise Segmentation of Tiny Cracks in Pavement images,"** for your consideration for publication in *Construction and Building Materials*. This research addresses the critical need for accurate, efficient, and scalable solutions in automated pavement distress detection.

Pavement distresses, particularly tiny and subtle cracks, often serve as early indicators of structural degradation. Failure to detect these small cracks at an early stage can result in their progression into larger, costlier damages, significantly impacting road safety and increasing maintenance expenses. However, the precise segmentation of these fine cracks remains a formidable challenge due to their low contrast, variable shapes, and noisy textures, especially in high-resolution pavement imagery. Addressing this challenge is vital for enabling proactive maintenance strategies, optimizing resource allocation, and ensuring the longevity of pavement infrastructure.

Our manuscript makes several notable contributions: (1) we propose **Context-CrackNet**, a novel segmentation framework specifically designed for crack detection challenges; (2) we introduce the **Region-Focused Enhancement Module** (RFEM), which improves our model's ability to detect very small and tiny crack patterns; (3) we develop the **Context-Aware Global Module** (CAGM), which effectively captures broader contextual information to accurately identify larger, more complex distress patterns. Through rigorous evaluations on ten publicly available crack datasets, Context-CrackNet consistently outperformed all selected state-of-the-art segmentation models, demonstrating superior segmentation performance.

We believe this study aligns closely with the scope (which is **highway pavements**) and objectives of *Construction and Building Materials* and offers valuable insights for researchers, engineers, and policymakers in the field. The manuscript is original, has not been published elsewhere, and is not under consideration by any other publication. All authors have reviewed and approved the manuscript, and we declare no conflicts of interest related to this work. Thank you for considering our submission.

Sincerely,
Armstrong Aboah
Assistant Professor
armstrong.aboah@ndsu.edu

Context-CrackNet: A Context-Aware Framework for Precise Segmentation of Tiny Cracks in Pavement Images

Blessing Agyei Kyem[a], Joshua Kofi Asamoah[a], Armstrong Aboah[a,*]

[a]*North Dakota State University, Fargo, 58105, North Dakota, United States*

Abstract

The accurate detection and segmentation of pavement distresses, particularly tiny and small cracks, are critical for early intervention and preventive maintenance in transportation infrastructure. Traditional manual inspection methods are labor-intensive and inconsistent, while existing deep learning models struggle with fine-grained segmentation and computational efficiency. To address these challenges, this study proposes Context-CrackNet, a novel encoder-decoder architecture featuring the Region-Focused Enhancement Module (RFEM) and Context-Aware Global Module (CAGM). These innovations enhance the model's ability to capture fine-grained local details and global contextual dependencies, respectively. Context-CrackNet was rigorously evaluated on ten publicly available crack segmentation datasets, covering diverse pavement distress scenarios. The model consistently outperformed 9 state-of-the-art segmentation frameworks, achieving superior performance metrics such as mIoU and Dice score, while maintaining competitive inference efficiency. Ablation studies confirmed the complementary roles of RFEM and CAGM, with notable improvements in mIoU and Dice score when both modules were integrated. Additionally, the model's balance of precision and computational efficiency highlights its potential for real-time deployment in large-scale pavement monitoring systems.

Keywords: pavement distress, segmentation, deep learning, cracks, Context-CrackNet, region-focused enhancement, global context modeling

*Corresponding author

1. Introduction

Transportation infrastructure is essential to modern society, forming the backbone of economic development by enabling the efficient movement of people and goods. Among these infrastructures, road networks are critical, and maintaining their integrity is imperative to ensure public safety and economic efficiency. Pavement distresses such as cracks and potholes which develop on these road networks not only compromise safety but also lead to costly repairs if not promptly detected and addressed. Accurately detecting these distresses remains a significant challenge due to their irregular shapes, varying sizes, diverse surface textures, and environmental factors such as fluctuating lighting conditions and the presence of debris. Traditionally, the detection of these distresses has relied on manual inspections, which are not only time-consuming and labor-intensive but also prone to subjectivity and inconsistency. These limitations underscore the critical need for automated solutions to improve efficiency and accuracy. To address these challenges, researchers have increasingly turned to automated approaches that leverage advanced image processing and machine learning techniques, offering a more robust and scalable alternative to traditional methods. While early image processing approaches were often inadequate due to the complex nature of pavement surfaces, the introduction of deep learning models particularly convolutional neural networks has significantly advanced the field. These models effectively identify and segment pavement distresses by learning hierarchical features and capturing spatial context. Nevertheless, despite these advancements, current models still face significant limitations that hinder their performance, necessitating further refinements to achieve accurate pavement distress segmentation.

Several deep learning approaches have been proposed for pavement distress segmentation. For instance, Wen et al. proposed PDSNet [1], an efficient framework achieving an MIoU of 83.7% on manually collected 2D and 3D private pavement dataset. However, it struggles with small and tiny cracks. Sarmiento [2] utilized YOLOv4 for detecting and DeepLabv3 for segmenting the pavement distresses. While effective for simpler distresses like delaminations, both models struggle with tiny cracks, scaling, and texture variations, leading to misclassifications and false negatives. Li et al. [3] further introduced a variant of DeepLabV3+ with an adaptive probabilistic sampling method and external attention for pavement distress segmentation. It was evaluated on the CRACK500 dataset, achieving a Mean Intersection over

Union (MIoU) of 54.91%. Tong et al. introduced Evidential Segmentation Transformer (ES-Transformer) [4], combining Dempster–Shafer theory with a transformer backbone for improved segmentation and calibration. Evaluated on the Crack500 dataset, it achieved a Mean Intersection over Union (MIoU) of 59.85%, demonstrating superior performance. However, the architecture introduced is computationally expensive since the transformer architecture used scales quadratically with the input data. Kyem et al. [5] used YOLOv8 and the Segment Anything Model (SAM) for segmentation in their PaveCap framework, utilizing SAM's zero-shot capability for generating binary masks. However, the method struggled with accurately segmenting mixed or overlapping pavement distresses. Owor et al. also introduced PaveSAM [6], a zero-shot segmentation model fine-tuned for pavement distresses using bounding box prompts, significantly reducing labeling costs and achieving strong performance. One problem with this model is its inability to segment fine-grained distresses in the pavement images.

Despite the significant progress achieved with deep learning models for pavement distress segmentation, several limitations remain. ne critical yet unresolved challenge is the accurate segmentation of tiny and small pavement cracks. Tiny cracks typically refer to pavement cracks narrower than 1 mm. These cracks are faint, often discontinuous, and extremely challenging to detect, particularly under varying environmental conditions [7, 8, 9]. Small cracks, slightly larger, range from about 1 mm to 3 mm in width [9] Although more visible than tiny cracks, small cracks still pose significant challenges for accurate segmentation due to their irregular shapes, textures, and subtle appearance in images. Identifying these small and tiny defects early enables preventive maintenance before they develop into more extensive damage. By intervening at this early stage, maintenance teams can prevent minor issues from escalating, thereby reducing repair costs and minimizing disruptions to traffic. To achieve these outcomes, it is essential to adopt advanced models capable of effectively handling both very small and larger cracks. However, achieving this goal is not without challenges, as many existing models struggle with multi-scale feature representation, which hinders their ability to effectively detect both small-scale and large-scale cracks [10]. In addition to this challenge is the lack of a comprehensive understanding of global context which often limits the model's ability to capture large-scale spatial relationships and distinguish between interconnected distresses and noise. This results in inconsistent segmentation of extensive distress patterns such as longitudinal and alligator cracks. Furthermore, many deep learning

models require high-resolution inputs to detect subtle crack features, significantly increasing memory usage and inference time [11]. This computational burden limits their suitability for real-time deployment and scalability for large-scale pavement monitoring systems. These above limitations highlight the pressing need for innovative approaches to enhance segmentation performance and address the shortcomings of existing methods.

To address the challenges of pavement distress segmentation, we propose Context-CrackNet, an encoder-decoder architecture built around two key innovations: the Region-Focused Enhancement Module (RFEM) and the Context-Aware Global Module (CAGM). The RFEM, embedded in the decoder pathway, prioritizes fine-grained features, enabling precise segmentation of small and tiny cracks. This ensures early detection of subtle distresses that traditional models often miss. The CAGM, positioned before the bottleneck, captures global context efficiently by integrating linear self-attention into its design. This allows the model to process high-resolution images and segment larger cracks, such as longitudinal and alligator cracks, without excessive computational costs.. The main contributions of our research has been outlined below:

- Proposed **Context-CrackNet**, a novel efficient architecture that integrates specialized modules designed for comprehensive crack detection at varying scales in high-resolution pavement images.
- Developed a **Region-Focused Enhancement Module** (RFEM) that employs targeted feature enhancement to capture fine-grained details of subtle cracks, enabling precise segmentation of small-scale pavement distress patterns.
- Introduced a **Context-Aware Global Module** (CAGM) that utilizes global contextual information to effectively identify and segment large-scale distress patterns while maintaining computational efficiency across high-resolution images.
- Trained and evaluated our proposed architecture on 10 publicly available crack datasets alongside several existing state-of-the-art segmentation models. Our proposed architecture consistently outperformed these models, achieving state-of-the-art performance across all benchmarks.

2. Related Works

Early attempts at pavement crack detection primarily relied on low-level image properties such as gradient, brightness, shape, and texture, as well as pixel intensity variance, edge orientation, and local binary patterns. Classic edge detection algorithms such as Sobel and Canny [12], Gabor filter-based methods [13, 14, 15], Prewitt [16], Roberts Cross [16], and Laplacian of Gaussian [17] identified crack characteristics by examining intensity variations and local directional patterns. Some researchers also employed threshold-based strategies such as Adaptive and Localized Thresholding [18, 19, 20], Triple-Thresholds Approach [21], Otsu thresholding [22, 23, 24] and wavelet transformations [25] to isolate cracks from background textures. Building on these foundations, early machine learning approaches [26, 27, 28, 29, 30, 31, 32, 33] framed crack extraction as a classification problem, distinguishing between crack and non-crack pixels using hand-engineered features.

However, these traditional methods often struggled with generalization [34]. Differences between training and testing datasets commonly led to performance degradation, and detecting tiny cracks proved particularly challenging. Moreover, the reliance on handcrafted features made these methods sensitive to environmental variations and morphological differences in cracks [35]. Noise and other forms of interference further undermined their stability and applicability.

With the rapid advancement of deep learning and its application in pavement assessment [36, 37], researchers turned to Convolutional Neural Networks (CNNs) [38] to develop more robust solutions. CNNs excel at feature extraction, inspiring the creation of models for crack image classification, crack detection and crack segmentation. For instance, Li et al. [39] proposed a CNN-based method to classify pavement patches into five categories using 3D pavement images, demonstrating high accuracy in distinguishing between the various crack types. Zhang et al. [40] developed a deep convolutional neural network (ConvNet) for pavement crack detection, learning features directly from raw image patches and outperforming traditional hand-crafted approaches. Subsequently, Liu et al. [41] proposed DeepCrack, a deep hierarchical CNN for pixel-wise crack segmentation, incorporating multi-scale feature fusion, deeply-supervised nets, and guided filtering learning methods that steadily improved segmentation performance.

Despite these advancements, significant challenges persisted. Many state-of-the-art methods emphasized performance metrics but gave limited atten-

tion to extracting subtle, tiny crack features. Additionally, as models became more complex, their computational and memory requirements increased, hindering deployment on resource-limited devices. Recognizing these issues, researchers began exploring lightweight model architectures capable of balancing detection accuracy with efficiency.

Li et al. [42] proposed CarNet, a lightweight encoder-decoder achieving an ODS F-score of 0.514 on Sun520 with an inference speed of 104 FPS, balancing performance and efficiency. Similarly, Zhou et al. [43] introduced LightCrackNet, a lightweight crack detection model designed to optimize performance and efficiency. The model utilizes Split Exchange Convolution (SEConv) and Multi-Scale Feature Exchange (MSFE) modules, achieving an F1-score of 0.867 DeepCrack dataset with only 1.3M parameters and 8 GFLOPs.

Nevertheless, significant hurdles remain in achieving high-performance, efficient crack detection. Although CNN-based methods excel at extracting local features, they struggle to aggregate global context when used alone, limiting their ability to identify tiny or small cracks. Traditional self-attention transformer models offer a promising solution for capturing global relationships, but their quadratic scaling with input data makes them computationally intensive and impractical for certain applications. Linear self-attention transformers [44, 45, 46] when used with CNNs address this challenge by reducing computational complexity, enabling efficient integration of global and local cues while focusing on extracting minute crack features—a key objective in the field. Some researchers have applied linear self-attention modules in different domains. For instance, Fang et al. [47] proposed CLFormer, a lightweight transformer combining convolutional embedding and linear self-attention (LSA) for bearing fault diagnosis. It achieves strong robustness and high accuracy under noise and limited data, with only 4.88K parameters and 0.12 MFLOPs. Guo et al. [48] introduced External Attention, a lightweight mechanism with linear complexity using two learnable memory layers, enhancing generalization and efficiency across visual tasks like segmentation and detection.

3. Method

3.1. Problem Structure and Overview

Detecting small and tiny cracks in pavement images is particularly challenging due to their subtle features, irregular shapes, and the presence of

noise such as shadows and debris. Existing deep learning models often struggle with capturing these fine-grained details, underscoring the need for more precise and efficient segmentation approaches. The problem definition has been formulated mathematically below.

Let $I \in \mathbb{R}^{H\times W\times C}$ represent a pavement image, where H, W, and C denote the height, width, and number of channels, respectively. The goal is to predict a segmentation map $\hat{S} \in \mathbb{R}^{H\times W\times K}$, where K represents the number of classes, including the background. Mathematically, this can be expressed as learning a mapping function $f : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W\times K}$, such that:

$$\hat{S} = f(I; \theta), \tag{1}$$

where θ denotes the learnable parameters of the segmentation model. The task involves accurately localizing and classifying pavement distresses of varying scales, shapes, and textures.

The proposed architecture, *Context-CrackNet*, addresses these challenges by introducing two novel components: the Context-Aware Global Module (CAGM) and the Region-Focused Enhancement Module (RFEM). The CAGM ensures efficient global context modeling, enabling the network to capture long-range dependencies and large-scale spatial relationships. The RFEM enhances the network's ability to focus on fine-grained details, ensuring precise segmentation of small and subtle distresses. Together, these components form a robust encoder-decoder framework optimized for pavement distress segmentation.

3.2. Overall Framework

The overall structure of *Context-CrackNet* integrates global and local feature refinement seamlessly, providing a balanced approach to segmentation. The network adopts an encoder-decoder structure, where the encoder extracts hierarchical features from the input image, the bottleneck incorporates global attention mechanisms, and the decoder reconstructs the segmentation map using refined skip connections.

The encoder is based on a ResNet50 backbone, which extracts features at multiple levels of abstraction. For an input image I, the encoder produces a sequence of feature maps:

$$\Phi_{\text{enc}}(I) = \{F_0, F_1, F_2, F_3, F_4\}, \tag{2}$$

where $F_0 \in \mathbb{R}^{H/2\times W/2\times 64}$ represents low-level spatial features, and $F_4 \in \mathbb{R}^{H/32\times W/32\times 2048}$ captures high-level semantic information. These feature

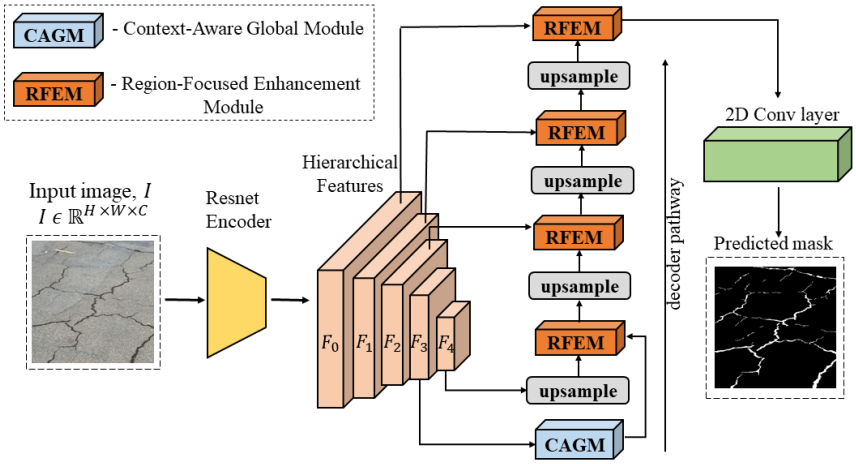

Figure 1: Overall Architecture of *Context-CrackNet*: The proposed framework adopts an encoder-decoder structure with two novel components: the Context-Aware Global Module (CAGM) and the Region-Focused Enhancement Module (RFEM). The ResNet-based encoder extracts hierarchical features $\{F_0, F_1, F_2, F_3, F_4\}$, where F_3 is processed by the CAGM to model global contextual relationships and generate the contextualized feature map. The decoder pathway integrates RFEMs at each stage, which refine the skip connections between encoder features and upsampled decoder outputs. This refinement enables effective feature modulation for precise segmentation. Finally, the decoder outputs the predicted segmentation mask $\hat{S}$, capturing fine-grained pavement distress details.

maps progressively encode spatial and contextual details, forming the foundation for subsequent processing stages.

At the bottleneck, the feature map F_3, which encapsulates rich semantic information, is passed through the Context-Aware Global Module (CAGM). The CAGM uses a linear self-attention mechanism to model long-range dependencies efficiently, reducing the computational complexity typically associated with traditional self-attention. This produces an enhanced feature map F_{CAGM}, expressed mathematically as:

$$F_{\text{CAGM}} = f_{\text{CAGM}}(F_3), \tag{3}$$

where f_{CAGM} represents the operations within the module.

In the decoder pathway, the Region-Focused Enhancement Module (RFEM) plays a critical role in refining the skip connections between the encoder and

decoder. For each decoder stage l, the RFEM processes the corresponding encoder feature map $F_{e,l}$ and the upsampled feature map $F_{d,l+1}$ from the previous decoder stage. The refined feature map $F_{\text{RFEM},l}$ is computed as:

$$F_{\text{RFEM},l} = f_{\text{RFEM}}(F_{e,l}, F_{d,l+1}), \tag{4}$$

where f_{RFEM} represents the attention mechanism used to focus on the most relevant spatial regions. This refinement ensures that critical features are emphasized while irrelevant activations are suppressed.

The decoder reconstructs the segmentation map by iteratively combining the refined features from the RFEM with the upsampled feature maps. Starting with the output of the CAGM, the decoder applies a series of upsampling and refinement operations to produce the final segmentation map:

$$\hat{S} = f_{\text{decoder}}(F_{\text{RFEM}}), \tag{5}$$

where f_{decoder} represents the decoding operations, including upsampling, concatenation, and convolution.

This framework effectively addresses the challenges associated with multi-scale feature representation and computational efficiency. By combining the strengths of the CAGM and RFEM, *Context-CrackNet* achieves a balance between global context understanding and fine-grained detail enhancement, enabling robust and accurate pavement distress segmentation. The subsequent sections goes deeper into the mathematical details and implementation of the CAGM and RFEM modules.

3.3. Context-Aware Global Module (CAGM)

The **Context-Aware Global Module (CAGM)** addresses the challenge of capturing long-range dependencies and global contextual relationships in the feature map F_3. This capability is crucial for accurately segmenting large-scale pavement distresses, such as longitudinal and alligator cracks, which require an understanding of spatial relationships across distant regions. To achieve this, the CAGM employs a linear self-attention mechanism, reducing the quadratic complexity of traditional self-attention to linear, thereby enabling efficient processing of high-resolution images.

Let $F_3 \in \mathbb{R}^{B\times C\times H\times W}$ represent the input feature map at the bottleneck stage, where B is the batch size, C is the number of channels, and H, W are the spatial dimensions. The first step in the CAGM is to transform the

Algorithm 1 Context-CrackNet Framework

Input image $I \in \mathbb{R}^{H \times W \times C}$
Predicted segmentation mask $\hat{S} \in \mathbb{R}^{H \times W \times K}$
Stage 1: Encoder Pathway
$\{F_0, F_1, F_2, F_3, F_4\} \leftarrow \text{ResNetEncoder}(I)$ ▷ Extract features
Stage 2: Bottleneck Processing
$F_{\text{CAGM}} \leftarrow \text{CAGM}(F_3)$ ▷ Global context modeling
Stage 3: Decoder Pathway
Initialize $D_4 \leftarrow F_{\text{CAGM}}$ $l \in \{3, 2, 1, 0\}$
$D_{\text{up}} \leftarrow \text{Upsample}(D_{l+1})$ ▷ $2\times$ spatial size
$D_l \leftarrow \text{RFEM}(F_l, D_{\text{up}})$ ▷ Feature refinement
Stage 4: Final Prediction
$\hat{S} \leftarrow \text{Conv2D}(D_0)$ ▷ K-class prediction
$\hat{S}$

spatial dimensions H and W into a sequence of length $N = H \times W$. This can be mathematically expressed as:

$$X_b = \{F_3[b, :, i, j] \mid i \in \{1, \ldots, H\}, j \in \{1, \ldots, W\}\}, \quad X \in \mathbb{R}^{B \times N \times C}, \quad (6)$$

where X_b is the reshaped feature map for the b-th sample in the batch, and X concatenates all spatial positions into a sequence.

The sequence X is projected into query (Q), key (K), and value (V) spaces using learned linear transformations:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \quad (7)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{C \times d_k}$ are learnable weight matrices, and d_k is the dimensionality of the query and key vectors.

To reduce the computational cost, the key and value matrices are projected into lower-dimensional spaces:

$$K_{\text{proj}} = KE, \quad V_{\text{proj}} = VF, \quad (8)$$

where $E, F \in \mathbb{R}^{N \times k}$ are learnable projection matrices, and $k \ll N$ is the reduced dimension of the projected key and value spaces.

The attention weights are computed as:

$$A = \text{softmax}\left(\frac{QK_{\text{proj}}^{\top}}{\sqrt{d_k}}\right), \quad A \in \mathbb{R}^{B \times N \times k}, \quad (9)$$

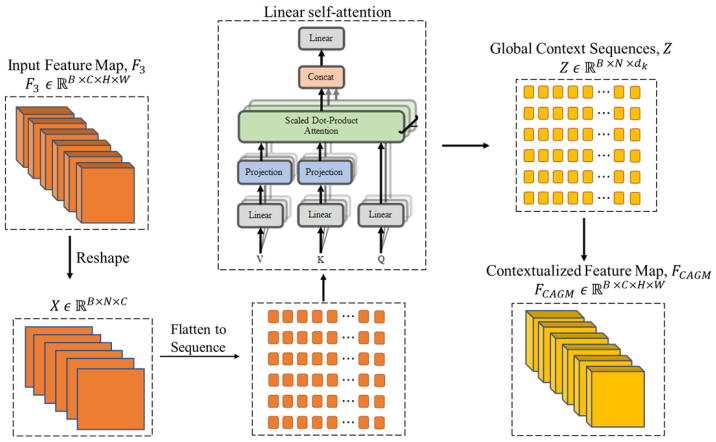

Figure 2: Context-Aware Global Module (CAGM): The module processes the input feature map $F_3 \in \mathbb{R}^{B\times C\times H\times W}$, reshaping it into a sequence $X \in \mathbb{R}^{B\times N\times C}$, where $N = H \times W$. Using a **Linear Self-Attention Mechanism**, query (Q), key (K), and value (V) projections generate **Global Context Sequences** ($Z \in \mathbb{R}^{B\times N\times d_k}$). These sequences are reconstructed into the **Contextualized Feature Map** ($F_{\text{CAGM}} \in \mathbb{R}^{B\times C\times H\times W}$), embedding global dependencies efficiently.

where softmax ensures the weights A sum to 1 across the key dimension for each query.

The weighted output is then computed by aggregating the projected values:

$$Z = AV_{\text{proj}}, \quad Z \in \mathbb{R}^{B\times N\times d_k}, \tag{10}$$

Finally, the output sequence Z is mapped back to the original channel dimension and reconstructed into its spatial structure:

$$F_{\text{CAGM}}[b, :, i, j] = Z[b, n]W_O, \quad n = (i-1)\times W + j, \tag{11}$$

where $W_O \in \mathbb{R}^{d_k\times C}$ is a learnable weight matrix, and $F_{\text{CAGM}} \in \mathbb{R}^{B\times C\times H\times W}$ is the enhanced feature map.

By explicitly modeling global relationships across spatial regions, the CAGM integrates information from distant parts of the pavement image, enabling the network to detect and segment large-scale cracks and patterns. Its efficient linear self-attention mechanism ensures scalability, making it suitable for high-resolution images while maintaining computational feasibility.

Rationale for Using Linear Self-Attention. Traditional self-attention has a quadratic complexity $\mathcal{O}(N^2)$, which limits its scalability to high-resolu-

Algorithm 2 Context-Aware Global Module (CAGM)

Require:
Input feature map $F_3 \in \mathbb{R}^{B\times C\times H\times W}$
Ensure:
Contextualized feature map $F_{\text{CAGM}} \in \mathbb{R}^{B\times C\times H\times W}$
Feature Reshaping:
$X \leftarrow \text{Reshape}(F_3)$ ▷ $X \in \mathbb{R}^{B\times N\times C}$, where $N = H \times W$
Linear Self-Attention:
$Q \leftarrow \text{Linear}(X)$ ▷ Query projection
$K \leftarrow \text{Linear}(X)$ ▷ Key projection
$V \leftarrow \text{Linear}(X)$ ▷ Value projection
$A \leftarrow \text{SDP-Attention}(Q, K, V)$ ▷ Scaled dot-product attention
$Z \leftarrow \text{Concat}[A, X]$ ▷ $Z \in \mathbb{R}^{B\times N\times d_k}$
$Z \leftarrow \text{Linear}(Z)$ ▷ Final projection
Global Context Reconstruction:
$F_{\text{CAGM}} \leftarrow \text{Reshape}(Z, [B, C, H, W])$ ▷ Restore spatial dimensions
return F_{CAGM}
Note: SDP-Attention computes $\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$

tion images. In contrast, the linear self-attention used in the CAGM reduces this to $\mathcal{O}(N \cdot k)$ by projecting keys and values into a lower-dimensional space, enabling efficient global context modeling. This is particularly important for pavement images, where cracks can span large, non-contiguous regions. The CAGM allows each spatial location to incorporate information from the entire image, improving segmentation of extensive or fragmented crack patterns. Positioned at the bottleneck, it complements the RFEM by enriching features with global cues before fine-grained refinement in the decoder.

3.4. Region-Focused Enhancement Module (RFEM)

The **Region-Focused Enhancement Module (RFEM)** refines the skip connections between the encoder and decoder to enhance the segmentation of fine-grained details such as small and subtle pavement cracks. By dynamically modulating encoder features using spatial context from the decoder, the RFEM ensures that relevant features are emphasized, while irrelevant ones are suppressed.

Let $F_{e,l} \in \mathbb{R}^{B\times C_e\times H_e\times W_e}$ denote the encoder feature map at level l, and

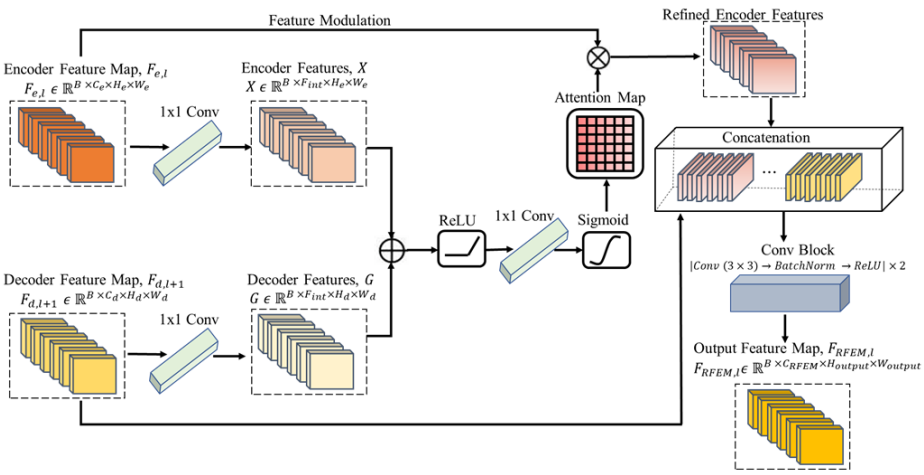

Figure 3: Region-Focused Enhancement Module (RFEM): The module refines encoder features $F_{e,l} \in \mathbb{R}^{B\times C_e\times H_e\times W_e}$ and decoder features $F_{d,l+1} \in \mathbb{R}^{B\times C_d\times H_d\times W_d}$ by transforming them into intermediate features X and G, respectively. These are fused and processed through a ReLU and sigmoid to generate an attention map. The attention-refined encoder features are then concatenated with decoder features and passed through a convolutional block, yielding $F_{\text{RFEM},l}$.

$F_{d,l+1} \in \mathbb{R}^{B\times C_d\times H_d\times W_d}$ the upsampled decoder feature map from the previous stage. To align the features for fusion, both are projected to a shared space:

$$G = F_{d,l+1} * W_g + b_g, \quad X = F_{e,l} * W_x + b_x, \tag{12}$$

where W_g, W_x and b_g, b_x are learnable parameters for 1×1 convolutions.

These projected features are fused via element-wise addition, capturing complementary spatial cues. A ReLU activation is then applied to retain positive activations and introduce non-linearity:

$$Y = \max(0, G + X), \tag{13}$$

To localize salient regions, an attention map Ψ is derived by passing the fused features through a convolution followed by a sigmoid activation:

$$\Psi = \frac{1}{1 + \exp\left(-\left(Y * W_\psi + b_\psi\right)\right)}, \tag{14}$$

where W_ψ and b_ψ are learnable parameters. The sigmoid scales attention weights to [0,1], guiding selective emphasis.

The attention map modulates the encoder features via element-wise multiplication:

$$F_{\text{refined}} = \Psi \odot F_{e,l}, \tag{15}$$

where $\odot$ denotes element-wise multiplication, resulting in spatially weighted encoder features.

These refined features are then concatenated with decoder features along the channel dimension:

$$F_{\text{concat}}[b, c, i, j] = \begin{cases} F_{\text{refined}}[b, c, i, j], & \text{if } c < C_{\text{refined}}, \\ F_{d,l+1}[b, c - C_{\text{refined}}, i, j], & \text{otherwise,} \end{cases} \tag{16}$$

ensuring that both attention-enhanced and high-level decoder context are retained.

Finally, a convolutional refinement block $\gamma(\cdot)$, typically composed of two Conv-BatchNorm-ReLU layers, produces the final output:

$$F_{\text{RFEM},l} = \gamma(F_{\text{concat}}), \tag{17}$$

By focusing on critical local features, RFEM strengthens the model's ability to detect tiny cracks while complementing the broader context captured by CAGM.

3.5. Loss Functions

The proposed framework uses tailored loss functions to optimize the segmentation task across both binary and multi-class scenarios, ensuring accurate prediction of pavement distress patterns. These loss functions are carefully designed to balance class contributions, address class imbalance, and effectively capture fine-grained details.

3.5.1. Binary Segmentation Loss

For binary segmentation tasks, where the goal is to classify each pixel as either belonging to a crack (1) or not (0), we employ a combination of the Binary Cross Entropy (BCE) loss and the Dice loss. The combined loss is formulated as:

$$\mathcal{L}_{\text{binary}} = \alpha\, \mathcal{L}_{\text{BCE}} + \beta\, \mathcal{L}_{\text{Dice}}, \tag{18}$$

where α and β are weights that control the contribution of each term.

Algorithm 3 Region-Focused Enhancement Module (RFEM)

Require:
Encoder feature map $F_{e,l} \in \mathbb{R}^{B \times C_e \times H_e \times W_e}$
Decoder feature map $F_{d,l+1} \in \mathbb{R}^{B \times C_d \times H_d \times W_d}$
Ensure:
Refined feature map $F_{\text{RFEM},l} \in \mathbb{R}^{B \times C_{\text{RFEM}} \times H_{\text{output}} \times W_{\text{output}}}$
Feature Transformation:
$X \leftarrow \text{Conv}_{1\times1}(F_{e,l})$ ▷ Transform to $\mathbb{R}^{B \times F_{\text{int}} \times H_e \times W_e}$
$G \leftarrow \text{Conv}_{1\times1}(F_{d,l+1})$ ▷ Transform to $\mathbb{R}^{B \times F_{\text{int}} \times H_d \times W_d}$
Attention Map Generation:
$Y \leftarrow \text{ReLU}(G + X)$ ▷ Element-wise addition
$\Psi \leftarrow \sigma(\text{Conv}_{1\times1}(Y))$ ▷ σ: Sigmoid activation
Feature Modulation:
$F_{\text{refined}} \leftarrow \Psi \otimes F_{e,l}$ ▷ Channel-wise multiplication
Feature Fusion:
$F_{\text{concat}} \leftarrow \text{Concat}([F_{\text{refined}}, F_{d,l+1}])$
Conv Block Refinement:
for $i \leftarrow 1$ **to** 2 **do**
    $F_{\text{RFEM},l} \leftarrow \text{ReLU}(\text{BatchNorm}(\text{Conv}_{3\times3}(F_{\text{concat}})))$
end for
    return $F_{\text{RFEM},l}$

1. Binary Dice Loss. The Dice loss, designed to handle imbalances in pixel classes, measures the overlap between the predicted segmentation map $\hat{S}$ and the ground truth S.

Let $\hat{S} \in [0,1]^{H\times W}$ and $S \in \{0,1\}^{H\times W}$ denote the predicted and ground truth maps, respectively. The Dice loss is computed as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} \hat{S}_i S_i + \epsilon}{\sum_{i=1}^{N} \hat{S}_i + \sum_{i=1}^{N} S_i + \epsilon}, \tag{19}$$

where ϵ is a smoothing constant to prevent division by zero, and $N = H \times W$ represents the total number of pixels. The Dice loss encourages the model to maximize the overlap between $\hat{S}$ and S, ensuring robust segmentation of small and subtle cracks.

2. Binary Cross Entropy Loss. The BCE loss penalizes the deviation between predicted probabilities and ground truth labels. It is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[S_i \log(\hat{S}_i) + (1 - S_i)\log(1 - \hat{S}_i)\right], \tag{20}$$

This term provides pixel-wise supervision, complementing the Dice loss by ensuring accurate classification even in cases of severe class imbalance.

3.5.2. Multi-Class Segmentation Loss

For multi-class segmentation tasks, where the pavement image contains multiple types of distresses, we adopt a combined loss function comprising the Cross-Entropy (CE) loss and a multi-class Dice loss:

$$\mathcal{L}_{\text{multi-class}} = \gamma\,\mathcal{L}_{\text{CE}} + \delta\,\mathcal{L}_{\text{Dice}}, \tag{21}$$

where γ and δ are weighting factors.

1. Multi-Class Dice Loss. The Multi-Class Dice Loss extends the principles of the Binary Dice Loss to multi-class segmentation tasks, ensuring fair optimization for each class, including the background. By individually evaluating the overlap between the predicted segmentation map $\hat{S}_k$ and the ground truth map S_k for each class k, it addresses class imbalances and promotes accurate segmentation across all categories.

Let $\hat{S}_k$ and S_k represent the predicted and ground truth maps for class k, respectively, where $k \in \{1, \ldots, K\}$. The Dice loss for class k is defined as:

$$\mathcal{L}_{\text{Dice},k} = 1 - \frac{2\sum_{i=1}^{N}\hat{S}_{k,i}S_{k,i} + \epsilon}{\sum_{i=1}^{N}\hat{S}_{k,i} + \sum_{i=1}^{N}S_{k,i} + \epsilon}, \tag{22}$$

where N denotes the total number of pixels and ϵ is a small constant to avoid division by zero.

The total multi-class Dice loss is calculated as the average Dice loss across all K classes:

$$\mathcal{L}_{\text{Dice}} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\text{Dice},k}, \tag{23}$$

This formulation encourages precise segmentation across all classes, ensuring small or underrepresented categories are effectively captured.

2. Cross-Entropy Loss. The CE loss measures the pixel-wise classification error, weighted by class frequencies to handle imbalance.

Let ω_k denote the weight for class k, derived from the inverse class frequency. The CE loss is defined as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\omega_k S_{k,i}\log(\hat{S}_{k,i}), \tag{24}$$

where ω_k ensures that the model does not bias toward dominant classes.

4. Experiments

To validate the effectiveness of the proposed *Context-CrackNet*, we conduct comprehensive experiments designed to evaluate its performance on pavement distress segmentation tasks. This section details the datasets used, the implementation specifics of *Context-CrackNet*, and the experimental setup. Furthermore, ablation studies are performed to assess the contribution of each module and design choice to the overall performance. Finally, we compare *Context-CrackNet* with state-of-the-art methods to highlight its advantages in addressing the challenges of fine-grained and multi-scale pavement distress detection.

4.1. Dataset

To train and evaluate the proposed *Context-CrackNet*, we utilize 10 publicly available binary crack datasets: CFD [49], Crack500 [50], CrackTree200 [51], DeepCrack [41], Eugen Miller [52], Forest [53], GAPs [54], Rissbilder [55], Sylvie [56], and Volker [55]. These datasets comprehensively cover diverse crack detection scenarios, including road cracks, concrete cracks in tunnels, and wall cracks. This diversity demonstrates the robustness of *Context-CrackNet* beyond pavement applications, highlighting its potential to handle various types of cracks across different construction materials and structures.

The datasets exhibit a significant class imbalance, with approximately 97.2% of pixels belonging to the background class and only 2.8% corresponding to the crack class. This imbalance reflects the real-world challenges of identifying subtle cracks against vast background regions.

All the images across the datasts have a resolution of 448×448 pixels. The datasets were then split into training and testing sets with an 80:20 ratio, ensuring a balanced distribution of samples across both sets.

Table 1: Datasets and their corresponding crack types

Dataset name	**Crack type**
CFD	Road crack
CRACK500	Road crack
CrackTree200	Road crack
DeepCrack	Road crack
Forest	Road crack
GAPs	Road crack
Sylvie	Road crack
Rissbilder	Wall crack
Volker	Wall crack
Eugen Miller	Concrete crack on Tunnels

To further enhance the diversity of the training data and improve the model's generalization capabilities, a series of data augmentation techniques were employed. These augmentations included spatial transformations such as horizontal and vertical flips, random rotations by 90°, and shift-scale-rotate operations, which varied the spatial properties of the images while maintaining their structural integrity. Additionally, pixel-level augmentations such as Gaussian noise and color jittering were applied to introduce variations in brightness, contrast, saturation, and hue, simulating real-world variations in lighting and camera conditions.

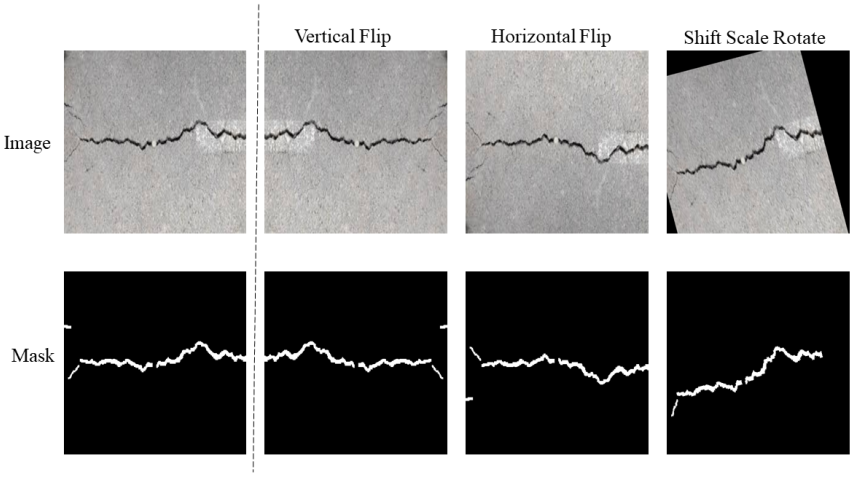

Figure 4: Examples of data augmentation techniques applied to crack images and their corresponding masks in the DeepCrack dataset. Augmentations include vertical flip, horizontal flip, and shift-scale-rotate, showcasing the spatial transformations employed to enhance diversity and robustness in the training dataset. The top row illustrates augmented images, while the bottom row presents their respective masks.

For preprocessing, the images were normalized using the mean and standard deviation values of the ResNet backbone: $\mu(0.485, 0.456, 0.406)$ and $\sigma(0.229, 0.224, 0.225)$, respectively. This normalization step ensures compatibility with the pre-trained ResNet model used in the encoder.

By employing a diverse dataset that encompasses road, wall, and concrete cracks, the proposed framework is equipped to handle the complexities of real-world crack detection scenarios, addressing challenges such as class imbalance, varying scales, and noise effectively.

4.1.1. Custom Dataset

4.2. Implementation details

4.2.1. Training settings

The proposed *Context-CrackNet* was implemented using the PyTorch framework. The AdamW [57] optimizer was used for training, with a weight decay of 1×10^{-5} to prevent overfitting. The initial learning rate was set to 1×10^{-4} , and a using an adaptive learning rate scheduler was applied to adjust the learning rate dynamically. This scheduler reduced the learning rate by a factor of 0.5 after 5 epochs of no improvement in validation loss.

All models, including Context-CrackNet and the baseline models (U-Net, U-Net++, DeepLabV3, DeepLabV3+, FPN, PSPNet, LinkNet, MAnet, and PAN), were trained separately on each of the ten datasets. Each dataset was randomly split into training and validation sets, ensuring a consistent evaluation setup for every model. After training, each model was evaluated on the corresponding validation set of each dataset to measure performance.

The training was performed with a batch size of 32 over a total of 1000 epochs. All experiments were conducted on an NVIDIA A40 GPU with 48GB of memory, providing sufficient computational power to efficiently handle high-resolution images. Table 1 summarizes the training configurations used for all experiments.

4.2.2. Evaluation metrics

To comprehensively evaluate the performance of the proposed *Context-CrackNet* on pavement distress segmentation tasks, we employed the following segmentation metrics: Intersection over Union (IoU) score, Dice score, Precision, Recall, and F1 score. These metrics provide a holistic assessment of the model's ability to accurately segment fine-grained and multi-scale pavement cracks, balancing considerations of overlap, correctness, and completeness.

Mean Intersection over Union (mIoU). The mean Intersection over Union (mIoU) evaluates the average overlap between the predicted segmentation map $\hat{S}_k$ and the ground truth S_k across all classes k. For a single class k, the IoU is defined as:

$$\mathrm{IoU}_k = \frac{|\hat{S}_k \cap S_k|}{|\hat{S}_k \cup S_k|} = \frac{\sum_{i=1}^{N} \hat{S}_{k,i} S_{k,i}}{\sum_{i=1}^{N} \left(\hat{S}_{k,i} + S_{k,i} - \hat{S}_{k,i} S_{k,i} \right)}, \tag{25}$$

where N denotes the total number of pixels, and $\hat{S}_{k,i}, S_{k,i} \in \{0, 1\}$ represent the predicted and ground truth labels for pixel i in class k. The mIoU is then computed as the average IoU across all K classes:

$$\mathrm{mIoU} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{IoU}_k, \tag{26}$$

This metric provides a comprehensive assessment of segmentation performance by considering the overlap for all classes and averaging them to yield a single performance score.

Dice Score. The Dice score, also known as the Sørensen–Dice coefficient, quantifies the overlap between the predicted and ground truth segmentation maps. It is defined as:

$$\text{Dice} = \frac{2|\hat{S} \cap S|}{|\hat{S}| + |S|} = \frac{2\sum_{i=1}^{N} \hat{S}_i S_i}{\sum_{i=1}^{N} \hat{S}_i + \sum_{i=1}^{N} S_i}. \tag{27}$$

Dice score emphasizes the correct segmentation of smaller regions, making it particularly useful for evaluating fine-grained crack details.

Precision. Precision measures the proportion of correctly identified crack pixels to the total predicted crack pixels. It is defined as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}. \tag{28}$$

Recall. Recall quantifies the ability of the model to detect all crack pixels in the ground truth. It is defined as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}. \tag{29}$$

F1 Score. The F1 score provides a harmonic mean of Precision and Recall, balancing their trade-offs. It is expressed as:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{30}$$

These metrics collectively evaluate the model's segmentation performance, ensuring both spatial accuracy (IoU, Dice) and the balance between prediction correctness and completeness (Precision, Recall, F1).

Table 2: Model training settings.

Name	**Training setting**
Optimizer	AdamW
Learning Rate	1×10^{-4}
Batch Size	32
Weight Decay	1×10^{-5}
Number of Epochs	1000

4.2.3. Comparison with other methods

To evaluate the performance of the proposed *Context-CrackNet*, we compared it against state-of-the-art segmentation models, including U-Net [58], U-Net++ [59], DeepLabV3 [60], DeepLabV3+ [61], FPN [62], PSPNet [63], LinkNet [64], MAnet [65], and PAN [66]. All models were trained and evaluated on the same 10 crack detection datasets (see Section 4.1) under identical experimental conditions to ensure fairness. Each method used a ResNet50 encoder pre-trained on ImageNet, consistent with *Context-CrackNet*.

The datasets presented diverse challenges such as varying crack patterns, scales, and noise levels, providing a robust basis for comparison. Standardized training settings, including preprocessing, augmentations, and hyperparameters, ensured that performance differences reflected the strengths of the architectures rather than experimental inconsistencies.

5. Results and Discussion

This section presents the results of the proposed *Context-CrackNet* across multiple datasets. Both qualitative and quantitative evaluations are discussed, highlighting the model's performance in comparison with existing state-of-the-art methods.

5.1. Qualitative Analysis of Predictions

In this section, we analyze the qualitative results of *Context-CrackNet* compared to existing models, including MAnet, PSPNet, DeepLabV3+, and FPN. These results are evaluated across the ten diverse datasets containing different types of cracks, such as road cracks, wall cracks, and concrete cracks. The goal is to assess each model's ability to detect both prominent and tiny cracks, which are critical for reliable structural assessment.

Figure 5 shows segmentation results from various datasets. The first column displays the original crack images, followed by their ground truth masks. The subsequent columns show the predictions from *Context-CrackNet* and other models. Each row represents a dataset, showcasing the models' performance across different types of cracks.

The predictions demonstrate that *Context-CrackNet* performs consistently better in detecting fine, small, and subtle cracks compared to the other models. For instance, in the CRACK500 dataset, *Context-CrackNet* successfully identifies the small, interconnected cracks, which competing models often fail to detect. A similar trend is observed in the Rissbilder dataset, where

Context-CrackNet captures the thin wall cracks more accurately, while other models struggle with false positives or incomplete predictions.

5.1.1. Performance on Diverse Crack Types

The results emphasize *Context-CrackNet*'s adaptability to different crack types and contexts. In the DeepCrack dataset, characterized by dense and complex crack patterns, *Context-CrackNet* captures the overall structure of the cracks more effectively than models like PSPNet and DeepLabV3+, which tend to miss faint connections. Likewise, in the Eugen Miller dataset featuring tunnel cracks, *Context-CrackNet* produces cleaner and more detailed predictions, showing its robustness on surfaces with uniform textures.

5.1.2. Tiny Crack Detection and Generalization

A key strength of *Context-CrackNet* is its ability to detect tiny cracks that are often missed by other models. The red-marked areas in the predictions from competing models highlight their failure to consistently capture these smaller cracks. This reinforces the effectiveness of *Context-CrackNet*'s context-aware design, which allows it to focus on both small-scale details and larger crack patterns. This capability addresses the central challenge of detecting tiny cracks, which are crucial for identifying early signs of structural damage.

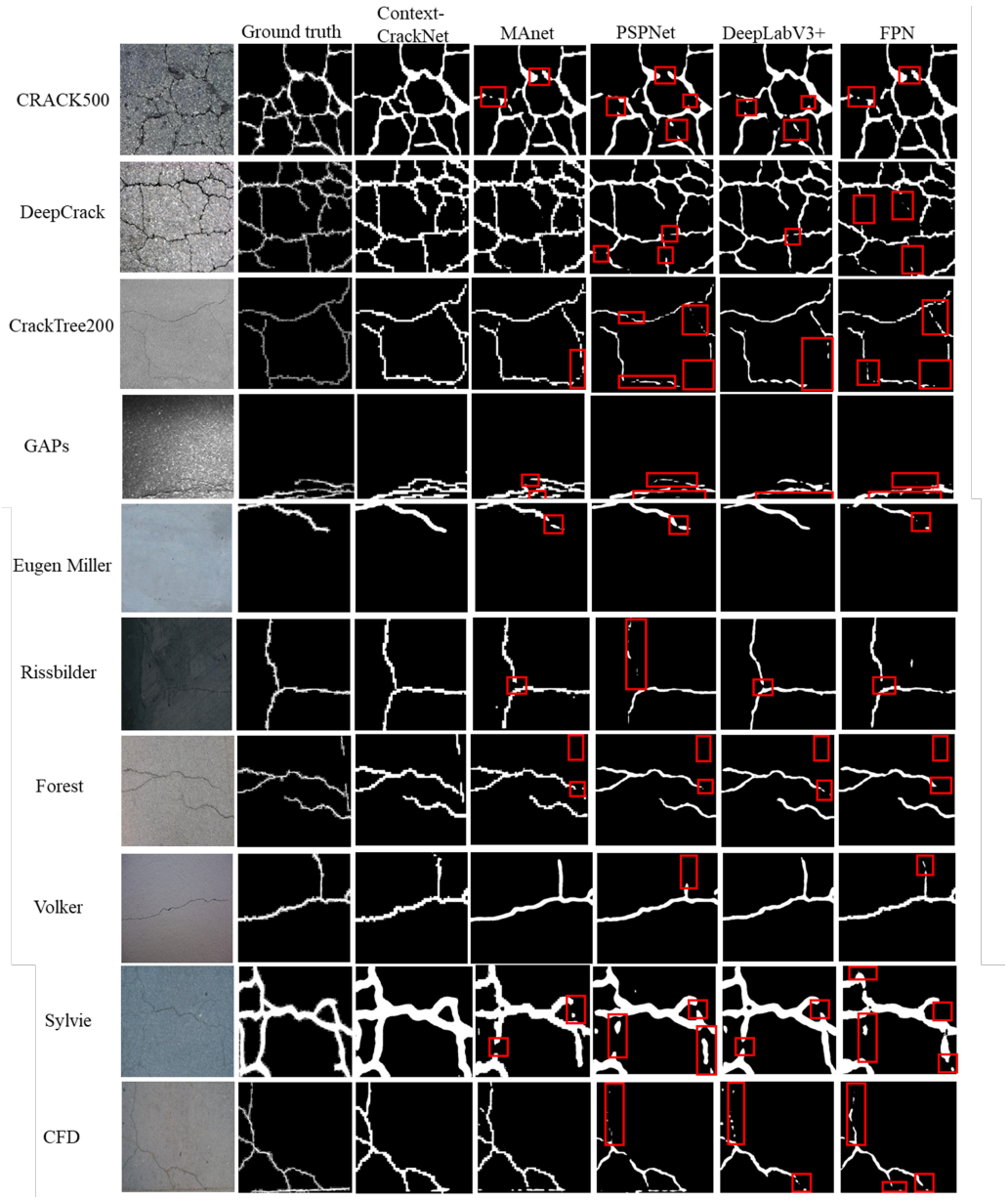

Figure 5: Visual comparison of crack detection results across various datasets. The first column displays the input images, followed by the ground truth masks in the second column. The third column shows the predictions of the proposed *Context-CrackNet*, while subsequent columns present predictions from comparison models including MAnet, PSPNet, DeepLabV3+, and FPN. Rows correspond to individual datasets (e.g., CRACK500, DeepCrack, CrackTree200, etc.). The red boxes highlight areas where comparison models fail to detect tiny cracks effectively, demonstrating the superior performance of *Context-CrackNet* in accurately capturing fine-grained crack details.

5.2. Quantitative Results

The quantitative results in Table 3 demonstrate that *Context-CrackNet* effectively addresses the challenge of detecting tiny and complex cracks, often missed by existing models. On the CFD dataset, it achieves the highest mIoU (**0.5668**) and Dice Score (**0.7235**), along with a recall of **0.8989**, showcasing its ability to capture subtle crack details where models like DeepLabV3 and FPN fall short.

In the CRACK500 dataset, which features diverse crack patterns, *Context-CrackNet* outperforms others with an mIoU of **0.6733** and Dice Score of **0.8046**, demonstrating robust generalization to varying pavement conditions. Similarly, on CrackTree200, characterized by sparse and narrow cracks, it achieves the highest recall (**0.9386**) and Dice Score (**0.6992**), proving its sensitivity to fine-grained cracks that models such as PAN and DeepLabV3Plus often miss.

The DeepCrack dataset further highlights Context-CrackNet's strengths, achieving an mIoU of **0.7401** and Dice Score of **0.8505**, validating its ability to detect tiny and small crack patterns with high precision. For non-road cracks in datasets like Eugen Miller and Rissbilder, *Context-CrackNet* shows strong adaptability with recalls of **0.9434** and **0.8469**, outperforming models that struggle with material and texture variations.

On datasets like GAPs and Forest, where irregular crack features dominate, *Context-CrackNet* consistently achieves superior metrics, confirming its effectiveness in challenging environments. Finally, on Sylvie and Volker, it maintains top performance with mIoUs of **0.6846** and **0.7668**, demonstrating its ability to handle varying complexities and environmental conditions. Figure 6 compares *Context-CrackNet* with other state-of-the-art models across the various crack datasets, highlighting Validation IoU, Dice Score, and Recall.

These results affirm that *Context-CrackNet* effectively addresses the limitations of existing models by reliably detecting tiny and complex cracks across diverse datasets, reinforcing its potential for real-world applications in crack detection and infrastructure monitoring.

5.3. Statistical Analysis

To begin our analysis, we first calculated the mean and standard deviation of the IoU and Dice Score across all datasets for each model, as

Table 3: Validation Results of *Context-CrackNet* and other models across all datasets. Each sub-table groups 4 metrics (mIoU, Dice, Recall, Precision) per dataset. Bold red values indicate the highest score in that metric for the dataset.

	CFD				CRACK500				CrackTree200				DeepCrack			
Model	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**
DeepLabV3	0.3194	0.4842	0.5065	0.4638	0.6420	0.7817	0.7612	0.8035	0.3174	0.4819	0.4305	0.5471	0.6813	0.8102	0.8246	0.7963
DeepLabV3Plus	0.3848	0.5558	0.5039	0.6195	0.6376	0.7784	0.7529	0.8060	0.2789	0.4361	0.3823	0.5077	0.6639	0.7977	0.7570	**0.8433**
FPN	0.4187	0.5902	0.5033	0.7134	0.6317	0.7740	0.7460	0.8047	0.2958	0.4566	0.3736	0.5869	0.6783	0.8081	0.7796	0.8387
Linknet	0.4664	0.6362	0.5951	0.6833	0.6450	0.7839	0.7855	0.7827	0.4807	0.6493	0.7066	0.6006	0.7047	0.8267	0.8402	0.8137
MAnet	0.5174	0.6819	0.6901	0.6740	0.6341	0.7757	0.7408	0.8142	0.4197	0.5913	0.6063	0.5770	0.7007	0.8238	0.8502	0.7991
PAN	0.3904	0.5616	0.4614	0.7173	0.6231	0.7674	0.7271	0.8132	0.2765	0.4333	0.3532	0.5602	0.6535	0.7902	0.7444	0.8421
PSPNet	0.3553	0.5244	0.4084	**0.7322**	0.6197	0.7649	0.7058	**0.8356**	0.2927	0.4528	0.3757	0.5697	0.6582	0.7936	0.7761	0.8120
Unet	0.1562	0.2703	0.8370	0.1611	0.6397	0.7800	0.7496	0.8133	0.4857	0.6539	0.7022	**0.6118**	0.7061	0.8275	0.8231	0.8320
UnetPlusPlus	0.5257	0.6891	0.6653	0.7147	0.6444	0.7835	0.7714	0.7960	0.4257	0.5972	0.6179	0.5778	0.7168	0.8349	0.8324	0.8374
Context-CrackNet (ours)	**0.5668**	**0.7235**	**0.8989**	0.6054	**0.6733**	**0.8046**	**0.7967**	0.8129	**0.5375**	**0.6992**	**0.9386**	0.5571	**0.7401**	**0.8505**	**0.9175**	0.7926

	Eugen Miller				Forest				GAPs			
Model	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**
DeepLabV3	0.6312	0.7739	0.8696	0.6971	0.4826	0.6510	0.6151	0.6915	0.3704	0.5405	0.4694	0.6393
DeepLabV3Plus	0.6051	0.7540	0.7717	0.7371	0.4616	0.6316	0.5473	**0.7465**	0.3153	0.4785	0.4142	0.5679
FPN	0.4763	0.6452	0.5366	0.8091	0.4356	0.6069	0.5352	0.7007	0.3071	0.4699	0.3729	0.6353
Linknet	0.6024	0.7519	0.7473	0.7565	0.4857	0.6538	0.6664	0.6417	0.4007	0.5721	0.5430	0.6050
MAnet	0.5887	0.7411	0.6777	**0.8176**	0.5170	0.6816	0.6650	0.6990	0.3832	0.5540	0.5196	0.5937
PAN	0.5794	0.7337	0.7198	0.7482	0.4579	0.6281	0.5605	0.7143	0.2758	0.4323	0.3200	0.6668
PSPNet	0.6058	0.7545	0.7311	0.7794	0.3848	0.5557	0.4687	0.6825	0.2898	0.4491	0.3518	0.6237
Unet	0.6323	0.7747	0.7728	0.7767	0.5390	0.7005	0.6723	0.7311	0.3837	0.5545	0.4809	0.6548
UnetPlusPlus	**0.7060**	**0.8277**	0.8689	0.7902	0.5457	0.7061	0.6896	0.7234	0.3927	0.5637	0.4894	**0.6669**
Context-CrackNet (ours)	0.6627	0.7971	**0.9434**	0.6901	**0.5699**	**0.7261**	**0.8758**	0.6201	**0.4743**	**0.6433**	**0.7838**	0.5456

	Rissbilder				Sylvie				Volker			
Model	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**
DeepLabV3	0.5965	0.7471	0.7442	0.7502	0.6642	0.7982	0.7377	0.8696	0.7209	0.8378	0.8280	0.8479
DeepLabV3Plus	0.5653	0.7221	0.7057	0.7395	0.6505	0.7882	0.6964	**0.9079**	0.6872	0.8146	0.8046	0.8249
FPN	0.5819	0.7356	0.7153	**0.7572**	0.5966	0.7474	0.6570	0.8666	0.7028	0.8254	0.7994	**0.8534**
Linknet	0.6068	0.7552	0.7632	0.7476	0.6449	0.7842	0.7481	0.8238	0.7233	0.8394	0.8841	0.7990
MAnet	0.6365	0.7777	0.8104	0.7477	0.6263	0.7702	0.6899	0.8718	0.7422	0.8520	0.8543	0.8497
PAN	0.5416	0.7024	0.6403	0.7783	0.6074	0.7558	0.6869	0.8401	0.6796	0.8092	0.7720	0.8504
PSPNet	0.5042	0.6700	0.6146	0.7381	0.5481	0.7081	0.5841	0.8989	0.6870	0.8144	0.7905	0.8400
Unet	0.6456	0.7845	0.8105	0.7602	0.6577	0.7935	0.7622	0.8275	0.7464	0.8548	0.8574	0.8523
UnetPlusPlus	**0.6568**	**0.7926**	0.8233	0.7644	0.6718	0.8037	0.7804	0.8284	0.7641	0.8663	0.8983	0.8365
Context-CrackNet (ours)	0.6553	0.7916	**0.8469**	0.7432	**0.6846**	**0.8128**	**0.8429**	0.7847	**0.7668**	**0.8680**	**0.9002**	0.8381

summarized in Table 4. Although these descriptive statistics provide an initial sense of overall performance variability, they do not by themselves clarify whether the observed differences are statistically significant.

Therefore, to rigorously evaluate whether the improvements achieved by Context-CrackNet were indeed meaningful relative to other baseline models, we employed the Wilcoxon signed-rank test, which determines if observed performance gains could simply stem from random variations.

Figure 7 presents the distribution of Validation IoU and Dice Scores across the evaluated segmentation frameworks, where the boxplots show that Context-CrackNet consistently attains higher median IoU and Dice values compared to other approaches. To substantiate these visual findings, we further computed Wilcoxon test p-values by contrasting the Validation IoU of Con-

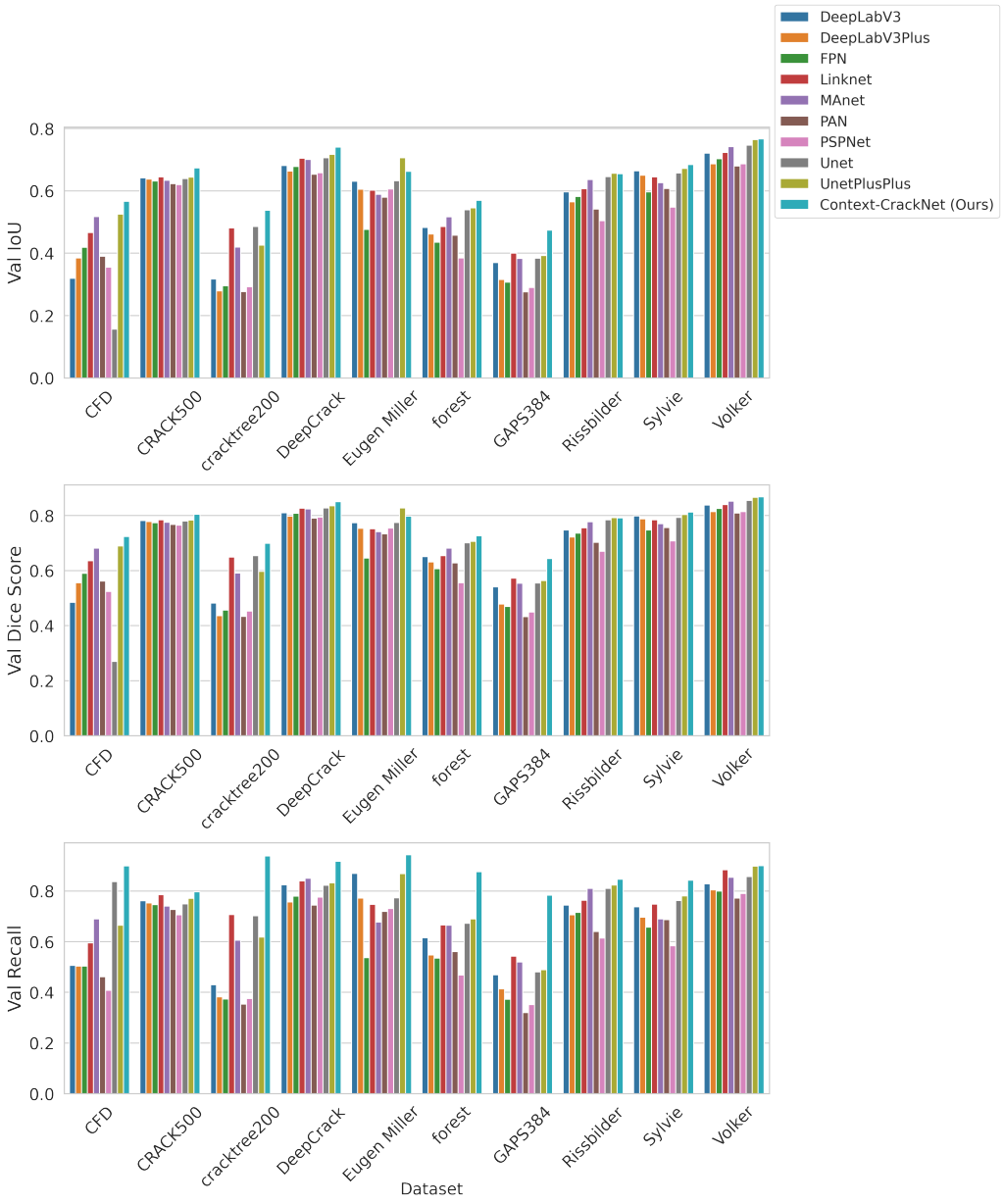

Figure 6: Performance comparison of the trained models across the different crack datasets. The top bar chart shows the Validation IoU, the middle bar chart shows the Validation Dice Score, and the bottom bar chart shows the Validation Recall. Higher bars indicate better performance.

Table 4: Mean ± standard deviation of the validation IoU and Dice score across the ten benchmark datasets.

Model	**IoU↑**	**Dice Score↑**
DeepLabV3	0.5426 ± 0.1556	0.6907 ± 0.1418
DeepLabV3Plus	0.5250 ± 0.1530	0.6757 ± 0.1398
FPN	0.5125 ± 0.1497	0.6665 ± 0.1358
Linknet	0.5761 ± 0.1148	0.7112 ± 0.1016
MAnet	0.5766 ± 0.1097	0.7149 ± 0.0987
PAN	0.5085 ± 0.1490	0.6578 ± 0.1375
PSPNet	0.4946 ± 0.1497	0.6490 ± 0.1379
Unet	0.5593 ± 0.1766	0.6979 ± 0.1687
UnetPlusPlus	0.6050 ± 0.1303	0.7386 ± 0.1161
Context-CrackNet (ours)	$\mathbf{0.6331 \pm 0.0945}$	$\mathbf{0.7718 \pm 0.0826}$

text-CrackNet with each baseline model (see Table 5). These results indicate that Context-CrackNet's improvements are statistically significant against all baselines except UnetPlusPlus, for which the difference, while substantial, did not meet the significance threshold ($p = 0.0645$).

Table 5: Wilcoxon Signed-Rank Test Results between Context-CrackNet and Baseline Models

Baseline Model	**p-value**
DeepLabV3	0.0020
DeepLabV3Plus	0.0020
FPN	0.0020
Linknet	0.0020
MAnet	0.0020
PAN	0.0020
PSPNet	0.0020
Unet	0.0020
UnetPlusPlus	0.0645

5.4. Attention Map Visualization from RFEM Modules

The attention maps in Figure 8 illustrate the regions of the input images that the RFEM modules in *Context-CrackNet* focuses on during segmenta-

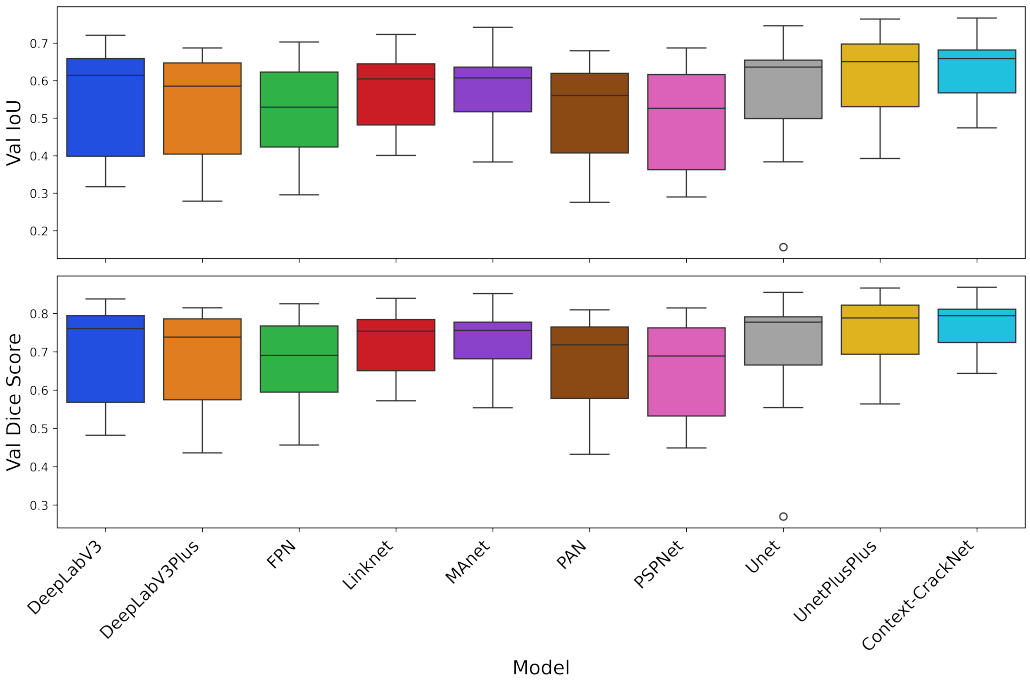

Figure 7: Distribution of Validation IoU and Val Dice Scores across Models

tion. These visualizations reveal that the model effectively identifies and emphasizes critical areas of distress across the different datasets. For example, in datasets like CRACK500 and DeepCrack, the attention maps distinctly capture intricate crack patterns, demonstrating the model's ability to localize fine-grained details. In complex cases like Rissbilder and Eugen Miller, the attention maps prioritize regions with subtle texture variations, ensuring accurate predictions even in challenging conditions. By highlighting relevant features, the attention maps provide interpretability to the model's decisions and validate its capability to generalize across datasets with varying characteristics. This insight is crucial for understanding how the model adapts to different pavement distress types.

5.5. Ablation studies

To thoroughly analyze the contributions of the RFEM and CAGM modules, we conducted ablation experiments by selectively enabling or disabling these modules in the proposed architecture. These experiments allow us to

quantify the individual and combined effects of RFEM and CAGM on segmentation performance. The results are summarized in Table 6.

Table 6: Ablation study results for RFEM and CAGM. Metrics include validation IoU, Dice Score, Precision, and Recall.

Configuration	**mIoU↑**	**Dice Score↑**	**Precision↑**	**Recall↑**
Baseline	0.4259	0.5929	0.5481	0.6691
RFEM Only	0.4355	0.6057	0.5439	0.6990
CAGM Only	0.4263	0.5954	0.5392	0.6827
RFEM + CAGM	**0.4743**	**0.6433**	**0.5456**	**0.7838**

Analysis and Discussion. The baseline model, without the RFEM and CAGM modules, achieved a validation IoU of 0.4259 and a Dice Score of 0.5929, demonstrating limited capability in capturing both local and global context. Adding RFEM alone improved the IoU to 0.4355 and the Dice Score to 0.6057, indicating that RFEM effectively enhances the model's ability to focus on critical local regions, especially fine-grained crack details. This is further reflected in the increase in recall from 0.6691 to 0.6990, as RFEM enables the model to identify more instances of distress.

When CAGM was included without RFEM, the IoU and Dice Score showed minimal improvements (0.4263 and 0.5954, respectively). While CAGM provides global context by capturing broader spatial dependencies, its contribution is less pronounced when local refinement (via RFEM) is absent. However, recall improved to 0.6827, suggesting that CAGM aids in generalizing to larger contextual regions, albeit at the expense of precision.

The model performed best when both RFEM and CAGM were included, achieving an IoU of 0.4743 and a Dice Score of 0.6433. The significant boost in recall to 0.7838 highlights the complementary roles of RFEM and CAGM. RFEM sharpens the model's focus on localized crack patterns, while CAGM enriches the global context, leading to better overall segmentation. Interestingly, precision did not increase significantly with the inclusion of both modules, remaining relatively stable. This suggests that while the model identifies distress regions more effectively, some misclassifications persist, warranting further refinement.

5.6. Error Cases and Failure Scenarios

Figure 9 highlights several failure scenarios from *Context-CrackNet* across some of the crack datasets showing errors between ground truth masks and

predicted masks. These failure scenarios often stem from inherent challenges in the datasets, including low contrast between the cracks and their surroundings, and noise or texture inconsistencies in the images. For instance, in the CRACK500 dataset, the network struggled to differentiate between closely spaced cracks and surrounding noise, leading to incomplete or missed segments. Similarly, in the DeepCrack dataset, cracks exhibiting faint textures or irregular patterns were either under-segmented or omitted, reflecting the difficulty in capturing fine-grained structures under varying lighting conditions.

In datasets like Rissbilder and Forest, the model's predictions were influenced by the complex background textures that mimic crack patterns. This suggests that the network may occasionally misinterpret irrelevant features as cracks, especially when contrast is minimal. The GAPs dataset, with its smooth and uniform background, exposed the network's sensitivity to subtle intensity variations, resulting in missed detections for faint cracks. Additionally, the Sylvie dataset presented a unique challenge where abrupt light intensity transitions and non-crack structures confused the model, leading to segmentation errors.

5.7. Model Complexity Analysis

Computational complexity significantly impacts real-time applications such as pavement distress monitoring. Table 7 compares models based on parameters, inference time, and GFLOPs. *Context-CrackNet*, with 82.05M parameters and 243.78 GFLOPs, achieves an inference time of 15.63 ms on high-performance GPUs. Although its parameter count is higher compared to lighter models such as FPN (26.12M, 6.35 ms) and DeepLabV3Plus (26.68M, 6.72 ms), it effectively captures complex spatial relationships critical for accurate crack segmentation.

Compared to UNetPlusPlus, which has higher FLOPs (352.68 GFLOPs) and slower inference (24.44 ms), *Context-CrackNet* offers a better balance between accuracy and efficiency (see Table 3). However, due to its relatively high computational requirements, deploying *Context-CrackNet* on resource-limited mobile or edge devices may require further optimizations like pruning or quantization.

Table 7: Comparison of model complexity metrics across different architectures. Metrics include: (1) The number of parameters (in millions), (2) The average inference time (in ms) to process a single 448×448 image, and (3) GFLOPs (the number of floating-point operations, in billions, required for a single forward pass).

Model	**Parameters (M)**	**Inference Time (ms) (448 x 448 Image)**	**GFLOPs**
DeepLabV3	39.63	17.19	251.28
DeepLabV3Plus	26.68	6.72	56.42
FPN	26.12	6.35	48.08
LinkNet	31.18	6.85	66.06
MANet	147.44	12.85	114.48
PAN	24.26	7.01	53.46
PSPNet	24.26	3.09	18.14
UNet	32.52	7.67	65.60
UNetPlusPlus	48.99	24.44	352.68
Context-CrackNet	82.05	15.63	243.78

6. Conclusion

This study introduces Context-CrackNet, a new deep learning model specifically developed to address the challenges of accurately detecting tiny and subtle pavement cracks. To achieve this goal, we designed two key modules: the Region-Focused Enhancement Module (RFEM), which helps the model better identify fine details, and the Context-Aware Global Module (CAGM), which captures important global context from images. Together, these modules overcome major limitations found in previous crack segmentation approaches. Experiments conducted across ten diverse crack datasets showed that Context-CrackNet consistently outperformed existing state-of-the-art models, particularly when handling complex cracks of varying sizes under realistic conditions.

Beyond achieving high accuracy, Context-CrackNet also offers a good balance between precision and computational efficiency. This balance makes it highly suitable for practical, real-time pavement monitoring systems. By reliably detecting small cracks at an early stage, the model supports preventive maintenance strategies, potentially reducing long-term repair costs and extending the life of roads. This demonstrates the clear practical value of our research for infrastructure maintenance and management.

Furthermore, the innovations introduced in Context-CrackNet have implications beyond pavement distress detection. The RFEM and CAGM modules can be adapted to other areas requiring detailed and precise segmentation tasks under limited computational resources. Thus, our research lays the groundwork for broader applications in infrastructure monitoring and beyond.

Looking forward, future studies could explore advanced data augmentation and domain adaptation methods to help the model generalize better to unseen real-world scenarios. Additionally, developing lighter versions of the attention mechanisms and applying model compression techniques could significantly reduce computational costs, enabling deployment on edge devices with limited resources. Expanding the approach to handle multiple classes of pavement distress or integrating it into predictive maintenance systems are also promising paths to enhance infrastructure management further.

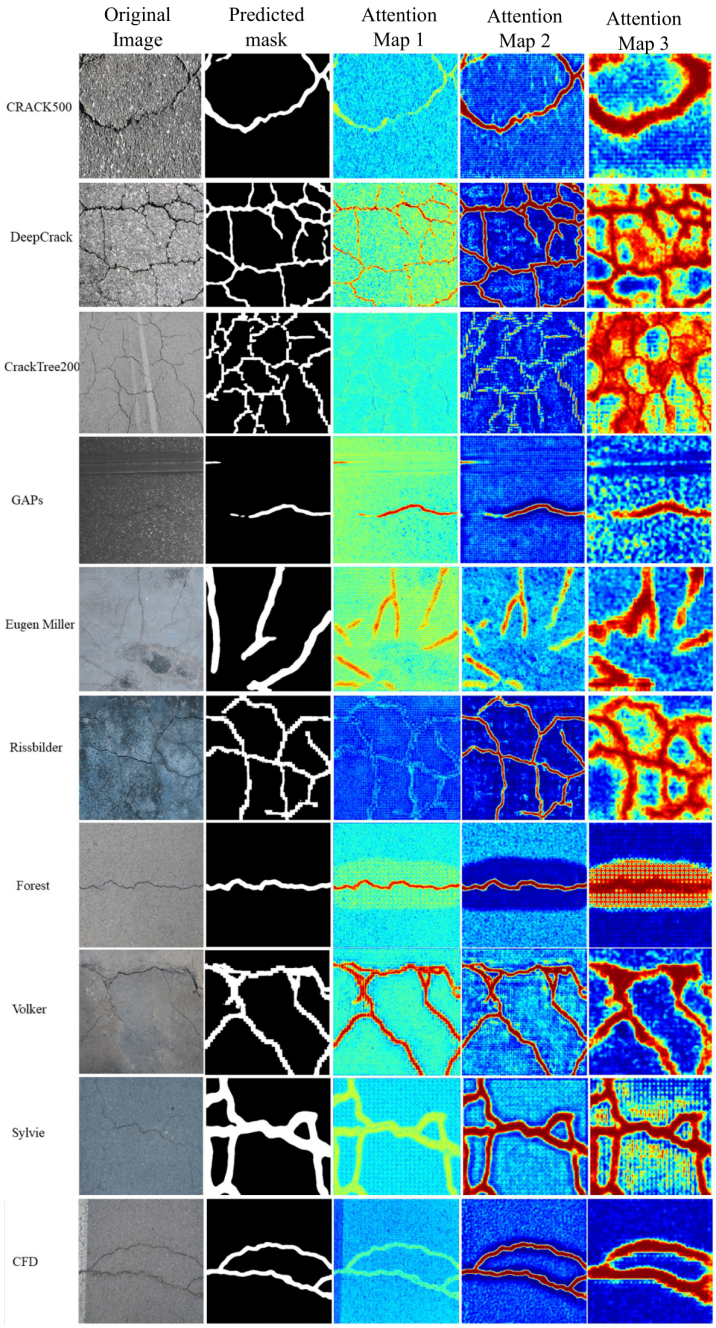

Figure 8: Visualization of predicted masks and attention maps from RFEM across various datasets. The attention maps highlight regions of interest contributing to the segmentation predictions from *Context-CrackNet*

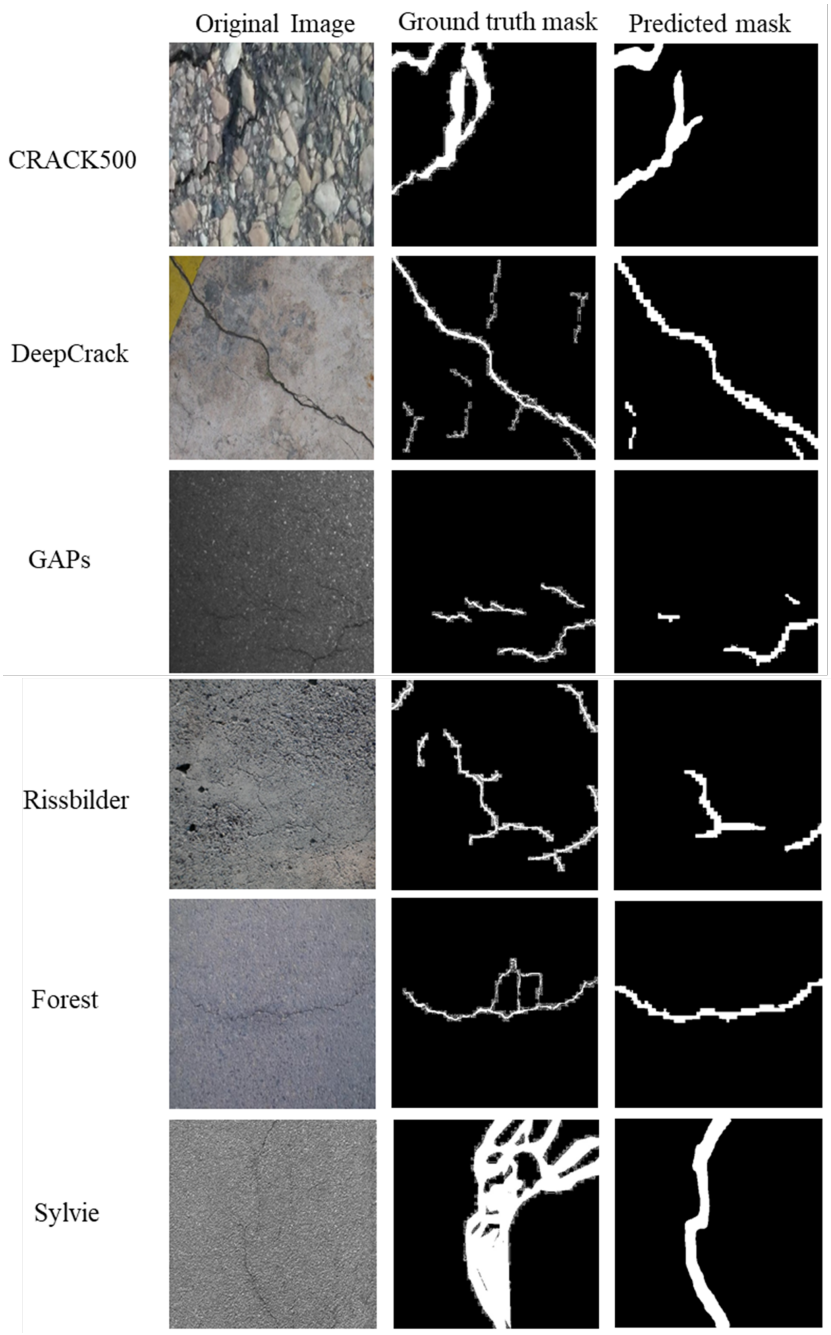

Figure 9: Failure cases from Context-CrackNet across various crack datasets, showing discrepancies between ground truth and predicted masks.

References

[1] Tian Wen, Shuo Ding, Hong Lang, Jian John Lu, Ye Yuan, Yichuan Peng, Jiang Chen, and Aidi Wang. Automated pavement distress segmentation on asphalt surfaces using a deep learning network. *International Journal of Pavement Engineering*, 24(2):2027414, 2023.

[2] James-Andrew R. Sarmiento. Pavement distress detection and segmentation using yolov4 and deeplabv3 on pavements in the philippines. *ArXiv*, abs/2103.06467, 2021.

[3] Feifei Li, Yongli Mou, Zeyu Zhang, Quan Liu, and Sabina Jeschke. A novel model for the pavement distress segmentation based on multi-level attention deeplabv3+. *Engineering Applications of Artificial Intelligence*, 137:109175, 2024.

[4] Zheng Tong, Tao Ma, Weiguang Zhang, and Ju Huyan. Evidential transformer for pavement distress segmentation. *Computer-Aided Civil and Infrastructure Engineering*, pages 2317–2338, 2023.

[5] Blessing Agyei Kyem, Eugene Kofi Okrah Denteh, Joshua Kofi Asamoah, and Armstrong Aboah. Pavecap: The first multimodal framework for comprehensive pavement condition assessment with dense captioning and pci estimation, 2024.

[6] Armstrong Aboah Neema Jakisa Owor, Yaw Adu-Gyamfi and Mark Amo-Boateng. Pavesam – segment anything for pavement distress. *Road Materials and Pavement Design*, 0(0):1–25, 2024.

[7] Chong Zhang, Yang Chen, Luliang Tang, Xu Chu, and Chaokui Li. Ctcd-net: A cross-layer transmission network for tiny road crack detection. *Remote Sensing*, 15(8), 2023.

[8] Ghada Moussa and Khaled Hussain. A new technique for automatic detection and parameters estimation of pavement crack. 07 2011.

[9] Dennis A. Morian, Douglas Frith, Shelley Stoffels, and Shervin Jahangirnejad. Developing guidelines for cracking assessment for use in vendor selection process for pavement crack data collection/analysis systems and/or services. Technical Report FHWA-RC-20-0005, Federal

Highway Administration, Office of Technical Services, Baltimore, MD, Mar 2020. Prepared by Quality Engineering Solutions, Inc.

[10] Sheng Zhang, Zhenghao Bei, Tonghua Ling, Qianqian Chen, and Liang Zhang. Research on high-precision recognition model for multi-scene asphalt pavement distresses based on deep learning. *Sci. Rep.*, 14(1):25416, October 2024.

[11] Wen-Qing Huang, Liu Feng, and Yuan-Lie He. LTPLN: Automatic pavement distress detection. *PLoS One*, 19(10):e0309172, October 2024.

[12] Hoang Nhat-Duc, Quoc-Lam Nguyen, and Van-Duc Tran. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Automation in Construction*, 94:203–213, 2018.

[13] M. Salman, S. Mathavan, K. Kamal, and M. Rahman. Pavement crack detection using the gabor filter. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 2039–2044, 2013.

[14] Eduardo Zalama, Jaime Gómez-García-Bermejo, Roberto Medina, and José Llamas. Road crack detection using visual features extracted by gabor filters. *Computer-Aided Civil and Infrastructure Engineering*, 29(5):342–358, September 2013.

[15] Xiaodong Chen, Dahang Ai, Jiachen Zhang, Huaiyu Cai, and Kerang Cui. Gabor filter fusion network for pavement crack detection. *Chinese Optics*, 2020.

[16] Allen A. Zhang, Q. Li, Kelvin C. P. Wang, and Shi Qiu. Matched filtering algorithm for pavement cracking detection. *Transportation Research Record*, 2367:30 – 42, 2013.

[17] S. Dorafshan, R. Thomas, and Marc Maguire. Benchmarking image processing algorithms for unmanned aerial system-assisted crack detection in concrete structures. *Infrastructures*, 2019.

[18] Dejin Zhang, Qingquan Li, Ying Chen, Min Cao, Li He, and Bailing Zhang. An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection. *Image Vis. Comput.*, 57:130–146, 2017.

[19] Zhongbo Li, Chao Yin, and Xixuan Zhang. Crack segmentation extraction and parameter calculation of asphalt pavement based on image processing. *Sensors (Basel, Switzerland)*, 23, 2023.

[20] Using image processing for automatic detection of pavement surface distress. *Al-Salam Journal for Engineering and Technology*, 2022.

[21] Cheng Peng, Mingqiang Yang, Qinghe Zheng, Jiong Zhang, Deqiang Wang, Ruyu Yan, Jiaxing Wang, and Bangjun Li. A triple-thresholds pavement crack detection method leveraging random structured forest. *Construction and Building Materials*, 2020.

[22] Jia Liang, Xingyu Gu, and Yizheng Chen. Fast and robust pavement crack distress segmentation utilizing steerable filtering and local order energy. *Construction and Building Materials*, 262:120084, 2020.

[23] Abdul Rahim Ahmad, Muhammad Khusairi Osman, Khairul Azman Ahmad, Muhammad Amiruddin Anuar, and Nor Aizam Muhamed Yusof. Image segmentation for pavement crack detection system. In *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 153–157, 2020.

[24] Amila Akagic, Emir Buza, Samir Omanovic, and Almir Karabegovic. Pavement crack detection using otsu thresholding for image segmentation. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1092–1097, 2018.

[25] Peggy Subirats, Jean Dumoulin, Vincent Legeay, and Dominique Barba. Automation of pavement surface crack detection using the continuous wavelet transform. In *2006 International Conference on Image Processing*, pages 3037–3040, 2006.

[26] S. Liang, Jianchun Xing, and Zhang Xun. An extraction and classification algorithm for concrete cracks based on machine vision. *IEEE Access*, 6:45051–45061, 2018.

[27] Nhat-Duc Hoang and Quoc-Lam Nguyen. A novel method for asphalt pavement crack classification based on image processing and machine learning. *Engineering with Computers*, 35:487 – 498, 2018.

[28] Abbas Ahmadi, Sadjad Khalesi, and A. Golroo. An integrated machine learning model for automatic road crack detection and classification in urban areas. *International Journal of Pavement Engineering*, 23:3536 – 3552, 2021.

[29] I. Barkiah and Yuslena Sari. Overcoming overfitting challenges with hog feature extraction and xgboost-based classification for concrete crack monitoring. *International Journal of Electronics and Telecommunications*, 2023.

[30] Adrien Müller, N. Karathanasopoulos, C. Roth, and D. Mohr. Machine learning classifiers for surface crack detection in fracture experiments. *International Journal of Mechanical Sciences*, 209:106698, 2021.

[31] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[32] Wenyu Zhang, Zhenjiang Zhang, Dapeng Qi, and Yun Liu. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors (Basel, Switzerland)*, 14:19307 – 19328, 2014.

[33] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[34] Jie Gao, Dongdong Yuan, Zheng Tong, Jiangang Yang, and Di Yu. Autonomous pavement distress detection using ground penetrating radar and region-based deep learning. *Measurement*, 164:108077, 2020.

[35] Suli Bai, Lei Yang, Yanhong Liu, and Hongnian Yu. Dmf-net: A dual-encoding multi-scale fusion network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 25:5981–5996, 2024.

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[37] Blessing Agyei Kyem, Eugene Kofi Okrah Denteh, Joshua Kofi Asamoah, Kenneth Adomako Tutu, and Armstrong Aboah. Advancing pavement distress detection in developing countries: A novel deep learning approach with locally-collected datasets, 2024.

[38] Yann Lecun and Y. Bengio. Convolutional networks for images, speech, and time-series. 01 1995.

[39] Baoxian Li, Kelvin Wang, Allen Zhang, Enhui Yang, and Guolong Wang. Automatic classification of pavement crack using deep convolutional neural network. *International Journal of Pavement Engineering*, 21:1–7, 06 2018.

[40] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3708–3712, 2016.

[41] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 04 2019.

[42] Kai Li, Jie Yang, Siwei Ma, Bo Wang, Shanshe Wang, Yingjie Tian, and Zhiquan Qi. Rethinking lightweight convolutional neural networks for efficient and high-quality pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 25:237–250, 01 2024.

[43] Qiang Zhou, Zhong Qu, and Fang-rong Ju. A lightweight network for crack detection with split exchange convolution and multi-scale features fusion. *IEEE Transactions on Intelligent Vehicles*, 8(3):2296–2306, 2023.

[44] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768, 2020.

[45] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, T. Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. pages 21297–21309, 2021.

[46] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Y. Teh. Set transformer. *ArXiv*, abs/1810.00825, 2018.

[47] Hairui Fang, Jin Deng, Yaoxu Bai, Bo Feng, Sheng Li, Siyu Shao, and Dongsheng Chen. Clformer: A lightweight transformer based on convolutional embedding and linear self-attention with strong robustness for

bearing fault diagnosis under limited sample conditions. *IEEE Transactions on Instrumentation and Measurement*, 71:1–8, 2022.

[48] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shimin Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:5436–5447, 2021.

[49] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[50] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3708–3712. IEEE, 2016.

[51] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[52] Sangwoo Ham, Soohyeon Bae, Hwiyoung Kim, Impyeong Lee, Gyu-Phil Lee, and Donggyu Kim. Training a semantic segmentation model for cracks in the concrete lining of tunnel. 11 2021.

[53] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[54] Markus Eisenbach, Ronny Stricker, Daniel Seichter, Karl Amende, Klaus Debes, Maximilian Sesselmann, Dirk Ebersbach, Ulrike Stoeckert, and Horst-Michael Gross. How to get pavement distress detection ready for deep learning? a systematic approach. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2039–2047, 2017.

[55] Myeongsuk Pak and Sanghoon Kim. Crack detection using fully convolutional network in wall-climbing robot. In James J. Park, Simon James Fong, Yi Pan, and Yunsick Sung, editors, *Advances in Computer Science*

and Ubiquitous Computing, pages 267–272, Singapore, 2021. Springer Singapore.

[56] Rabih Amhaz, Sylvie Chambon, Jérôme Idier, and Vincent Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2718–2729, 2016.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[59] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, 39(6):1856–1867, 2020.

[60] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 06 2017.

[61] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing.

[62] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. pages 6392–6401, 06 2019.

[63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017.

[64] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2017.

[65] RUI LI, Shunyi Zheng, Ce Zhang, Chenxi Duan, Jianlin Su, Libo Wang, and Peter Atkinson. Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–13, 07 2021.

[66] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. 05 2018.

`

Authors' Response to Reviews of

Article Title: "Context-CrackNet: A Context-Aware Framework for Precise Segmentation of Tiny Cracks in Pavement images"

Blessing Agyei Kyem, Joshua Kofi Asamoah, Armstrong Aboah

Construction and Building Materials Submission Date: 04/10/2025

Dear Editor**,**

Thank you for allowing a resubmission of our manuscript, with an opportunity to address the reviewers' comments.

We appreciate the insightful feedback provided by the reviewers, which has significantly enhanced the quality and clarity of our manuscript. We have carefully considered each concern and implemented appropriate revisions, as detailed in the individual responses provided for each reviewer comment. These revisions are summarized below. To address the reviewer comments comprehensively, we updated the manuscript by:

1. Correcting equation punctuation, added missing equation numbers, unified citation style, and capitalized "Images".
2. Defining *tiny* (< 1 mm) and *small* (1–3 mm) cracks and expanded introduction with concrete examples of segmentation and efficiency challenges.
3. Adding mean ± standard-deviation reporting plus Wilcoxon significance tests (new Sec. 5.3, Table 4, Fig. 6).
4. Inserting step-by-step explanatory text between key RFEM and CAGM equations to clarify feature flow.
5. Adding rationale for linear self-attention and its O(N) efficiency relative to classical $O(N^2)$ attention.
6. Extending model-complexity table and discussion; noted need for pruning/quantization for mobile use.
7. Clarifying that all datasets are natively 448 × 448, so no resolution-induced detail loss; flagged multi-scale training as future work.
8. Expanding cross-dataset results analysis and included failure-case discussion with new visual examples (Fig. 8).
9. Adding RFEM attention-map visualizations (Fig. 7) to show where the network focuses.
10. Performing ablation study demonstrating combined RFEM + CAGM gains (+5 Dice) over baseline.
11. Discussed practical impact of seemingly small F1-score gains for preventive maintenance.
12. Re-wrote conclusion to emphasize contributions, real-world benefits, and future lightweight extensions.
13. Reduced repetitive wording and polished language throughout the manuscript.

`

The subsequent pages provide a point-by-point response to all reviewer comments with detailed justifications and revisions. We are confident that these improvements significantly strengthen our manuscript and address the reviewers' concerns.

Thank you for your time and consideration.

Best Regards,
Armstrong Aboah, Ph.D,
Department of Civil, Environmental and Construction Engineering,
North Dakota State University.
04/10/2025.

`

Response to Reviewers

REVIEWER 1:

Comment 1: Equations (10), (20), and (23) should end with a comma rather than a period. Equations in lines 32 and 40 on page 20 are not numbered. The citation format is inconsistent, e.g., "Li et al [3]" and "Kyem et al. [5]". The title "images" should have the first letter capitalized.

Author Response:

Thank you very much for pointing out these formatting issues. We have carefully addressed each point to enhance consistency and readability in the manuscript.

- We corrected the punctuation in Equations (10), (20), and (23), changing periods to commas as suggested, ensuring consistency throughout the equations.
- We numbered the equations located on lines 32 and 40 on page 20 to maintain clarity and proper referencing.
- We standardized all citation formats in the manuscript to consistently follow the recommended format, using the style "Li et al. [3]" and "Kyem et al. [5]" throughout.
- We capitalized the title "Images" appropriately to adhere to proper capitalization rules and manuscript conventions.

Comment 2: The introduction section mentions the challenges of existing deep learning models in fine-grained segmentation and computational efficiency but fails to elaborate on the specific manifestations of these challenges. This part of the content could be further refined to better highlight the necessity of the research.

Author Response:

Thank you very much for highlighting this important aspect. We fully agree that elaborating further on the specific manifestations of fine-grained segmentation and computational efficiency challenges strengthens the necessity and clarity of our research.

First, we have expanded the Introduction section by clearly describing the practical manifestations of these challenges. Specifically, we explain how difficult it is to detect tiny cracks under real-world conditions such as variable lighting or the presence of debris. Similarly, we discuss how small cracks frequently present challenges due to their irregular geometry and subtle texture patterns, which can easily be misclassified or overlooked. These concrete examples clarify the nature of fine-grained segmentation challenges faced by existing models and emphasize why accurate early detection remains a critical goal.

Moreover, we explain that many existing deep learning models lack adequate multi-scale feature representation, which hinders their ability to consistently detect cracks across different sizes—from subtle, fine-grained cracks to larger, more prominent ones. Compounding this issue is the

`

computational burden associated with processing high-resolution images, which are often necessary to preserve fine crack details. These high-resolution requirements substantially increase memory consumption and inference time. This, in turn, restricts the real-time deployment potential of these models, particularly in resource-constrained environments such as mobile platforms or large-scale infrastructure systems.

Below is the highlighted paragraph in the Introduction that talks about that:

Despite the significant progress achieved with deep learning models for pavement distress segmentation, several limitations remain. ne critical yet unresolved challenge is the accurate segmentation of tiny and small pavement cracks. Tiny cracks typically refer to pavement cracks narrower than 1 mm. These cracks are faint, often discontinuous, and extremely challenging to detect, particularly under varying environmental conditions [7], [8], [9]. Small cracks, slightly larger, range from about 1 mm to 3 mm in width [9]. Although more visible than tiny cracks, small cracks still pose significant challenges for accurate segmentation due to their irregular shapes, textures, and subtle appearance in images. Identifying these small and tiny defects early enables preventive maintenance before they develop into more extensive damage. By intervening at this early stage, maintenance teams can prevent minor issues from escalating, thereby reducing repair costs and minimizing disruptions to traffic. To achieve these outcomes, it is essential to adopt advanced models capable of effectively handling both very small and larger cracks. However, achieving this goal is not without challenges, as many existing models struggle with multi-scale feature representation, which hinders their ability to effectively detect both small-scale and large-scale cracks [10]. In addition to this challenge is the lack of a comprehensive understanding of global context which often limits the model's ability to capture large-scale spatial relationships and distinguish between interconnected distresses and noise. This results in inconsistent segmentation of extensive distress patterns such as longitudinal and alligator cracks. Furthermore, many deep learning models require high-resolution inputs to detect subtle crack features, significantly increasing memory usage and inference time [11]. This computational burden limits their suitability for real-time deployment and scalability for large-scale pavement monitoring systems. These above limitations highlight the pressing need for innovative approaches to enhance segmentation performance and address the shortcomings of existing methods.

Comment 3: The paper reports key metrics such as mIoU and Dice Score, but does not provide error ranges (e.g., standard deviation), making it difficult to assess the stability and reliability of the results. Additionally, the paper lacks statistical significance testing (e.g., p-value), which makes it impossible to confirm whether the improvements are statistically significant. It is recommended that the authors provide the mean ± standard deviation to quantify the variability of

`

the results, and perform significance testing using t-tests or Wilcoxon tests to ensure the validity of the model improvements.

Author Response:

We sincerely thank Reviewer 1 for this important comment. It significantly enhanced the robustness and reliability of our results.

To address the concerns raised, we have added a comprehensive statistical analysis section to our manuscript (Section 5.3, "Statistical Analysis,") under the Results and Discussion section. Initially, we calculated the mean and standard deviation for the IoU and Dice Score metrics across all benchmark datasets and models, as summarized in Table 3. This table provides clarity about the variability and stability of each model's performance.

Moreover, we implemented the Wilcoxon signed-rank test to rigorously evaluate the statistical significance of the performance improvements achieved by Context-CrackNet compared to baseline models. The distribution of Validation IoU and Dice Scores for each model was visualized using boxplots (Figure 6), which clearly demonstrate that Context-CrackNet consistently achieves higher median values.

To statistically substantiate these visual observations, we computed Wilcoxon test p-values (Table 4), contrasting the Validation IoU scores between Context-CrackNet and each baseline model. Our findings indicate that the performance improvements by Context-CrackNet are statistically significant for all comparisons ($p = 0.002$) except with UnetPlusPlus, where the difference, although substantial, does not reach the conventional threshold for statistical significance ($p = 0.0645$).

We believe these additions comprehensively address the reviewer's concern regarding error ranges and statistical significance testing, improving the clarity, reliability, and interpretability of our results.

Below is the section of the manuscript that highlights the Statistical analysis:

To begin our analysis, we first calculated the mean and standard deviation of the IoU and Dice Score across all datasets for each model, as summarized in Table 6. Although these descriptive statistics provide an initial sense of overall performance variability, they do not by themselves clarify whether the observed differences are statistically significant.

Therefore, to rigorously evaluate whether the improvements achieved by Context-CrackNet were indeed meaningful relative to other baseline models, we employed the Wilcoxon signed-rank test, which determines if observed performance gains could simply stem from random variations. Figure 7 presents the distribution of Validation IoU and Dice Scores across the evaluated segmentation frameworks, where the boxplots show that Context-CrackNet consistently attains

`

higher median IoU and Dice values compared to other approaches. To substantiate these visual findings, we further computed Wilcoxon test p-values by contrasting the Validation IoU of Context-CrackNet with each baseline model (see Table 7). These results indicate that Context-CrackNet's improvements are statistically significant.

Comment 4: The logical connections between some equations in the description of the RFEM and CAGM modules are not sufficiently clear. For example, equations (12) and (13) lack necessary explanatory text, making it difficult for readers to understand the logical flow from feature transformation to the generation of the attention activation map. Detailed explanations should be added between equations to help the reader understand the purpose and function of each step.

Author Response:

We sincerely thank you for highlighting this critical point. We agree that enhancing the clarity and logical connection between equations significantly improves readability and understanding of our proposed modules.

Based on your suggestions, we revised Sections **3.3 (Context-Aware Global Module - CAGM)** and **3.4 (Region-Focused Enhancement Module - RFEM)** by explicitly detailing the intermediate logical steps and the purpose of each transformation between equations.

Specifically, we have added clear and concise explanatory text to better illustrate:

- The rationale behind reshaping and linear projections in the CAGM, explicitly clarifying how the spatial dimensions are flattened into sequences for self-attention computation.
- The reason for dimensionality reduction of the key and value matrices in the CAGM, and how it helps in computational efficiency.
- The interpretation of the softmax operation in attention score computation, detailing its role in generating normalized weights for aggregating global context.
- The logical steps for reconstructing the sequence output back into spatial feature maps.

In the RFEM, we have specifically clarified:

- The motivation behind projecting encoder and decoder features into a shared feature space, emphasizing their alignment and compatibility before fusion.
- The reasoning for applying element-wise addition followed by ReLU activation, explaining how this captures spatial interactions.
- The generation and application of the attention coefficient map, clearly describing how this step emphasizes spatially relevant features and suppresses less significant ones.

`

- The final concatenation and convolutional refinement steps, explicitly highlighting how the fusion of refined encoder and decoder features enhances fine-grained segmentation details.

Comment 5: While the use of a linear self-attention mechanism is mentioned in the description of the CAGM module, there is no detailed explanation of how this mechanism effectively captures global context information. It is suggested to provide a more detailed explanation of the principles and advantages of the linear self-attention mechanism, including why this mechanism was chosen over traditional self-attention mechanisms, and how it integrates with other parts of the model to achieve accurate segmentation of large-scale cracks.

Author Response:

We sincerely thank the reviewer for this insightful observation.

In response, we have added a dedicated explanatory paragraph at the end of Section **3.3 (Context-Aware Global Module - CAGM)** to clearly articulate the motivation and advantages of using a linear self-attention mechanism in our design.

In the revised text, we explain that traditional self-attention suffers from quadratic complexity, making it inefficient for high-resolution pavement images. In contrast, our linear self-attention formulation reduces complexity to $\mathcal{O}(N \cdot k)$, enabling scalable global context modeling. We also highlight how this mechanism allows each spatial position to incorporate global information which is critical for identifying long or disconnected crack patterns.

Additionally, we describe how the CAGM complements the RFEM by enriching bottleneck features with non-local context before refinement in the decoder. This integration improves the segmentation of large-scale and spatially complex pavement cracks. We believe this revision has significantly enhanced the clarity and justification of our design choices, and we are grateful for the suggestion.

Below is the section in the manuscript that talks about the importance of the linear self-attention mechanism:

Rationale for Using Linear Self-Attention. *Traditional self-attention has a quadratic complexity* $\mathcal{O}(N^2)$ *, which limits its scalability to high-resolution images. In contrast, the linear self-attention used in the CAGM reduces this to* $\mathcal{O}(N \cdot k)$ *by projecting keys and values into a lower-dimensional space, enabling efficient global context modeling. This is particularly important for pavement images, where cracks can span large, non-contiguous regions. The CAGM allows each spatial location to incorporate information from the entire image, improving segmentation of extensive or fragmented crack patterns. Positioned at the bottleneck, it complements the RFEM by enriching features with global cues before fine-grained refinement in the decoder.*

`

Comment 6: Although the paper emphasizes the balance between accuracy and computational cost, the model's actual parameter count (82.05M), FLOPs (243.78G), and inference time (15.63ms, based on high-performance GPUs) are still relatively high compared to lightweight models. Further validation is needed to confirm whether the model is truly suitable for mobile deployment.

Author Response:

Thank you for this insightful comment. We agree with your observation regarding the relatively high number of parameters (82.05M), computational complexity (243.78 GFLOPs), and inference time (15.63 ms) of our proposed Context-CrackNet model. These factors indeed affect the suitability of the model for resource-limited or mobile devices.

To address this concern, we have revised our manuscript's "Model Complexity Analysis" section (please see Section 5.7). In this updated section, we openly acknowledge that Context-CrackNet has higher computational requirements compared to lighter segmentation models. We clarify that while these increased parameters and computations allow our model to effectively handle complex spatial relationships essential for accurately segmenting tiny cracks, they may hinder direct use on edge or mobile devices.

We further note in the revised section that additional optimization techniques such as pruning or quantization may be necessary to reduce the computational load. This step would help ensure that our model can operate efficiently in resource-constrained environments.

Your comment has been valuable in highlighting this important aspect. It has encouraged us to transparently discuss the trade-offs between accuracy and model complexity. We appreciate your suggestion, as it has helped us improve the clarity and practical applicability of our research.

Below is the revised section in the manuscript:

5.7 Model Complexity Analysis

Computational complexity significantly impacts real-time applications such as pavement distress monitoring. Table 9 compares models based on parameters, inference time, and GFLOPs. Context-CrackNet, with 82.05M parameters and 243.78 GFLOPs, achieves an inference time of 15.63 ms on high-performance GPUs. Although its parameter count is higher compared to lighter models such as FPN (26.12M, 6.35 ms) and DeepLabV3Plus (26.68M, 6.72 ms), it effectively captures complex spatial relationships critical for accurate crack segmentation.

Compared to UNetPlusPlus, which has higher FLOPs (352.68 GFLOPs) and slower inference (24.44 ms), Context-CrackNet offers a better balance between accuracy and efficiency (see Tables 3, 4, and 5). However, due to its relatively high computational requirements, deploying Context-CrackNet on resource-limited mobile or edge devices may require further optimizations like pruning or quantization.

`

Comment 7: The paper resizes all datasets to a uniform resolution of 448×448 pixels, which simplifies the experiment but may cause the loss of details in high-resolution images and blur low-resolution images, thereby affecting the detection accuracy of small cracks. It is recommended that the authors conduct comparative experiments at different resolutions to assess the impact of this adjustment on model performance, and consider using multi-scale training to improve the model's robustness.

Author Response:

We thank the reviewer for drawing attention to the possible influence of image resizing on segmentation performance.

Upon re-examining the data preparation procedure, we confirmed that all images utilized in this study were sourced directly from the official repositories of the ten benchmark crack segmentation datasets. Each image was originally provided at a fixed resolution of 448 × 448 pixels. As a result of this uniformity in image dimensions, the processing pipeline did not require any additional resizing, thereby preserving the original visual and structural detail of each image.

Comment 8: Although the paper provides performance metrics, it lacks a deeper analysis of the results, such as the reasons for performance differences across datasets. It is suggested that the authors conduct a detailed analysis of the performance differences on various datasets in the results section and explore the possible reasons. For example, the model's higher mIoU on the CFD dataset might be due to the clearer crack features in that dataset, while the more complex crack features in the CRACK500 dataset could lead to a slight decrease in performance.

Author Response:

Thank you for the thoughtful and valuable feedback regarding the need for a deeper analysis of performance differences across datasets.

We would like to note that our manuscript already includes a dedicated subsection **Section 5.5: Error Cases and Failure Scenarios** that addresses this concern. In this section, we analyze several representative failure cases and discuss how specific dataset characteristics contribute to performance fluctuations.

For example, we observe that datasets like **CFD**, which feature clearer and more distinct crack patterns, allow the model to achieve higher mIoU and Dice scores. On the other hand, datasets such as **CRACK500** present more complex scenarios such as low contrast, overlapping cracks, and background noise which sometimes lead to under-segmentation or missed detections. These issues are explicitly highlighted in the paragraph.

We further extend this analysis to other datasets:

- In **DeepCrack**, faint or irregular crack textures pose challenges under varied lighting conditions.

`

- In **Rissbilder** and **Forest**, complex textures can resemble cracks, introducing false positives.
- In **GAPs**, subtle intensity variations make faint cracks difficult to detect.
- In **Sylvie**, abrupt lighting changes and non-crack structures impact segmentation accuracy.

The section is supported by **Figure 8**, which visually illustrates discrepancies between predicted masks and ground truth labels across these challenging cases.

Below is the section in the manuscript that highlights this:

5.6 Error Cases and Failure Scenarios

Figure 9 highlights several failure scenarios from Context-CrackNet across some of the crack datasets showing errors between ground truth masks and predicted masks. These failure scenarios often stem from inherent challenges in the datasets, including low contrast between the cracks and their surroundings, and noise or texture inconsistencies in the images. For instance, in the CRACK500 dataset, the network struggled to differentiate between closely spaced cracks and surrounding noise, leading to incomplete or missed segments. Similarly, in the DeepCrack dataset, cracks exhibiting faint textures or irregular patterns were either under-segmented or omitted, reflecting the difficulty in capturing fine-grained structures under varying lighting conditions.

In datasets like Rissbilder and Forest, the model's predictions were influenced by the complex background textures that mimic crack patterns. This suggests that the network may occasionally misinterpret irrelevant features as cracks, especially when contrast is minimal. The GAPs dataset, with its smooth and uniform background, exposed the network's sensitivity to subtle intensity variations, resulting in missed detections for faint cracks. Additionally, the Sylvie dataset presented a unique challenge where abrupt light intensity transitions and non-crack structures confused the model, leading to segmentation errors.

REVIEWER 3:

Comment 1: This study presents Context-CrackNet, a novel architecture aiming to tackle the long - standing challenge of precisely segmenting tiny and subtle cracks in pavements. The research is of great significance, with a clear logical structure, rich content, and detailed demonstrations. However, the conclusion section still needs further modification and improvement to better highlight the research value of this paper.

Author Response: Thank you very much for your valuable feedback and suggestion. We sincerely appreciate your recognition of the significance, structure, content, and demonstration of our work. As suggested, the authors have revised the conclusion section of the manuscript to better highlight the research value of our study clearly and explicitly.

`

Specifically, we improved the conclusion by explicitly emphasizing

1. The distinct advantages and contributions of Context-CrackNet over existing models, emphasizing its value in accurately segmenting tiny and subtle pavement cracks.
2. The practical impact of our research on real-world pavement management, including preventive maintenance and resource efficiency.
3. Clear articulation of the broader implications of our findings for infrastructure monitoring and maintenance beyond pavement crack segmentation.

The changes have been included in the manuscript as shown below:

Revised Conclusion:

This study introduces Context-CrackNet, a novel deep learning architecture specifically designed to overcome persistent challenges in accurately segmenting tiny and subtle pavement cracks. By integrating the Region-Focused Enhancement Module (RFEM) and the Context-Aware Global Module (CAGM), Context-CrackNet effectively captures fine-grained crack details and robustly incorporates global contextual information, addressing significant limitations observed in current segmentation frameworks. Comprehensive evaluations across ten diverse and publicly available crack datasets have consistently demonstrated superior performance in comparison to state-of-the-art models, particularly highlighting the model's capability to handle multi-scale crack patterns and various real-world complexities.

The demonstrated balance between segmentation accuracy and computational efficiency underscores Context-CrackNet's significant potential for practical deployment in real-time pavement monitoring systems. Its ability to reliably detect tiny cracks early on facilitates preventive maintenance strategies, thereby significantly reducing long-term repair costs and extending the longevity of transportation infrastructure. This practical benefit underscores the real-world impact and inherent value of our research to stakeholders involved in infrastructure management and maintenance.

Moreover, our study sets a broader foundation for future research into advanced context-aware segmentation frameworks. The developed modules, RFEM and CAGM, offer versatile contributions that can be effectively adapted to other domains that require detailed segmentation of subtle features under computational constraints.

Future research directions include employing sophisticated data augmentation and domain adaptation techniques to further improve generalizability across unseen scenarios. Additionally, efforts toward lightweight attention mechanisms and model compression strategies could substantially reduce computational demands, facilitating deployment on edge and resource-constrained devices. Extending Context-CrackNet to multi-class segmentation and integrating it with predictive maintenance platforms presents promising avenues to enhance infrastructure management comprehensively.

`

REVIEWER 4:

Comment 1: The authors used the phrase "small and tiny cracks" over 20 times in the paper. However, no definition(s) are provided. A clear definition should be provided.

Author Response:

Thank you very much for highlighting this important point. We agree that clearly defining the terminology enhances the clarity and readability of the manuscript.

In response, we have now explicitly provided definitions for the terms "**small cracks**" and "**tiny cracks**" within the manuscript's introduction (beginning of Paragraph 3). We defined these terms based on crack widths and visibility criteria commonly used in the pavement industry and pavement distress literature [7], [8], [9].

The updates have been included in the manuscript as shown below:

Tiny cracks typically refer to pavement cracks narrower than 1 mm. These cracks are faint, often discontinuous, and extremely challenging to detect, particularly under varying environmental conditions [7], [8], [9]. Small cracks, slightly larger, range from about 1 mm to 3 mm in width [9]. Although more visible than tiny cracks, small cracks still pose significant challenges for accurate segmentation due to their irregular shapes, textures, and subtle appearance in images.

Comment 2: Page 22 Figure 5: To avoid being accused of bias, the authors should also highlight the limitation of their model. For example, looking at the CrackTree200 images, one could argue that Cracknet also missed some of cracks in the ground truth (see upper and lower portion of the left hand side of the image. There are many examples like that in the other images.

Author Response:

Thank you for your valuable feedback and for highlighting this important consideration. We completely agree with your suggestion that clearly acknowledging the limitations of our model is crucial for a balanced presentation and to avoid potential bias.

To address your comment, we would like to clarify that our manuscript already includes a dedicated subsection titled "**Error Cases and Failure Scenarios**" (Section 5.5). This section explicitly discusses various limitations and challenges encountered by Context-CrackNet, such as difficulty distinguishing closely spaced cracks, sensitivity to faint textures, and issues arising from complex background textures which affected the segmentation of the cracks.

Additionally, the manuscript already includes a Figure (**Figure 8),** which visually highlights several failure scenarios across different datasets, clearly illustrating the discrepancies between ground truth masks and predicted masks. This figure helps readers visually appreciate and

`

understand the specific instances where Context-CrackNet faces challenges, similar to the ones you observed in CrackTree200 and other datasets shown in Figure 5.

Below highlights the section of the manuscript that discusses the Failure scenarios of Context-CrackNet. Figure 8 has also been shown below.

5.5 Error Cases and Failure Scenarios

Figure 8 highlights several failure scenarios from Context-CrackNet across some of the crack datasets showing errors between ground truth masks and predicted masks. These failure scenarios often stem from inherent challenges in the datasets, including low contrast between the cracks and their surroundings, and noise or texture inconsistencies in the images. For instance, in the CRACK500 dataset, the network struggled to differentiate between closely spaced cracks and surrounding noise, leading to incomplete or missed segments. Similarly, in the DeepCrack dataset, cracks exhibiting faint textures or irregular patterns were either under-segmented or omitted, reflecting the difficulty in capturing fine-grained structures under varying lighting conditions.

In datasets like Rissbilder and Forest, the model's predictions were influenced by the complex background textures that mimic crack patterns. This suggests that the network may occasionally misinterpret irrelevant features as cracks, especially when contrast is minimal. The GAPs dataset, with its smooth and uniform background, exposed the network's sensitivity to subtle intensity variations, resulting in missed detections for faint cracks. Additionally, the Sylvie dataset presented a unique challenge where abrupt light intensity transitions and non-crack structures confused the model, leading to segmentation errors.

`

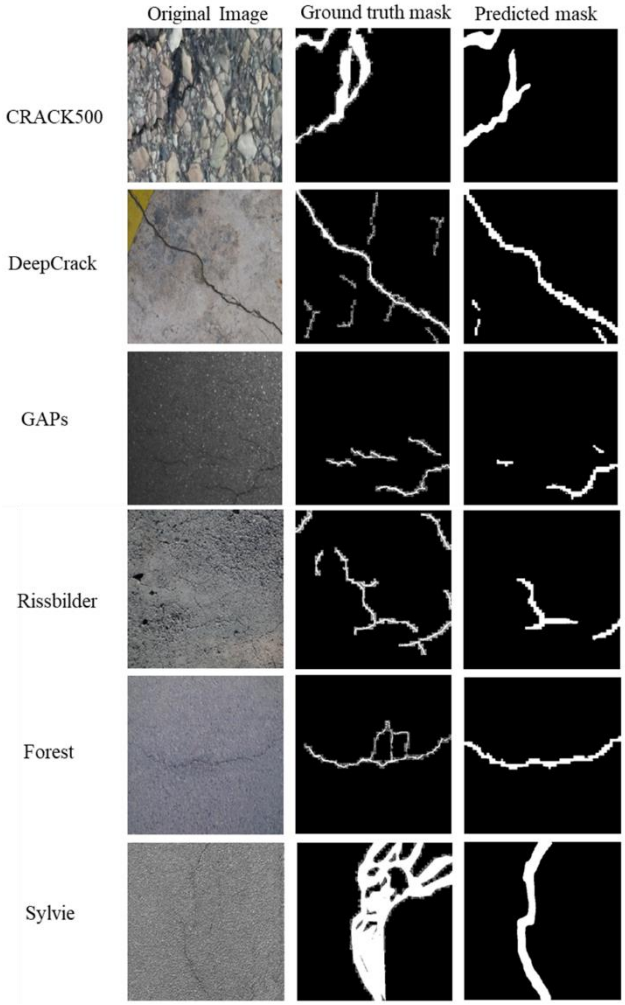

Figure 8: Failure cases from Context-CrackNet across various crack datasets, showing discrepancies between ground truth and predicted masks.

Comment 3: Table 3: Indicate the variability in the data reported. What does the results represent? Are they the result of one run or multiple runs of the same model. This information is important because I am assuming the authors randomly assigned images to the training and verification/testing sample.

Author Response:

Thank you very much for highlighting this important point. We agree that clearly describing the experimental setup and variability is crucial for properly interpreting our results.

The results are multiple runs of the different models on different datasets. To further clarify how we ran our experiments, we have revised the **"Training Settings" subsection (Section 4.2.1)** within the Implementation details (**Section 4.2**) to clearly describe our evaluation procedure. Specifically, we randomly split each of the ten crack datasets into separate and consistent training and validation sets. All models, including Context-CrackNet and baseline models (U-Net, U-Net++, DeepLabV3, DeepLabV3+, FPN, PSPNet, LinkNet, MAnet, and PAN), were individually trained on each dataset's training set and subsequently evaluated on the respective validation sets.

`

This standardized approach ensured consistency and rigorous comparisons across all models and datasets.

Consequently, the results presented in **Table 3** represent the average performance metrics (mIoU, Dice, Precision, Recall, and F1 Score) calculated across the validation sets of each dataset. These metrics reflect the performance obtained from independently evaluating each trained model on their respective validation datasets.

Below is the section in the manuscript that highlights how the experiments were conducted:

All models, including Context-CrackNet and the baseline models (U-Net, U-Net++, DeepLabV3, DeepLabV3+, FPN, PSPNet, LinkNet, MAnet, and PAN), were trained separately on each of the ten crack datasets. Each dataset was randomly split into training and validation sets, ensuring a consistent evaluation setup for every model. After training, each model was evaluated on the corresponding validation set of each dataset to measure performance.

Comment 4: Table 3: Following on from my previous comment, can we say the authors model is significantly better. For example looking at CFD data, can we say an F1 score of 0.8046 for Context-CrackNet is significantly higher than 0.7835 for UnetPlusPlus?

Author Response:

Thank you very much for your thoughtful observation. We understand your concern about whether the performance differences shown in Table 3 are practically meaningful and statistically significant.

At first glance, the difference in F1 scores such as 0.8046 for Context-CrackNet versus 0.7835 for U-Net++ on the CFD dataset might seem modest. However, it is important to recognize the significance of a metric such as F1 score in pavement distress segmentation. For instance, F1 score balances precision (correctly identified cracks versus false detections) and recall (the ability to identify all true cracks), making it a crucial metric for evaluating segmentation performance. Even small increases in the F1 score indicate a meaningful improvement in accurately and reliably detecting pavement distresses.

From a practical perspective, this improvement means Context-CrackNet can more consistently identify subtle and challenging pavement cracks, reducing the likelihood of missed detections and false alarms. In real-world pavement management, this translates into better preventive maintenance strategies, as road maintenance teams can detect cracks earlier and more reliably. Early and accurate detection of these cracks directly supports proactive interventions, significantly reducing the severity and cost of pavement repairs and extending pavement lifespan. Consequently, this not only provides cost savings but also enhances roadway safety by minimizing disruptions and ensuring optimal pavement conditions.

`

Thus, although numerically the difference may appear small, practically and technically, the improvement offered by Context-CrackNet significantly benefits real-world pavement maintenance operations and preventive management strategies.

`

REFERENCES

[7] Chong Zhang, Yang Chen, Luliang Tang, Xu Chu, and Chaokui Li. Ctcd-net: A cross-layer transmission network for tiny road crack detection. Remote Sensing, 15(8), 2023.

[8] Ghada Moussa and Khaled Hussain. A new technique for automatic detection and parameters estimation of pavement crack. 07 2011.

[9] Dennis A. Morian, Douglas Frith, Shelley Stoffels, and Shervin Jahangirnejad. Developing guidelines for cracking assessment for use in vendor selection process for pavement crack data collection/analysis systems and/or services. Technical Report FHWA-RC-20-0005, Federal Highway Administration, Office of Technical Services, Baltimore, MD, Mar 2020. Prepared by Quality Engineering Solutions, Inc.

- Context-CrackNet effectively segments diverse road distress particularly, including tiny, small cracks and extensive damage.
- Region-Focused Enhancement Module (RFEM) enhances segmentation of smaller and subtle cracks.
- Context-Aware Global Module (CAGM) captures broader contextual features for larger distress patterns.
- Outperformed ten state-of-the-art models across publicly available crack datasets
- Balanced high accuracy (mIoU, Dice score) with efficient inference for scalable real-time deployment.

Context-CrackNet: A Context-Aware Framework for Precise Segmentation of Tiny Cracks in Pavement images

Blessing Agyei Kyem[a], Joshua Kofi Asamoah[a], Armstrong Aboah[a],*

[a]North Dakota State University, Fargo, 58105, North Dakota, United States

Context-CrackNet: A Context-Aware Framework for Precise Segmentation of Tiny Cracks in Pavement Images

Blessing Agyei Kyem[a], Joshua Kofi Asamoah[a], Armstrong Aboah[a,*]

[a]*North Dakota State University, Fargo, 58105, North Dakota, United States*

Abstract

The accurate detection and segmentation of pavement distresses, particularly tiny and small cracks, are critical for early intervention and preventive maintenance in transportation infrastructure. Traditional manual inspection methods are labor-intensive and inconsistent, while existing deep learning models struggle with fine-grained segmentation and computational efficiency. To address these challenges, this study proposes Context-CrackNet, a novel encoder-decoder architecture featuring the Region-Focused Enhancement Module (RFEM) and Context-Aware Global Module (CAGM). These innovations enhance the model's ability to capture fine-grained local details and global contextual dependencies, respectively. Context-CrackNet was rigorously evaluated on ten publicly available crack segmentation datasets, covering diverse pavement distress scenarios. The model consistently outperformed 9 state-of-the-art segmentation frameworks, achieving superior performance metrics such as mIoU and Dice score, while maintaining competitive inference efficiency. Ablation studies confirmed the complementary roles of RFEM and CAGM, with notable improvements in mIoU and Dice score when both modules were integrated. Additionally, the model's balance of precision and computational efficiency highlights its potential for real-time deployment in large-scale pavement monitoring systems.

Keywords: pavement distress, segmentation, deep learning, cracks, Context-CrackNet, region-focused enhancement, global context modeling

*Corresponding author

1. Introduction

Transportation infrastructure is essential to modern society, forming the backbone of economic development by enabling the efficient movement of people and goods. Among these infrastructures, road networks are critical, and maintaining their integrity is imperative to ensure public safety and economic efficiency. Pavement distresses such as cracks and potholes which develop on these road networks not only compromise safety but also lead to costly repairs if not promptly detected and addressed. Accurately detecting these distresses remains a significant challenge due to their irregular shapes, varying sizes, diverse surface textures, and environmental factors such as fluctuating lighting conditions and the presence of debris. Traditionally, the detection of these distresses has relied on manual inspections, which are not only time-consuming and labor-intensive but also prone to subjectivity and inconsistency. These limitations underscore the critical need for automated solutions to improve efficiency and accuracy. To address these challenges, researchers have increasingly turned to automated approaches that leverage advanced image processing and machine learning techniques, offering a more robust and scalable alternative to traditional methods. While early image processing approaches were often inadequate due to the complex nature of pavement surfaces, the introduction of deep learning models particularly convolutional neural networks has significantly advanced the field. These models effectively identify and segment pavement distresses by learning hierarchical features and capturing spatial context. Nevertheless, despite these advancements, current models still face significant limitations that hinder their performance, necessitating further refinements to achieve accurate pavement distress segmentation.

Several deep learning approaches have been proposed for pavement distress segmentation. For instance, Wen et al. proposed PDSNet [1], an efficient framework achieving an MIoU of 83.7% on manually collected 2D and 3D private pavement dataset. However, it struggles with small and tiny cracks . Sarmiento [2] utilized YOLOv4 for detecting and DeepLabv3 for segmenting the pavement distresses. While effective for simpler distresses like delaminations, both models struggle with tiny cracks, scaling, and texture variations, leading to misclassifications and false negatives. Li et al. [3] further introduced a variant of DeepLabV3+ with an adaptive probabilistic sampling method and external attention for pavement distress segmentation. It was evaluated on the CRACK500 dataset, achieving a Mean Intersection over

Union (MIoU) of 54.91%. Tong et al. introduced Evidential Segmentation Transformer (ES-Transformer) [4], combining DempsterShafer theory with a transformer backbone for improved segmentation and calibration. Evaluated on the Crack500 dataset, it achieved a Mean Intersection over Union (MIoU) of 59.85%, demonstrating superior performance. However, the architecture introduced is computationally expensive since the transformer architecture used scales quadratically with the input data. Kyem et al. [5] used YOLOv8 and the Segment Anything Model (SAM) for segmentation in their PaveCap framework, utilizing SAMs zero-shot capability for generating binary masks. However, the method struggled with accurately segmenting mixed or overlapping pavement distresses. Owor et al. also introduced PaveSAM [6], a zero-shot segmentation model fine-tuned for pavement distresses using bounding box prompts, significantly reducing labeling costs and achieving strong performance. One problem with this model is its inability to segment fine-grained distresses in the pavement images.

Despite the significant progress achieved with deep learning models for pavement distress segmentation, several limitations remain. ne critical yet unresolved challenge is the accurate segmentation of tiny and small pavement cracks. Tiny cracks typically refer to pavement cracks narrower than 1 mm. These cracks are faint, often discontinuous, and extremely challenging to detect, particularly under varying environmental conditions [7, 8, 9]. Small cracks, slightly larger, range from about 1 mm to 3 mm in width [9]. Although more visible than tiny cracks, small cracks still pose significant challenges for accurate segmentation due to their irregular shapes, textures, and subtle appearance in images. Identifying these small and tiny defects early enables preventive maintenance before they develop into more extensive damage. By intervening at this early stage, maintenance teams can prevent minor issues from escalating, thereby reducing repair costs and minimizing disruptions to traffic. To achieve these outcomes, it is essential to adopt advanced models capable of effectively handling both very small and larger cracks. However, achieving this goal is not without challenges, as many existing models struggle with multi-scale feature representation, which hinders their ability to effectively detect both small-scale and large-scale cracks [10]. In addition to this challenge is the lack of a comprehensive understanding of global context which often limits the models ability to capture large-scale spatial relationships and distinguish between interconnected distresses and noise. This results in inconsistent segmentation of extensive distress patterns such as longitudinal and alligator cracks. Furthermore, many deep learning

models require high-resolution inputs to detect subtle crack features, significantly increasing memory usage and inference time [11]. This computational burden limits their suitability for real-time deployment and scalability for large-scale pavement monitoring systems. These above limitations highlight the pressing need for innovative approaches to enhance segmentation performance and address the shortcomings of existing methods.

To address the challenges of pavement distress segmentation, we propose Context-CrackNet, an encoder-decoder architecture built around two key innovations: the Region-Focused Enhancement Module (RFEM) and the Context-Aware Global Module (CAGM). The RFEM, embedded in the decoder pathway, prioritizes fine-grained features, enabling precise segmentation of small and tiny cracks. This ensures early detection of subtle distresses that traditional models often miss. The CAGM, positioned before the bottleneck, captures global context efficiently by integrating linear self-attention into its design. This allows the model to process high-resolution images and segment larger cracks, such as longitudinal and alligator cracks, without excessive computational costs.. The main contributions of our research has been outlined below:

- Proposed **Context-CrackNet**, a novel efficient architecture that integrates specialized modules designed for comprehensive crack detection at varying scales in high-resolution pavement images.
- Developed a **Region-Focused Enhancement Module** (RFEM) that employs targeted feature enhancement to capture fine-grained details of subtle cracks, enabling precise segmentation of small-scale pavement distress patterns.
- Introduced a **Context-Aware Global Module** (CAGM) that utilizes global contextual information to effectively identify and segment large-scale distress patterns while maintaining computational efficiency across high-resolution images.
- Trained and evaluated our proposed architecture on 10 publicly available crack datasets alongside several existing state-of-the-art segmentation models. Our proposed architecture consistently outperformed these models, achieving state-of-the-art performance across all benchmarks.

2. Related Works

Early attempts at pavement crack detection primarily relied on low-level image properties such as gradient, brightness, shape, and texture, as well as pixel intensity variance, edge orientation, and local binary patterns. Classic edge detection algorithms such as Sobel and Canny [12], Gabor filter-based methods [13, 14, 15], Prewitt [16], Roberts Cross [16], and Laplacian of Gaussian [17] identified crack characteristics by examining intensity variations and local directional patterns. Some researchers also employed threshold-based strategies such as Adaptive and Localized Thresholding [18, 19, 20], Triple-Thresholds Approach [21], Otsu thresholding [22, 23, 24] and wavelet transformations [25] to isolate cracks from background textures. Building on these foundations, early machine learning approaches [26, 27, 28, 29, 30, 31, 32, 33] framed crack extraction as a classification problem, distinguishing between crack and non-crack pixels using hand-engineered features.

However, these traditional methods often struggled with generalization [34]. Differences between training and testing datasets commonly led to performance degradation, and detecting tiny cracks proved particularly challenging. Moreover, the reliance on handcrafted features made these methods sensitive to environmental variations and morphological differences in cracks [35]. Noise and other forms of interference further undermined their stability and applicability.

With the rapid advancement of deep learning and its application in pavement assessment [36, 37], researchers turned to Convolutional Neural Networks (CNNs) [38] to develop more robust solutions. CNNs excel at feature extraction, inspiring the creation of models for crack image classification, crack detection and crack segmentation. For instance, Li et al. [39] proposed a CNN-based method to classify pavement patches into five categories using 3D pavement images, demonstrating high accuracy in distinguishing between the various crack types. Zhang et al. [40] developed a deep convolutional neural network (ConvNet) for pavement crack detection, learning features directly from raw image patches and outperforming traditional hand-crafted approaches. Subsequently, Liu et al. [41] proposed DeepCrack, a deep hierarchical CNN for pixel-wise crack segmentation, incorporating multi-scale feature fusion, deeply-supervised nets, and guided filtering learning methods that steadily improved segmentation performance.

Despite these advancements, significant challenges persisted. Many state-of-the-art methods emphasized performance metrics but gave limited atten-

tion to extracting subtle, tiny crack features. Additionally, as models became more complex, their computational and memory requirements increased, hindering deployment on resource-limited devices. Recognizing these issues, researchers began exploring lightweight model architectures capable of balancing detection accuracy with efficiency.

Li et al. [42] proposed CarNet, a lightweight encoder-decoder achieving an ODS F-score of 0.514 on Sun520 with an inference speed of 104 FPS, balancing performance and efficiency. Similarly, Zhou et al. [43] introduced LightCrackNet, a lightweight crack detection model designed to optimize performance and efficiency. The model utilizes Split Exchange Convolution (SEConv) and Multi-Scale Feature Exchange (MSFE) modules, achieving an F1-score of 0.867 DeepCrack dataset with only 1.3M parameters and 8 GFLOPs.

Nevertheless, significant hurdles remain in achieving high-performance, efficient crack detection. Although CNN-based methods excel at extracting local features, they struggle to aggregate global context when used alone, limiting their ability to identify tiny or small cracks. Traditional self-attention transformer models offer a promising solution for capturing global relationships, but their quadratic scaling with input data makes them computationally intensive and impractical for certain applications. Linear self-attention transformers [44, 45, 46] when used with CNNs address this challenge by reducing computational complexity, enabling efficient integration of global and local cues while focusing on extracting minute crack featuresa key objective in the field. Some researchers have applied linear self-attention modules in different domains. For instance, Fang et al. [47] proposed CLFormer, a lightweight transformer combining convolutional embedding and linear self-attention (LSA) for bearing fault diagnosis. It achieves strong robustness and high accuracy under noise and limited data, with only 4.88K parameters and 0.12 MFLOPs. Guo et al. [48] introduced External Attention, a lightweight mechanism with linear complexity using two learnable memory layers, enhancing generalization and efficiency across visual tasks like segmentation and detection.

3. Method

3.1. Problem Structure and Overview

Detecting small and tiny cracks in pavement images is particularly challenging due to their subtle features, irregular shapes, and the presence of

noise such as shadows and debris. Existing deep learning models often struggle with capturing these fine-grained details, underscoring the need for more precise and efficient segmentation approaches. The problem definition has been formulated mathematically below.

Let $I \in \mathbb{R}^{H\times W\times C}$ represent a pavement image, where H, W, and C denote the height, width, and number of channels, respectively. The goal is to predict a segmentation map $\hat{S} \in \mathbb{R}^{H\times W\times K}$, where K represents the number of classes, including the background. Mathematically, this can be expressed as learning a mapping function $f : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W\times K}$, such that:

$$\hat{S} = f(I; \theta), \tag{1}$$

where θ denotes the learnable parameters of the segmentation model. The task involves accurately localizing and classifying pavement distresses of varying scales, shapes, and textures.

The proposed architecture, *Context-CrackNet*, addresses these challenges by introducing two novel components: the Context-Aware Global Module (CAGM) and the Region-Focused Enhancement Module (RFEM). The CAGM ensures efficient global context modeling, enabling the network to capture long-range dependencies and large-scale spatial relationships. The RFEM enhances the network's ability to focus on fine-grained details, ensuring precise segmentation of small and subtle distresses. Together, these components form a robust encoder-decoder framework optimized for pavement distress segmentation.

3.2. Overall Framework

The overall structure of *Context-CrackNet* integrates global and local feature refinement seamlessly, providing a balanced approach to segmentation. The network adopts an encoder-decoder structure, where the encoder extracts hierarchical features from the input image, the bottleneck incorporates global attention mechanisms, and the decoder reconstructs the segmentation map using refined skip connections.

The encoder is based on a ResNet50 backbone, which extracts features at multiple levels of abstraction. For an input image I, the encoder produces a sequence of feature maps:

$$\Phi_{\text{enc}}(I) = \{F_0, F_1, F_2, F_3, F_4\}, \tag{2}$$

where $F_0 \in \mathbb{R}^{H/2\times W/2\times 64}$ represents low-level spatial features, and $F_4 \in \mathbb{R}^{H/32\times W/32\times 2048}$ captures high-level semantic information. These feature

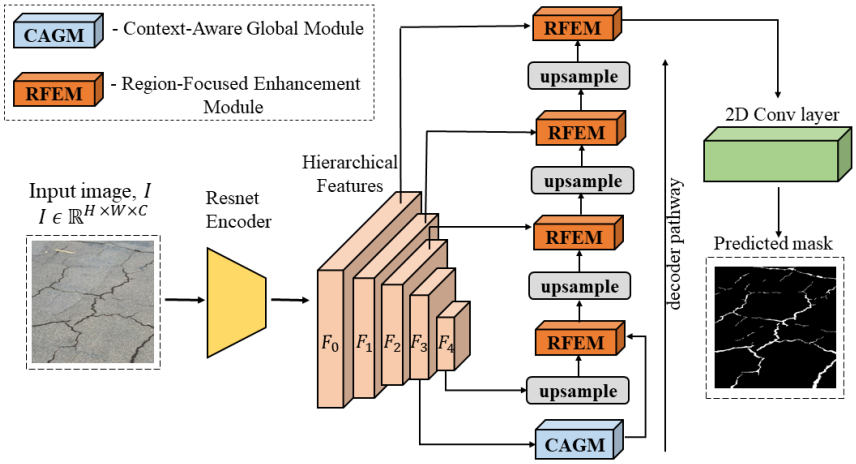

Figure 1: Overall Architecture of *Context-CrackNet*: The proposed framework adopts an encoder-decoder structure with two novel components: the Context-Aware Global Module (CAGM) and the Region-Focused Enhancement Module (RFEM). The ResNet-based encoder extracts hierarchical features $\{F_0, F_1, F_2, F_3, F_4\}$, where F_3 is processed by the CAGM to model global contextual relationships and generate the contextualized feature map. The decoder pathway integrates RFEMs at each stage, which refine the skip connections between encoder features and upsampled decoder outputs. This refinement enables effective feature modulation for precise segmentation. Finally, the decoder outputs the predicted segmentation mask $\hat{S}$, capturing fine-grained pavement distress details.

maps progressively encode spatial and contextual details, forming the foundation for subsequent processing stages.

At the bottleneck, the feature map F_3, which encapsulates rich semantic information, is passed through the Context-Aware Global Module (CAGM). The CAGM uses a linear self-attention mechanism to model long-range dependencies efficiently, reducing the computational complexity typically associated with traditional self-attention. This produces an enhanced feature map F_{CAGM}, expressed mathematically as:

$$F_{\text{CAGM}} = f_{\text{CAGM}}(F_3), \tag{3}$$

where f_{CAGM} represents the operations within the module.

In the decoder pathway, the Region-Focused Enhancement Module (RFEM) plays a critical role in refining the skip connections between the encoder and

decoder. For each decoder stage l, the RFEM processes the corresponding encoder feature map $F_{e,l}$ and the upsampled feature map $F_{d,l+1}$ from the previous decoder stage. The refined feature map $F_{\text{RFEM},l}$ is computed as:

$$F_{\text{RFEM},l} = f_{\text{RFEM}}(F_{e,l}, F_{d,l+1}), \tag{4}$$

where f_{RFEM} represents the attention mechanism used to focus on the most relevant spatial regions. This refinement ensures that critical features are emphasized while irrelevant activations are suppressed.

The decoder reconstructs the segmentation map by iteratively combining the refined features from the RFEM with the upsampled feature maps. Starting with the output of the CAGM, the decoder applies a series of upsampling and refinement operations to produce the final segmentation map:

$$\hat{S} = f_{\text{decoder}}(F_{\text{RFEM}}), \tag{5}$$

where f_{decoder} represents the decoding operations, including upsampling, concatenation, and convolution.

This framework effectively addresses the challenges associated with multi-scale feature representation and computational efficiency. By combining the strengths of the CAGM and RFEM, *Context-CrackNet* achieves a balance between global context understanding and fine-grained detail enhancement, enabling robust and accurate pavement distress segmentation. The subsequent sections goes deeper into the mathematical details and implementation of the CAGM and RFEM modules.

3.3. Context-Aware Global Module (CAGM)

The **Context-Aware Global Module (CAGM)** addresses the challenge of capturing long-range dependencies and global contextual relationships in the feature map F_3. This capability is crucial for accurately segmenting large-scale pavement distresses, such as longitudinal and alligator cracks, which require an understanding of spatial relationships across distant regions. To achieve this, the CAGM employs a linear self-attention mechanism, reducing the quadratic complexity of traditional self-attention to linear, thereby enabling efficient processing of high-resolution images.

Let $F_3 \in \mathbb{R}^{B\times C\times H\times W}$ represent the input feature map at the bottleneck stage, where B is the batch size, C is the number of channels, and H, W are the spatial dimensions. The first step in the CAGM is to transform the

Algorithm 1 Context-CrackNet Framework

Input image $I \in \mathbb{R}^{H \times W \times C}$
Predicted segmentation mask $\hat{S} \in \mathbb{R}^{H \times W \times K}$
Stage 1: Encoder Pathway
$\{F_0, F_1, F_2, F_3, F_4\} \leftarrow \text{ResNetEncoder}(I)$ ▷ Extract features
Stage 2: Bottleneck Processing
$F_{\text{CAGM}} \leftarrow \text{CAGM}(F_3)$ ▷ Global context modeling
Stage 3: Decoder Pathway
Initialize $D_4 \leftarrow F_{\text{CAGM}}$ $l \in \{3, 2, 1, 0\}$
$D_{\text{up}} \leftarrow \text{Upsample}(D_{l+1})$ ▷ $2\times$ spatial size
$D_l \leftarrow \text{RFEM}(F_l, D_{\text{up}})$ ▷ Feature refinement
Stage 4: Final Prediction
$\hat{S} \leftarrow \text{Conv2D}(D_0)$ ▷ K-class prediction
$\hat{S}$

spatial dimensions H and W into a sequence of length $N = H \times W$. This can be mathematically expressed as:

$$X_b = \{F_3[b, :, i, j] \mid i \in \{1, \ldots, H\}, j \in \{1, \ldots, W\}\}, \quad X \in \mathbb{R}^{B \times N \times C}, \quad (6)$$

where X_b is the reshaped feature map for the b-th sample in the batch, and X concatenates all spatial positions into a sequence.

The sequence X is projected into query (Q), key (K), and value (V) spaces using learned linear transformations:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \quad (7)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{C \times d_k}$ are learnable weight matrices, and d_k is the dimensionality of the query and key vectors.

To reduce the computational cost, the key and value matrices are projected into lower-dimensional spaces:

$$K_{\text{proj}} = KE, \quad V_{\text{proj}} = VF, \quad (8)$$

where $E, F \in \mathbb{R}^{N \times k}$ are learnable projection matrices, and $k \ll N$ is the reduced dimension of the projected key and value spaces.

The attention weights are computed as:

$$A = \text{softmax}\left(\frac{QK_{\text{proj}}^{\top}}{\sqrt{d_k}}\right), \quad A \in \mathbb{R}^{B \times N \times k}, \quad (9)$$

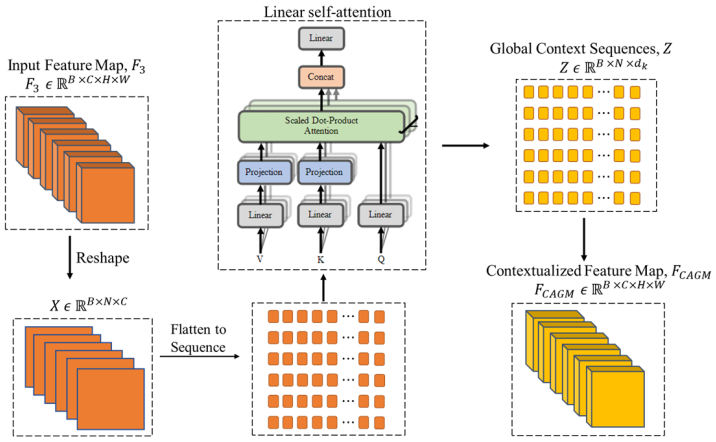

Figure 2: Context-Aware Global Module (CAGM): The module processes the input feature map $F_3 \in \mathbb{R}^{B\times C\times H\times W}$, reshaping it into a sequence $X \in \mathbb{R}^{B\times N\times C}$, where $N = H \times W$. Using a **Linear Self-Attention Mechanism**, query (Q), key (K), and value (V) projections generate **Global Context Sequences** ($Z \in \mathbb{R}^{B\times N\times d_k}$). These sequences are reconstructed into the **Contextualized Feature Map** ($F_{\text{CAGM}} \in \mathbb{R}^{B\times C\times H\times W}$), embedding global dependencies efficiently.

where softmax ensures the weights A sum to 1 across the key dimension for each query.

The weighted output is then computed by aggregating the projected values:

$$Z = AV_{\text{proj}}, \quad Z \in \mathbb{R}^{B\times N\times d_k}, \tag{10}$$

Finally, the output sequence Z is mapped back to the original channel dimension and reconstructed into its spatial structure:

$$F_{\text{CAGM}}[b, :, i, j] = Z[b, n]W_O, \quad n = (i-1) \times W + j, \tag{11}$$

where $W_O \in \mathbb{R}^{d_k\times C}$ is a learnable weight matrix, and $F_{\text{CAGM}} \in \mathbb{R}^{B\times C\times H\times W}$ is the enhanced feature map.

By explicitly modeling global relationships across spatial regions, the CAGM integrates information from distant parts of the pavement image, enabling the network to detect and segment large-scale cracks and patterns. Its efficient linear self-attention mechanism ensures scalability, making it suitable for high-resolution images while maintaining computational feasibility.

Rationale for Using Linear Self-Attention. Traditional self-attention has a quadratic complexity $\mathcal{O}(N^2)$, which limits its scalability to high-resolution

Algorithm 2 Context-Aware Global Module (CAGM)

Require:
Input feature map $F_3 \in \mathbb{R}^{B\times C\times H\times W}$
Ensure:
Contextualized feature map $F_{\text{CAGM}} \in \mathbb{R}^{B\times C\times H\times W}$
Feature Reshaping:
$X \leftarrow \text{Reshape}(F_3)$ ▷ $X \in \mathbb{R}^{B\times N\times C}$, where $N = H \times W$
Linear Self-Attention:
$Q \leftarrow \text{Linear}(X)$ ▷ Query projection
$K \leftarrow \text{Linear}(X)$ ▷ Key projection
$V \leftarrow \text{Linear}(X)$ ▷ Value projection
$A \leftarrow \text{SDP-Attention}(Q, K, V)$ ▷ Scaled dot-product attention
$Z \leftarrow \text{Concat}[A, X]$ ▷ $Z \in \mathbb{R}^{B\times N\times d_k}$
$Z \leftarrow \text{Linear}(Z)$ ▷ Final projection
Global Context Reconstruction:
$F_{\text{CAGM}} \leftarrow \text{Reshape}(Z, [B, C, H, W])$ ▷ Restore spatial dimensions
return F_{CAGM}
Note: SDP-Attention computes $\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$

images. In contrast, the linear self-attention used in the CAGM reduces this to $\mathcal{O}(N \cdot k)$ by projecting keys and values into a lower-dimensional space, enabling efficient global context modeling. This is particularly important for pavement images, where cracks can span large, non-contiguous regions. The CAGM allows each spatial location to incorporate information from the entire image, improving segmentation of extensive or fragmented crack patterns. Positioned at the bottleneck, it complements the RFEM by enriching features with global cues before fine-grained refinement in the decoder.

3.4. Region-Focused Enhancement Module (RFEM)

The **Region-Focused Enhancement Module (RFEM)** refines the skip connections between the encoder and decoder to enhance the segmentation of fine-grained details such as small and subtle pavement cracks. By dynamically modulating encoder features using spatial context from the decoder, the RFEM ensures that relevant features are emphasized, while irrelevant ones are suppressed.

Let $F_{e,l} \in \mathbb{R}^{B\times C_e\times H_e\times W_e}$ denote the encoder feature map at level l, and

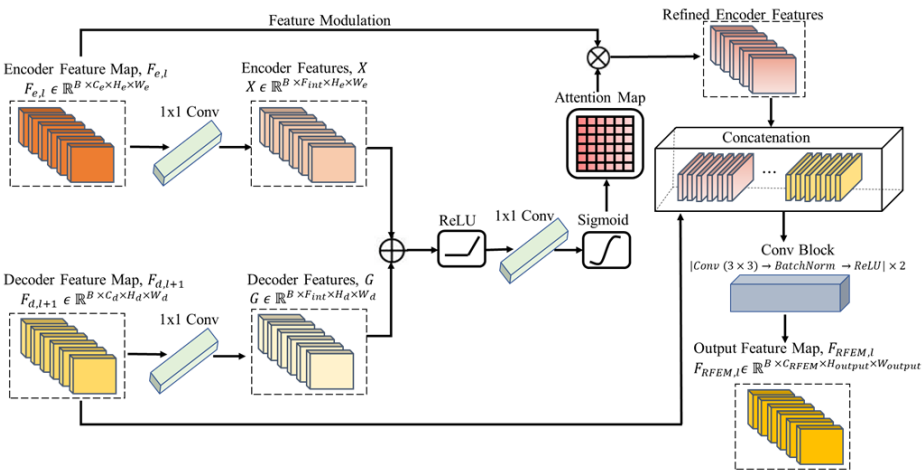

Figure 3: Region-Focused Enhancement Module (RFEM): The module refines encoder features $F_{e,l} \in \mathbb{R}^{B\times C_e\times H_e\times W_e}$ and decoder features $F_{d,l+1} \in \mathbb{R}^{B\times C_d\times H_d\times W_d}$ by transforming them into intermediate features X and G, respectively. These are fused and processed through a ReLU and sigmoid to generate an attention map. The attention-refined encoder features are then concatenated with decoder features and passed through a convolutional block, yielding $F_{\text{RFEM},l}$.

$F_{d,l+1} \in \mathbb{R}^{B\times C_d\times H_d\times W_d}$ the upsampled decoder feature map from the previous stage. To align the features for fusion, both are projected to a shared space:

$$G = F_{d,l+1} * W_g + b_g, \quad X = F_{e,l} * W_x + b_x, \tag{12}$$

where W_g, W_x and b_g, b_x are learnable parameters for 1Œ1 convolutions.

These projected features are fused via element-wise addition, capturing complementary spatial cues. A ReLU activation is then applied to retain positive activations and introduce non-linearity:

$$Y = \max(0, G + X), \tag{13}$$

To localize salient regions, an attention map Ψ is derived by passing the fused features through a convolution followed by a sigmoid activation:

$$\Psi = \frac{1}{1 + \exp\left(-\left(Y * W_\psi + b_\psi\right)\right)}, \tag{14}$$

where W_ψ and b_ψ are learnable parameters. The sigmoid scales attention weights to [0,1], guiding selective emphasis.

The attention map modulates the encoder features via element-wise multiplication:

$$F_{\text{refined}} = \Psi \odot F_{e,l}, \tag{15}$$

where $\odot$ denotes element-wise multiplication, resulting in spatially weighted encoder features.

These refined features are then concatenated with decoder features along the channel dimension:

$$F_{\text{concat}}[b, c, i, j] = \begin{cases} F_{\text{refined}}[b, c, i, j], & \text{if } c < C_{\text{refined}}, \\ F_{d,l+1}[b, c - C_{\text{refined}}, i, j], & \text{otherwise,} \end{cases} \tag{16}$$

ensuring that both attention-enhanced and high-level decoder context are retained.

Finally, a convolutional refinement block $\gamma(\cdot)$, typically composed of two Conv-BatchNorm-ReLU layers, produces the final output:

$$F_{\text{RFEM},l} = \gamma(F_{\text{concat}}), \tag{17}$$

By focusing on critical local features, RFEM strengthens the models ability to detect tiny cracks while complementing the broader context captured by CAGM.

3.5. Loss Functions

The proposed framework uses tailored loss functions to optimize the segmentation task across both binary and multi-class scenarios, ensuring accurate prediction of pavement distress patterns. These loss functions are carefully designed to balance class contributions, address class imbalance, and effectively capture fine-grained details.

3.5.1. Binary Segmentation Loss

For binary segmentation tasks, where the goal is to classify each pixel as either belonging to a crack (1) or not (0), we employ a combination of the Binary Cross Entropy (BCE) loss and the Dice loss. The combined loss is formulated as:

$$\mathcal{L}_{\text{binary}} = \alpha\, \mathcal{L}_{\text{BCE}} + \beta\, \mathcal{L}_{\text{Dice}}, \tag{18}$$

where α and β are weights that control the contribution of each term.

Algorithm 3 Region-Focused Enhancement Module (RFEM)

Require:

Encoder feature map $F_{e,l} \in \mathbb{R}^{B \times C_e \times H_e \times W_e}$

Decoder feature map $F_{d,l+1} \in \mathbb{R}^{B \times C_d \times H_d \times W_d}$

Ensure:

Refined feature map $F_{\text{RFEM},l} \in \mathbb{R}^{B \times C_{\text{RFEM}} \times H_{\text{output}} \times W_{\text{output}}}$

Feature Transformation:

$X \leftarrow \text{Conv}_{1\times1}(F_{e,l})$ ▷ Transform to $\mathbb{R}^{B \times F_{\text{int}} \times H_e \times W_e}$

$G \leftarrow \text{Conv}_{1\times1}(F_{d,l+1})$ ▷ Transform to $\mathbb{R}^{B \times F_{\text{int}} \times H_d \times W_d}$

Attention Map Generation:

$Y \leftarrow \text{ReLU}(G + X)$ ▷ Element-wise addition

$\Psi \leftarrow \sigma(\text{Conv}_{1\times1}(Y))$ ▷ σ: Sigmoid activation

Feature Modulation:

$F_{\text{refined}} \leftarrow \Psi \otimes F_{e,l}$ ▷ Channel-wise multiplication

Feature Fusion:

$F_{\text{concat}} \leftarrow \text{Concat}([F_{\text{refined}}, F_{d,l+1}])$

Conv Block Refinement:

for $i \leftarrow 1$ **to** 2 **do**

    $F_{\text{RFEM},l} \leftarrow \text{ReLU}(\text{BatchNorm}(\text{Conv}_{3\times3}(F_{\text{concat}})))$

end for

return $F_{\text{RFEM},l}$

1. Binary Dice Loss. The Dice loss, designed to handle imbalances in pixel classes, measures the overlap between the predicted segmentation map $\hat{S}$ and the ground truth S.

Let $\hat{S} \in [0,1]^{H\times W}$ and $S \in \{0,1\}^{H\times W}$ denote the predicted and ground truth maps, respectively. The Dice loss is computed as:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} \hat{S}_i S_i + \epsilon}{\sum_{i=1}^{N} \hat{S}_i + \sum_{i=1}^{N} S_i + \epsilon}, \tag{19}$$

where ϵ is a smoothing constant to prevent division by zero, and $N = H \times W$ represents the total number of pixels. The Dice loss encourages the model to maximize the overlap between $\hat{S}$ and S, ensuring robust segmentation of small and subtle cracks.

2. Binary Cross Entropy Loss. The BCE loss penalizes the deviation between predicted probabilities and ground truth labels. It is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[S_i \log(\hat{S}_i) + (1 - S_i)\log(1 - \hat{S}_i)\right], \tag{20}$$

This term provides pixel-wise supervision, complementing the Dice loss by ensuring accurate classification even in cases of severe class imbalance.

3.5.2. Multi-Class Segmentation Loss

For multi-class segmentation tasks, where the pavement image contains multiple types of distresses, we adopt a combined loss function comprising the Cross-Entropy (CE) loss and a multi-class Dice loss:

$$\mathcal{L}_{\text{multi-class}} = \gamma\,\mathcal{L}_{\text{CE}} + \delta\,\mathcal{L}_{\text{Dice}}, \tag{21}$$

where γ and δ are weighting factors.

1. Multi-Class Dice Loss. The Multi-Class Dice Loss extends the principles of the Binary Dice Loss to multi-class segmentation tasks, ensuring fair optimization for each class, including the background. By individually evaluating the overlap between the predicted segmentation map $\hat{S}_k$ and the ground truth map S_k for each class k, it addresses class imbalances and promotes accurate segmentation across all categories.

Let $\hat{S}_k$ and S_k represent the predicted and ground truth maps for class k, respectively, where $k \in \{1, \ldots, K\}$. The Dice loss for class k is defined as:

$$\mathcal{L}_{\text{Dice},k} = 1 - \frac{2\sum_{i=1}^{N}\hat{S}_{k,i}S_{k,i} + \epsilon}{\sum_{i=1}^{N}\hat{S}_{k,i} + \sum_{i=1}^{N}S_{k,i} + \epsilon}, \tag{22}$$

where N denotes the total number of pixels and ϵ is a small constant to avoid division by zero.

The total multi-class Dice loss is calculated as the average Dice loss across all K classes:

$$\mathcal{L}_{\text{Dice}} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{\text{Dice},k}, \tag{23}$$

This formulation encourages precise segmentation across all classes, ensuring small or underrepresented categories are effectively captured.

2. Cross-Entropy Loss. The CE loss measures the pixel-wise classification error, weighted by class frequencies to handle imbalance.

Let ω_k denote the weight for class k, derived from the inverse class frequency. The CE loss is defined as:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\omega_k S_{k,i}\log(\hat{S}_{k,i}), \tag{24}$$

where ω_k ensures that the model does not bias toward dominant classes.

4. Experiments

To validate the effectiveness of the proposed *Context-CrackNet*, we conduct comprehensive experiments designed to evaluate its performance on pavement distress segmentation tasks. This section details the datasets used, the implementation specifics of *Context-CrackNet*, and the experimental setup. Furthermore, ablation studies are performed to assess the contribution of each module and design choice to the overall performance. Finally, we compare *Context-CrackNet* with state-of-the-art methods to highlight its advantages in addressing the challenges of fine-grained and multi-scale pavement distress detection.

4.1. Dataset

To train and evaluate the proposed *Context-CrackNet*, we utilize 10 publicly available binary crack datasets: CFD [49], Crack500 [50], CrackTree200 [51], DeepCrack [41], Eugen Miller [52], Forest [53], GAPs [54], Rissbilder [55], Sylvie [56], and Volker [55]. These datasets comprehensively cover diverse crack detection scenarios, including road cracks, concrete cracks in tunnels, and wall cracks. This diversity demonstrates the robustness of *Context-CrackNet* beyond pavement applications, highlighting its potential to handle various types of cracks across different construction materials and structures.

The datasets exhibit a significant class imbalance, with approximately 97.2% of pixels belonging to the background class and only 2.8% corresponding to the crack class. This imbalance reflects the real-world challenges of identifying subtle cracks against vast background regions.

All the images across the datasts have a resolution of 448×448 pixels. The datasets were then split into training and testing sets with an 80:20 ratio, ensuring a balanced distribution of samples across both sets.

Table 1: Datasets and their corresponding crack types

Dataset name	**Crack type**
CFD	Road crack
CRACK500	Road crack
CrackTree200	Road crack
DeepCrack	Road crack
Forest	Road crack
GAPs	Road crack
Sylvie	Road crack
Rissbilder	Wall crack
Volker	Wall crack
Eugen Miller	Concrete crack on Tunnels

To further enhance the diversity of the training data and improve the model's generalization capabilities, a series of data augmentation techniques were employed. These augmentations included spatial transformations such as horizontal and vertical flips, random rotations by 90°, and shift-scale-rotate operations, which varied the spatial properties of the images while maintaining their structural integrity. Additionally, pixel-level augmentations such as Gaussian noise and color jittering were applied to introduce variations in brightness, contrast, saturation, and hue, simulating real-world variations in lighting and camera conditions.

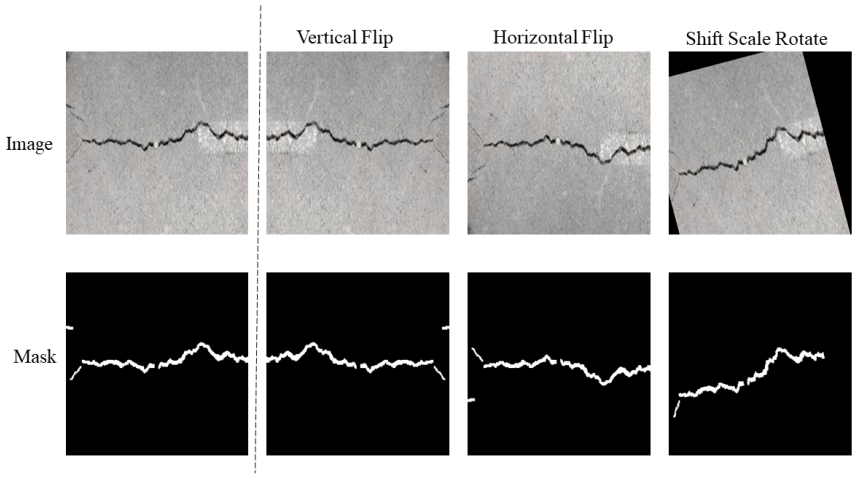

Figure 4: Examples of data augmentation techniques applied to crack images and their corresponding masks in the DeepCrack dataset. Augmentations include vertical flip, horizontal flip, and shift-scale-rotate, showcasing the spatial transformations employed to enhance diversity and robustness in the training dataset. The top row illustrates augmented images, while the bottom row presents their respective masks.

For preprocessing, the images were normalized using the mean and standard deviation values of the ResNet backbone: $\mu(0.485, 0.456, 0.406)$ and $\sigma(0.229, 0.224, 0.225)$, respectively. This normalization step ensures compatibility with the pre-trained ResNet model used in the encoder.

By employing a diverse dataset that encompasses road, wall, and concrete cracks, the proposed framework is equipped to handle the complexities of real-world crack detection scenarios, addressing challenges such as class imbalance, varying scales, and noise effectively.

4.1.1. Custom Dataset

4.2. Implementation details

4.2.1. Training settings

The proposed *Context-CrackNet* was implemented using the PyTorch framework. The AdamW [57] optimizer was used for training, with a weight decay of 1×10^{-5} to prevent overfitting. The initial learning rate was set to 1×10^{-4} , and a using an adaptive learning rate scheduler was applied to adjust the learning rate dynamically. This scheduler reduced the learning rate by a factor of 0.5 after 5 epochs of no improvement in validation loss.

All models, including Context-CrackNet and the baseline models (U-Net, U-Net++, DeepLabV3, DeepLabV3+, FPN, PSPNet, LinkNet, MAnet, and PAN), were trained separately on each of the ten datasets. Each dataset was randomly split into training and validation sets, ensuring a consistent evaluation setup for every model. After training, each model was evaluated on the corresponding validation set of each dataset to measure performance.

The training was performed with a batch size of 32 over a total of 1000 epochs. All experiments were conducted on an NVIDIA A40 GPU with 48GB of memory, providing sufficient computational power to efficiently handle high-resolution images. Table 1 summarizes the training configurations used for all experiments.

4.2.2. Evaluation metrics

To comprehensively evaluate the performance of the proposed *Context-CrackNet* on pavement distress segmentation tasks, we employed the following segmentation metrics: Intersection over Union (IoU) score, Dice score, Precision, Recall, and F1 score. These metrics provide a holistic assessment of the models ability to accurately segment fine-grained and multi-scale pavement cracks, balancing considerations of overlap, correctness, and completeness.

Mean Intersection over Union (mIoU). The mean Intersection over Union (mIoU) evaluates the average overlap between the predicted segmentation map $\hat{S}_k$ and the ground truth S_k across all classes k. For a single class k, the IoU is defined as:

$$\mathrm{IoU}_k = \frac{|\hat{S}_k \cap S_k|}{|\hat{S}_k \cup S_k|} = \frac{\sum_{i=1}^{N} \hat{S}_{k,i} S_{k,i}}{\sum_{i=1}^{N} \left(\hat{S}_{k,i} + S_{k,i} - \hat{S}_{k,i} S_{k,i} \right)}, \tag{25}$$

where N denotes the total number of pixels, and $\hat{S}_{k,i}, S_{k,i} \in \{0, 1\}$ represent the predicted and ground truth labels for pixel i in class k. The mIoU is then computed as the average IoU across all K classes:

$$\mathrm{mIoU} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{IoU}_k, \tag{26}$$

This metric provides a comprehensive assessment of segmentation performance by considering the overlap for all classes and averaging them to yield a single performance score.

Dice Score. The Dice score, also known as the SørensenDice coefficient, quantifies the overlap between the predicted and ground truth segmentation maps. It is defined as:

$$\text{Dice} = \frac{2|\hat{S} \cap S|}{|\hat{S}| + |S|} = \frac{2\sum_{i=1}^{N} \hat{S}_i S_i}{\sum_{i=1}^{N} \hat{S}_i + \sum_{i=1}^{N} S_i}. \tag{27}$$

Dice score emphasizes the correct segmentation of smaller regions, making it particularly useful for evaluating fine-grained crack details.

Precision. Precision measures the proportion of correctly identified crack pixels to the total predicted crack pixels. It is defined as:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}. \tag{28}$$

Recall. Recall quantifies the ability of the model to detect all crack pixels in the ground truth. It is defined as:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}. \tag{29}$$

F1 Score. The F1 score provides a harmonic mean of Precision and Recall, balancing their trade-offs. It is expressed as:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{30}$$

These metrics collectively evaluate the models segmentation performance, ensuring both spatial accuracy (IoU, Dice) and the balance between prediction correctness and completeness (Precision, Recall, F1).

Table 2: Model training settings.

Name	**Training setting**
Optimizer	AdamW
Learning Rate	1×10^{-4}
Batch Size	32
Weight Decay	1×10^{-5}
Number of Epochs	1000

4.2.3. Comparison with other methods

To evaluate the performance of the proposed *Context-CrackNet*, we compared it against state-of-the-art segmentation models, including U-Net [58], U-Net++ [59], DeepLabV3 [60], DeepLabV3+ [61], FPN [62], PSPNet [63], LinkNet [64], MAnet [65], and PAN [66]. All models were trained and evaluated on the same 10 crack detection datasets (see Section 4.1) under identical experimental conditions to ensure fairness. Each method used a ResNet50 encoder pre-trained on ImageNet, consistent with *Context-CrackNet*.

The datasets presented diverse challenges such as varying crack patterns, scales, and noise levels, providing a robust basis for comparison. Standardized training settings, including preprocessing, augmentations, and hyperparameters, ensured that performance differences reflected the strengths of the architectures rather than experimental inconsistencies.

5. Results and Discussion

This section presents the results of the proposed *Context-CrackNet* across multiple datasets. Both qualitative and quantitative evaluations are discussed, highlighting the model's performance in comparison with existing state-of-the-art methods.

5.1. Qualitative Analysis of Predictions

In this section, we analyze the qualitative results of *Context-CrackNet* compared to existing models, including MAnet, PSPNet, DeepLabV3+, and FPN. These results are evaluated across the ten diverse datasets containing different types of cracks, such as road cracks, wall cracks, and concrete cracks. The goal is to assess each models ability to detect both prominent and tiny cracks, which are critical for reliable structural assessment.

Figure 5 shows segmentation results from various datasets. The first column displays the original crack images, followed by their ground truth masks. The subsequent columns show the predictions from *Context-CrackNet* and other models. Each row represents a dataset, showcasing the models performance across different types of cracks.

The predictions demonstrate that *Context-CrackNet* performs consistently better in detecting fine, small, and subtle cracks compared to the other models. For instance, in the CRACK500 dataset, *Context-CrackNet* successfully identifies the small, interconnected cracks, which competing models often fail to detect. A similar trend is observed in the Rissbilder dataset, where

Context-CrackNet captures the thin wall cracks more accurately, while other models struggle with false positives or incomplete predictions.

5.1.1. Performance on Diverse Crack Types

The results emphasize *Context-CrackNet*'s adaptability to different crack types and contexts. In the DeepCrack dataset, characterized by dense and complex crack patterns, *Context-CrackNet* captures the overall structure of the cracks more effectively than models like PSPNet and DeepLabV3+, which tend to miss faint connections. Likewise, in the Eugen Miller dataset featuring tunnel cracks, *Context-CrackNet* produces cleaner and more detailed predictions, showing its robustness on surfaces with uniform textures.

5.1.2. Tiny Crack Detection and Generalization

A key strength of *Context-CrackNet* is its ability to detect tiny cracks that are often missed by other models. The red-marked areas in the predictions from competing models highlight their failure to consistently capture these smaller cracks. This reinforces the effectiveness of *Context-CrackNet*'s context-aware design, which allows it to focus on both small-scale details and larger crack patterns. This capability addresses the central challenge of detecting tiny cracks, which are crucial for identifying early signs of structural damage.

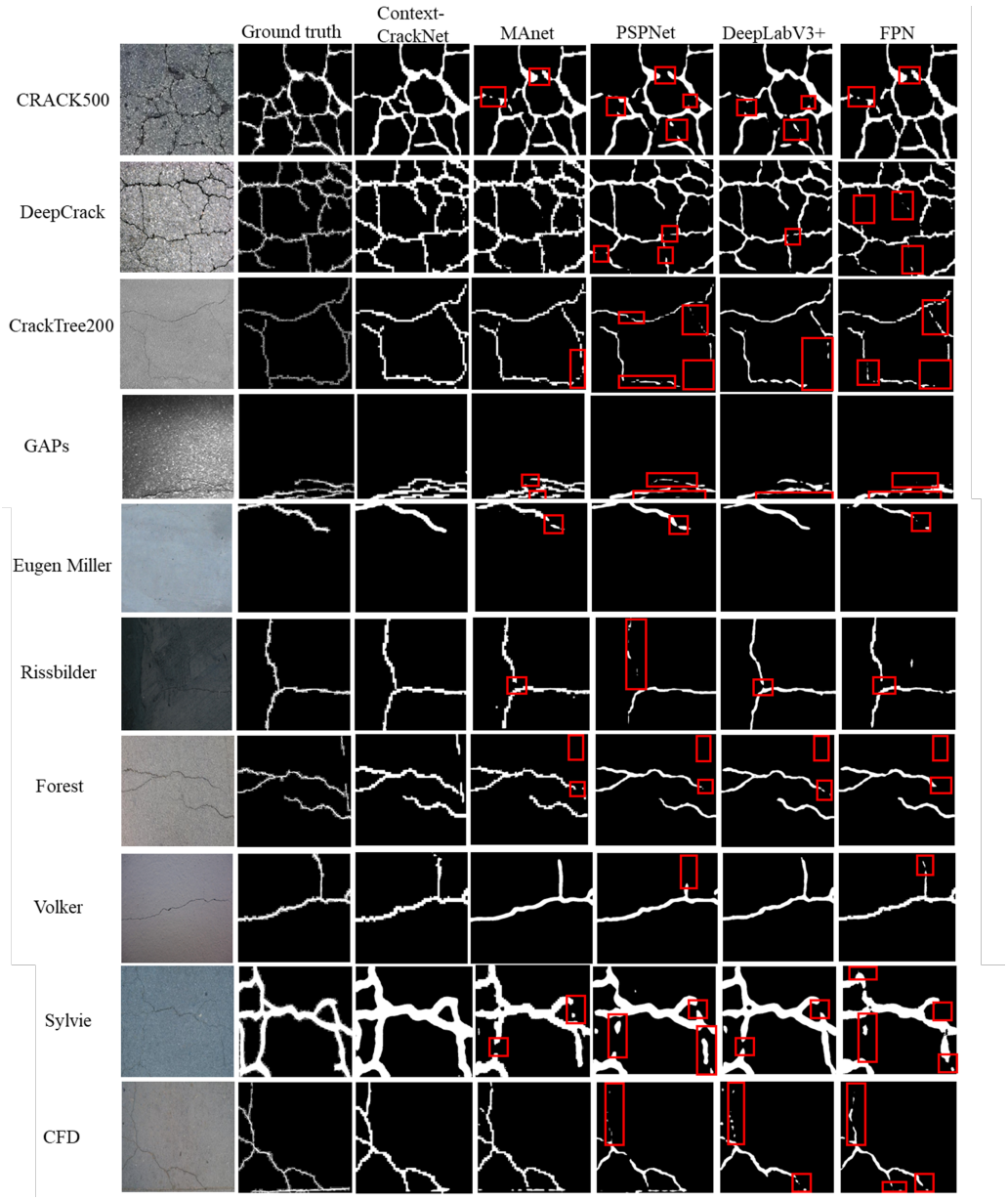

Figure 5: Visual comparison of crack detection results across various datasets. The first column displays the input images, followed by the ground truth masks in the second column. The third column shows the predictions of the proposed *Context-CrackNet*, while subsequent columns present predictions from comparison models including MAnet, PSPNet, DeepLabV3+, and FPN. Rows correspond to individual datasets (e.g., CRACK500, DeepCrack, CrackTree200, etc.). The red boxes highlight areas where comparison models fail to detect tiny cracks effectively, demonstrating the superior performance of *Context-CrackNet* in accurately capturing fine-grained crack details.

5.2. Quantitative Results

The quantitative results in Table 3 demonstrate that *Context-CrackNet* effectively addresses the challenge of detecting tiny and complex cracks, often missed by existing models. On the CFD dataset, it achieves the highest mIoU (**0.5668**) and Dice Score (**0.7235**), along with a recall of **0.8989**, showcasing its ability to capture subtle crack details where models like DeepLabV3 and FPN fall short.

In the CRACK500 dataset, which features diverse crack patterns, *Context-CrackNet* outperforms others with an mIoU of **0.6733** and Dice Score of **0.8046**, demonstrating robust generalization to varying pavement conditions. Similarly, on CrackTree200, characterized by sparse and narrow cracks, it achieves the highest recall (**0.9386**) and Dice Score (**0.6992**), proving its sensitivity to fine-grained cracks that models such as PAN and DeepLabV3Plus often miss.

The DeepCrack dataset further highlights Context-CrackNets strengths, achieving an mIoU of **0.7401** and Dice Score of **0.8505**, validating its ability to detect tiny and small crack patterns with high precision. For non-road cracks in datasets like Eugen Miller and Rissbilder, *Context-CrackNet* shows strong adaptability with recalls of **0.9434** and **0.8469**, outperforming models that struggle with material and texture variations.

On datasets like GAPs and Forest, where irregular crack features dominate, *Context-CrackNet* consistently achieves superior metrics, confirming its effectiveness in challenging environments. Finally, on Sylvie and Volker, it maintains top performance with mIoUs of **0.6846** and **0.7668**, demonstrating its ability to handle varying complexities and environmental conditions. Figure 6 compares *Context-CrackNet* with other state-of-the-art models across the various crack datasets, highlighting Validation IoU, Dice Score, and Recall.

These results affirm that *Context-CrackNet* effectively addresses the limitations of existing models by reliably detecting tiny and complex cracks across diverse datasets, reinforcing its potential for real-world applications in crack detection and infrastructure monitoring.

5.3. Statistical Analysis

To begin our analysis, we first calculated the mean and standard deviation of the IoU and Dice Score across all datasets for each model, as summarized in Table 4. Although these descriptive statistics provide an initial sense of

Table 3: Validation Results of *Context-CrackNet* and other models across all datasets. Each sub-table groups 4 metrics (mIoU, Dice, Recall, Precision) per dataset. Bold red values indicate the highest score in that metric for the dataset.

	CFD				CRACK500				CrackTree200				DeepCrack			
Model	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**
DeepLabV3	0.3194	0.4842	0.5065	0.4638	0.6420	0.7817	0.7612	0.8035	0.3174	0.4819	0.4305	0.5471	0.6813	0.8102	0.8246	0.7963
DeepLabV3Plus	0.3848	0.5558	0.5039	0.6195	0.6376	0.7784	0.7529	0.8060	0.2789	0.4361	0.3823	0.5077	0.6639	0.7977	0.7570	**0.8433**
FPN	0.4187	0.5902	0.5033	0.7134	0.6317	0.7740	0.7460	0.8047	0.2958	0.4566	0.3736	0.5869	0.6783	0.8081	0.7796	0.8387
Linknet	0.4664	0.6362	0.5951	0.6833	0.6450	0.7839	0.7855	0.7827	0.4807	0.6493	0.7066	0.6006	0.7047	0.8267	0.8402	0.8137
MAnet	0.5174	0.6819	0.6901	0.6740	0.6341	0.7757	0.7408	0.8142	0.4197	0.5913	0.6063	0.5770	0.7007	0.8238	0.8502	0.7991
PAN	0.3904	0.5616	0.4614	0.7173	0.6231	0.7674	0.7271	0.8132	0.2765	0.4333	0.3532	0.5602	0.6535	0.7902	0.7444	0.8421
PSPNet	0.3553	0.5244	0.4084	**0.7322**	0.6197	0.7649	0.7058	**0.8356**	0.2927	0.4528	0.3757	0.5697	0.6582	0.7936	0.7761	0.8120
Unet	0.1562	0.2703	0.8370	0.1611	0.6397	0.7800	0.7496	0.8133	0.4857	0.6539	0.7022	**0.6118**	0.7061	0.8275	0.8231	0.8320
UnetPlusPlus	0.5257	0.6891	0.6653	0.7147	0.6444	0.7835	0.7714	0.7960	0.4257	0.5972	0.6179	0.5778	0.7168	0.8349	0.8324	0.8374
Context-CrackNet (ours)	**0.5668**	**0.7235**	**0.8989**	0.6054	**0.6733**	**0.8046**	**0.7967**	0.8129	**0.5375**	**0.6992**	**0.9386**	0.5571	**0.7401**	**0.8505**	**0.9175**	0.7926

	Eugen Miller				Forest				GAPs			
Model	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**
DeepLabV3	0.6312	0.7739	0.8696	0.6971	0.4826	0.6510	0.6151	0.6915	0.3704	0.5405	0.4694	0.6393
DeepLabV3Plus	0.6051	0.7540	0.7717	0.7371	0.4616	0.6316	0.5473	**0.7465**	0.3153	0.4785	0.4142	0.5679
FPN	0.4763	0.6452	0.5366	0.8091	0.4356	0.6069	0.5352	0.7007	0.3071	0.4699	0.3729	0.6353
Linknet	0.6024	0.7519	0.7473	0.7565	0.4857	0.6538	0.6664	0.6417	0.4007	0.5721	0.5430	0.6050
MAnet	0.5887	0.7411	0.6777	**0.8176**	0.5170	0.6816	0.6650	0.6990	0.3832	0.5540	0.5196	0.5937
PAN	0.5794	0.7337	0.7198	0.7482	0.4579	0.6281	0.5605	0.7143	0.2758	0.4323	0.3200	0.6668
PSPNet	0.6058	0.7545	0.7311	0.7794	0.3848	0.5557	0.4687	0.6825	0.2898	0.4491	0.3518	0.6237
Unet	0.6323	0.7747	0.7728	0.7767	0.5390	0.7005	0.6723	0.7311	0.3837	0.5545	0.4809	0.6548
UnetPlusPlus	**0.7060**	**0.8277**	0.8689	0.7902	0.5457	0.7061	0.6896	0.7234	0.3927	0.5637	0.4894	**0.6669**
Context-CrackNet (ours)	0.6627	0.7971	**0.9434**	0.6901	**0.5699**	**0.7261**	**0.8758**	0.6201	**0.4743**	**0.6433**	**0.7838**	0.5456

	Rissbilder				Sylvie				Volker			
Model	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**	**mIoU↑**	**Dice↑**	**Recall↑**	**Prec.↑**
DeepLabV3	0.5965	0.7471	0.7442	0.7502	0.6642	0.7982	0.7377	0.8696	0.7209	0.8378	0.8280	0.8479
DeepLabV3Plus	0.5653	0.7221	0.7057	0.7395	0.6505	0.7882	0.6964	**0.9079**	0.6872	0.8146	0.8046	0.8249
FPN	0.5819	0.7356	0.7153	**0.7572**	0.5966	0.7474	0.6570	0.8666	0.7028	0.8254	0.7994	**0.8534**
Linknet	0.6068	0.7552	0.7632	0.7476	0.6449	0.7842	0.7481	0.8238	0.7233	0.8394	0.8841	0.7990
MAnet	0.6365	0.7777	0.8104	0.7477	0.6263	0.7702	0.6899	0.8718	0.7422	0.8520	0.8543	0.8497
PAN	0.5416	0.7024	0.6403	0.7783	0.6074	0.7558	0.6869	0.8401	0.6796	0.8092	0.7720	0.8504
PSPNet	0.5042	0.6700	0.6146	0.7381	0.5481	0.7081	0.5841	0.8989	0.6870	0.8144	0.7905	0.8400
Unet	0.6456	0.7845	0.8105	0.7602	0.6577	0.7935	0.7622	0.8275	0.7464	0.8548	0.8574	0.8523
UnetPlusPlus	**0.6568**	**0.7926**	0.8233	0.7644	0.6718	0.8037	0.7804	0.8284	0.7641	0.8663	0.8983	0.8365
Context-CrackNet (ours)	0.6553	0.7916	**0.8469**	0.7432	**0.6846**	**0.8128**	**0.8429**	0.7847	**0.7668**	**0.8680**	**0.9002**	0.8381

overall performance variability, they do not by themselves clarify whether the observed differences are statistically significant.

Therefore, to rigorously evaluate whether the improvements achieved by Context-CrackNet were indeed meaningful relative to other baseline models, we employed the Wilcoxon signed-rank test, which determines if observed performance gains could simply stem from random variations.

Figure 7 presents the distribution of Validation IoU and Dice Scores across the evaluated segmentation frameworks, where the boxplots show that Context-CrackNet consistently attains higher median IoU and Dice values compared to other approaches. To substantiate these visual findings, we further computed Wilcoxon test p-values by contrasting the Validation IoU of Context-CrackNet with each baseline model (see Table 5). These results

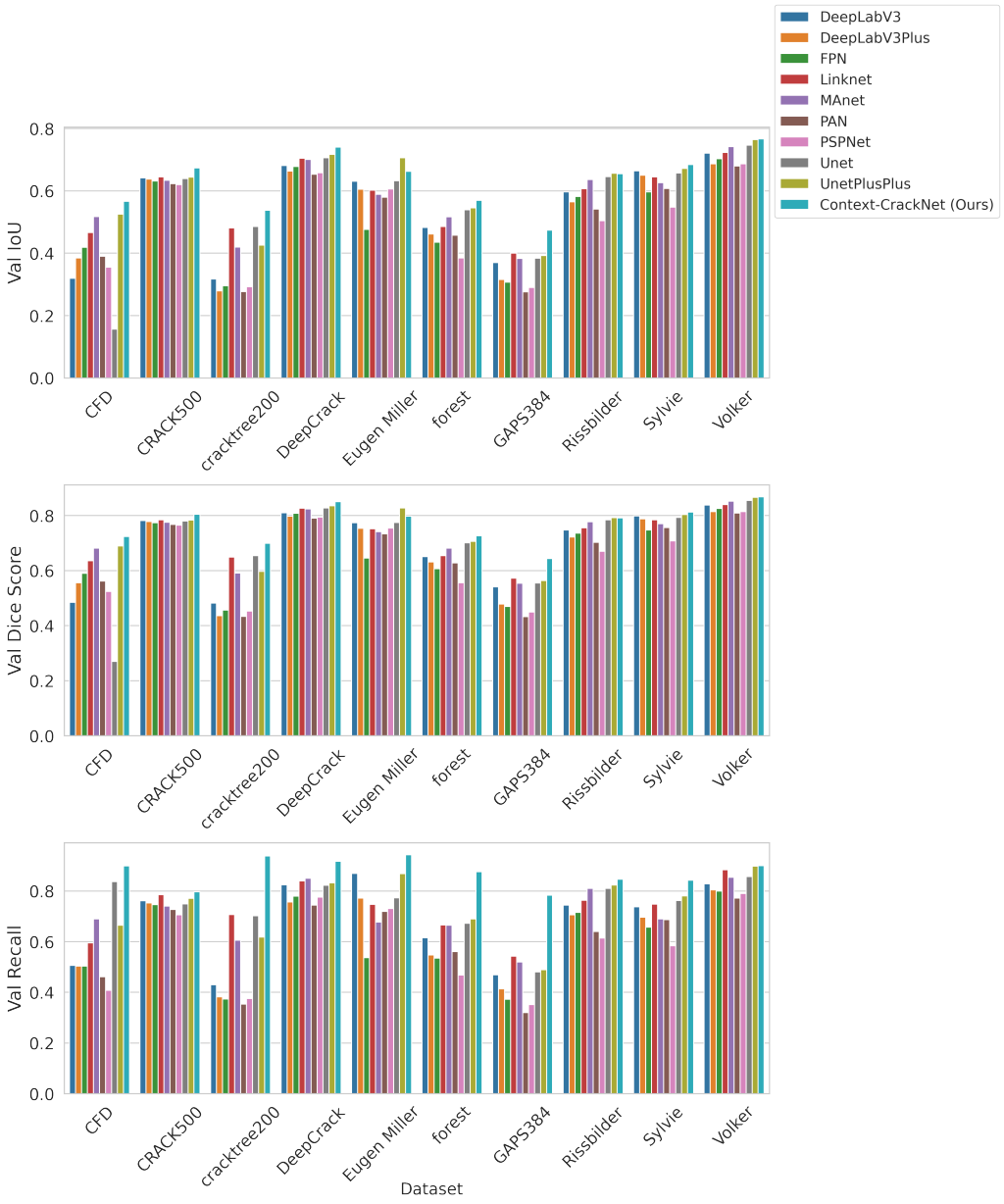

Figure 6: Performance comparison of the trained models across the different crack datasets. The top bar chart shows the Validation IoU, the middle bar chart shows the Validation Dice Score, and the bottom bar chart shows the Validation Recall. Higher bars indicate better performance.

Table 4: Mean ± standard deviation of the validation IoU and Dice score across the ten benchmark datasets.

Model	**IoU↑**	**Dice Score↑**
DeepLabV3	0.5426 ± 0.1556	0.6907 ± 0.1418
DeepLabV3Plus	0.5250 ± 0.1530	0.6757 ± 0.1398
FPN	0.5125 ± 0.1497	0.6665 ± 0.1358
Linknet	0.5761 ± 0.1148	0.7112 ± 0.1016
MAnet	0.5766 ± 0.1097	0.7149 ± 0.0987
PAN	0.5085 ± 0.1490	0.6578 ± 0.1375
PSPNet	0.4946 ± 0.1497	0.6490 ± 0.1379
Unet	0.5593 ± 0.1766	0.6979 ± 0.1687
UnetPlusPlus	0.6050 ± 0.1303	0.7386 ± 0.1161
ContextCrackNet (ours)	$\mathbf{0.6331 \pm 0.0945}$	$\mathbf{0.7718 \pm 0.0826}$

indicate that Context-CrackNets improvements are statistically significant against all baselines except UnetPlusPlus, for which the difference, while substantial, did not meet the significance threshold ($p = 0.0645$).

Table 5: Wilcoxon Signed-Rank Test Results between Context-CrackNet and Baseline Models

Baseline Model	**p-value**
DeepLabV3	0.0020
DeepLabV3Plus	0.0020
FPN	0.0020
Linknet	0.0020
MAnet	0.0020
PAN	0.0020
PSPNet	0.0020
Unet	0.0020
UnetPlusPlus	0.0645

5.4. Attention Map Visualization from RFEM Modules

The attention maps in Figure 8 illustrate the regions of the input images that the RFEM modules in *Context-CrackNet* focuses on during segmentation. These visualizations reveal that the model effectively identifies and em-

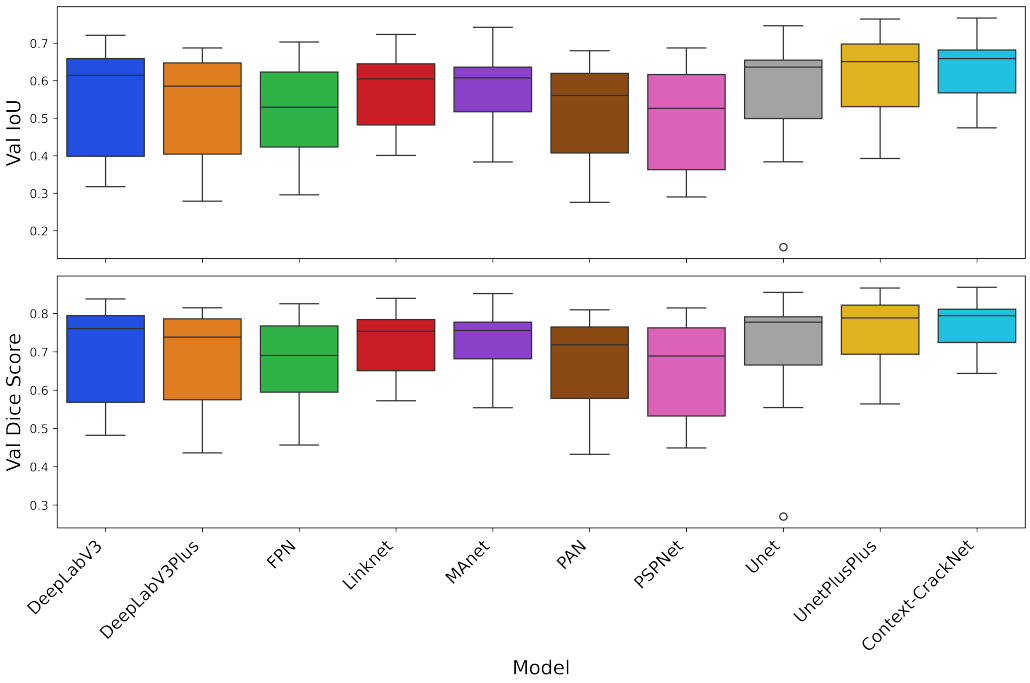

Figure 7: Distribution of Validation IoU and Val Dice Scores across Models

phasizes critical areas of distress across the different datasets. For example, in datasets like CRACK500 and DeepCrack, the attention maps distinctly capture intricate crack patterns, demonstrating the model's ability to localize fine-grained details. In complex cases like Rissbilder and Eugen Miller, the attention maps prioritize regions with subtle texture variations, ensuring accurate predictions even in challenging conditions. By highlighting relevant features, the attention maps provide interpretability to the models decisions and validate its capability to generalize across datasets with varying characteristics. This insight is crucial for understanding how the model adapts to different pavement distress types.

5.5. Ablation studies

To thoroughly analyze the contributions of the RFEM and CAGM modules, we conducted ablation experiments by selectively enabling or disabling these modules in the proposed architecture. These experiments allow us to quantify the individual and combined effects of RFEM and CAGM on segmentation performance. The results are summarized in Table 6.

Table 6: Ablation study results for RFEM and CAGM. Metrics include validation IoU, Dice Score, Precision, and Recall.

Configuration	mIoU↑	Dice Score↑	Precision↑	Recall↑
Baseline	0.4259	0.5929	0.5481	0.6691
RFEM Only	0.4355	0.6057	0.5439	0.6990
CAGM Only	0.4263	0.5954	0.5392	0.6827
RFEM + CAGM	**0.4743**	**0.6433**	**0.5456**	**0.7838**

Analysis and Discussion. The baseline model, without the RFEM and CAGM modules, achieved a validation IoU of 0.4259 and a Dice Score of 0.5929, demonstrating limited capability in capturing both local and global context. Adding RFEM alone improved the IoU to 0.4355 and the Dice Score to 0.6057, indicating that RFEM effectively enhances the model's ability to focus on critical local regions, especially fine-grained crack details. This is further reflected in the increase in recall from 0.6691 to 0.6990, as RFEM enables the model to identify more instances of distress.

When CAGM was included without RFEM, the IoU and Dice Score showed minimal improvements (0.4263 and 0.5954, respectively). While CAGM provides global context by capturing broader spatial dependencies, its contribution is less pronounced when local refinement (via RFEM) is absent. However, recall improved to 0.6827, suggesting that CAGM aids in generalizing to larger contextual regions, albeit at the expense of precision.

The model performed best when both RFEM and CAGM were included, achieving an IoU of 0.4743 and a Dice Score of 0.6433. The significant boost in recall to 0.7838 highlights the complementary roles of RFEM and CAGM. RFEM sharpens the models focus on localized crack patterns, while CAGM enriches the global context, leading to better overall segmentation. Interestingly, precision did not increase significantly with the inclusion of both modules, remaining relatively stable. This suggests that while the model identifies distress regions more effectively, some misclassifications persist, warranting further refinement.

5.6. Error Cases and Failure Scenarios

Figure 9 highlights several failure scenarios from *Context-CrackNet* across some of the crack datasets showing errors between ground truth masks and predicted masks. These failure scenarios often stem from inherent challenges

in the datasets, including low contrast between the cracks and their surroundings, and noise or texture inconsistencies in the images. For instance, in the CRACK500 dataset, the network struggled to differentiate between closely spaced cracks and surrounding noise, leading to incomplete or missed segments. Similarly, in the DeepCrack dataset, cracks exhibiting faint textures or irregular patterns were either under-segmented or omitted, reflecting the difficulty in capturing fine-grained structures under varying lighting conditions.

In datasets like Rissbilder and Forest, the models predictions were influenced by the complex background textures that mimic crack patterns. This suggests that the network may occasionally misinterpret irrelevant features as cracks, especially when contrast is minimal. The GAPs dataset, with its smooth and uniform background, exposed the network's sensitivity to subtle intensity variations, resulting in missed detections for faint cracks. Additionally, the Sylvie dataset presented a unique challenge where abrupt light intensity transitions and non-crack structures confused the model, leading to segmentation errors.

5.7. Model Complexity Analysis

Computational complexity significantly impacts real-time applications such as pavement distress monitoring. Table 7 compares models based on parameters, inference time, and GFLOPs. *Context-CrackNet*, with 82.05M parameters and 243.78 GFLOPs, achieves an inference time of 15.63 ms on high-performance GPUs. Although its parameter count is higher compared to lighter models such as FPN (26.12M, 6.35 ms) and DeepLabV3Plus (26.68M, 6.72 ms), it effectively captures complex spatial relationships critical for accurate crack segmentation.

Compared to UNetPlusPlus, which has higher FLOPs (352.68 GFLOPs) and slower inference (24.44 ms), *Context-CrackNet* offers a better balance between accuracy and efficiency (see Table 3). However, due to its relatively high computational requirements, deploying *Context-CrackNet* on resource-limited mobile or edge devices may require further optimizations like pruning or quantization.

6. Conclusion

This study introduces Context-CrackNet, a new deep learning model specifically developed to address the challenges of accurately detecting tiny

Table 7: Comparison of model complexity metrics across different architectures. Metrics include: (1) The number of parameters (in millions), (2) The average inference time (in ms) to process a single 448×448 image, and (3) GFLOPs (the number of floating-point operations, in billions, required for a single forward pass).

Model	**Parameters (M)**	**Inference Time (ms) (448 x 448 Image)**	**GFLOPs**
DeepLabV3	39.63	17.19	251.28
DeepLabV3Plus	26.68	6.72	56.42
FPN	26.12	6.35	48.08
LinkNet	31.18	6.85	66.06
MANet	147.44	12.85	114.48
PAN	24.26	7.01	53.46
PSPNet	24.26	3.09	18.14
UNet	32.52	7.67	65.60
UNetPlusPlus	48.99	24.44	352.68
Context-CrackNet	82.05	15.63	243.78

and subtle pavement cracks. To achieve this goal, we designed two key modules: the Region-Focused Enhancement Module (RFEM), which helps the model better identify fine details, and the Context-Aware Global Module (CAGM), which captures important global context from images. Together, these modules overcome major limitations found in previous crack segmentation approaches. Experiments conducted across ten diverse crack datasets showed that Context-CrackNet consistently outperformed existing state-of-the-art models, particularly when handling complex cracks of varying sizes under realistic conditions.

Beyond achieving high accuracy, Context-CrackNet also offers a good balance between precision and computational efficiency. This balance makes it highly suitable for practical, real-time pavement monitoring systems. By reliably detecting small cracks at an early stage, the model supports preventive maintenance strategies, potentially reducing long-term repair costs and extending the life of roads. This demonstrates the clear practical value of our research for infrastructure maintenance and management.

Furthermore, the innovations introduced in Context-CrackNet have implications beyond pavement distress detection. The RFEM and CAGM modules can be adapted to other areas requiring detailed and precise segmentation tasks under limited computational resources. Thus, our research lays the

groundwork for broader applications in infrastructure monitoring and beyond.

Looking forward, future studies could explore advanced data augmentation and domain adaptation methods to help the model generalize better to unseen real-world scenarios. Additionally, developing lighter versions of the attention mechanisms and applying model compression techniques could significantly reduce computational costs, enabling deployment on edge devices with limited resources. Expanding the approach to handle multiple classes of pavement distress or integrating it into predictive maintenance systems are also promising paths to enhance infrastructure management further.

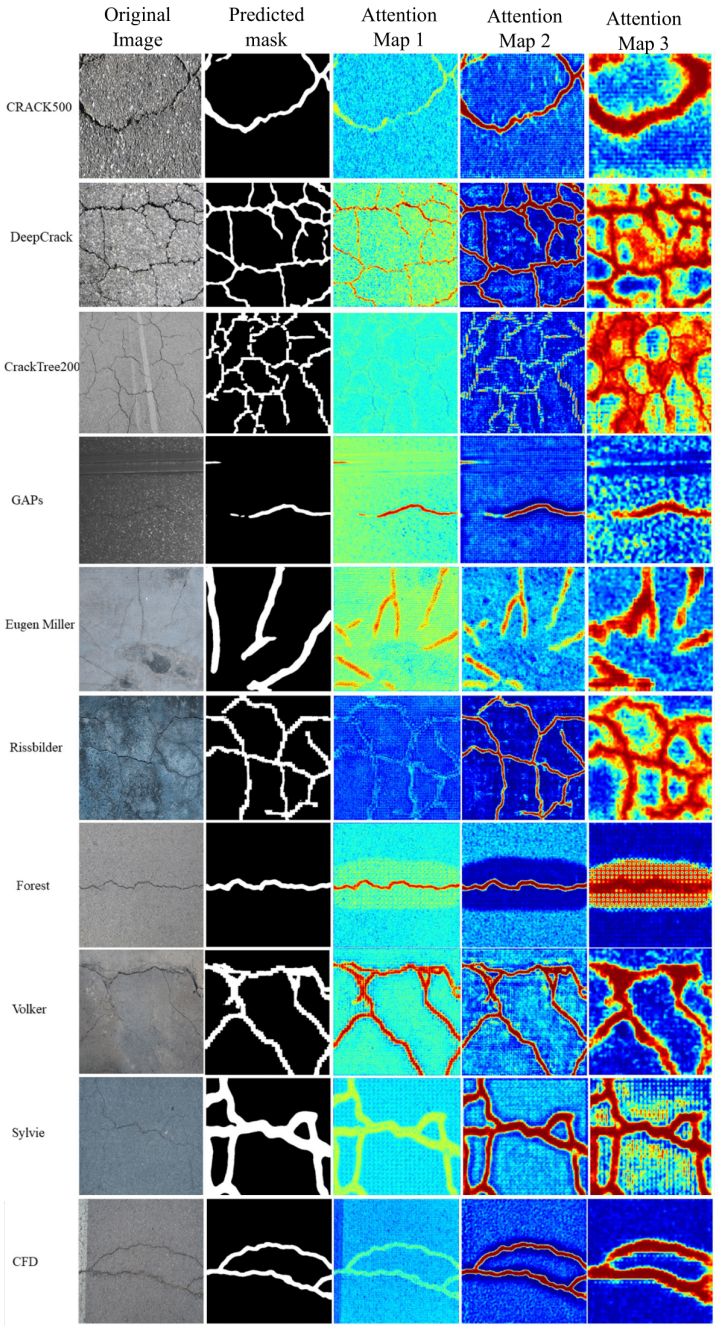

Figure 8: Visualization of predicted masks and attention maps from RFEM across various datasets. The attention maps highlight regions of interest contributing to the segmentation predictions from *Context-CrackNet*

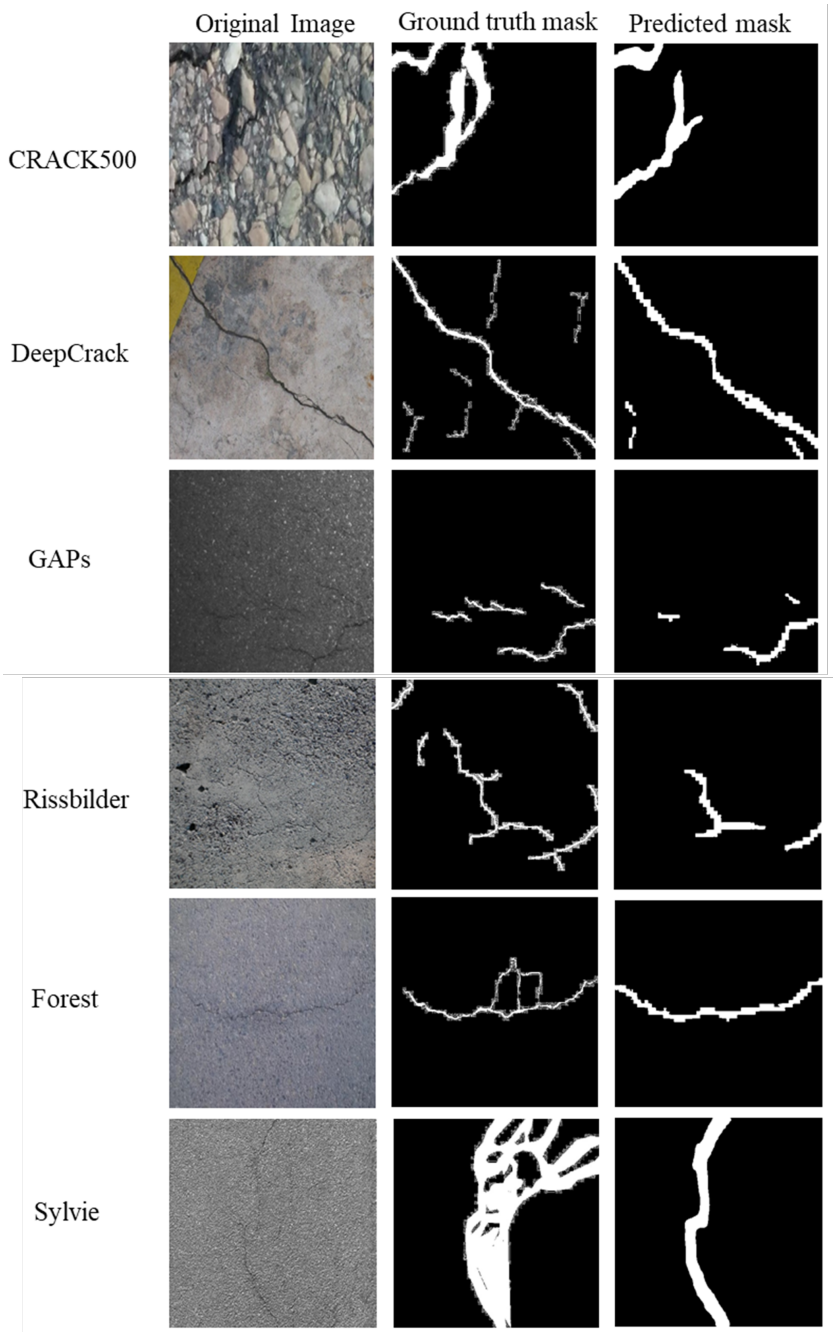

Figure 9: Failure cases from Context-CrackNet across various crack datasets, showing discrepancies between ground truth and predicted masks.

References

[1] Tian Wen, Shuo Ding, Hong Lang, Jian John Lu, Ye Yuan, Yichuan Peng, Jiang Chen, and Aidi Wang. Automated pavement distress segmentation on asphalt surfaces using a deep learning network. *International Journal of Pavement Engineering*, 24(2):2027414, 2023.

[2] James-Andrew R. Sarmiento. Pavement distress detection and segmentation using yolov4 and deeplabv3 on pavements in the philippines. *ArXiv*, abs/2103.06467, 2021.

[3] Feifei Li, Yongli Mou, Zeyu Zhang, Quan Liu, and Sabina Jeschke. A novel model for the pavement distress segmentation based on multi-level attention deeplabv3+. *Engineering Applications of Artificial Intelligence*, 137:109175, 2024.

[4] Zheng Tong, Tao Ma, Weiguang Zhang, and Ju Huyan. Evidential transformer for pavement distress segmentation. *Computer-Aided Civil and Infrastructure Engineering*, pages 2317–2338, 2023.

[5] Blessing Agyei Kyem, Eugene Kofi Okrah Denteh, Joshua Kofi Asamoah, and Armstrong Aboah. Pavecap: The first multimodal framework for comprehensive pavement condition assessment with dense captioning and pci estimation, 2024.

[6] Armstrong Aboah Neema Jakisa Owor, Yaw Adu-Gyamfi and Mark Amo-Boateng. Pavesam segment anything for pavement distress. *Road Materials and Pavement Design*, 0(0):1–25, 2024.

[7] Chong Zhang, Yang Chen, Luliang Tang, Xu Chu, and Chaokui Li. Ctcd-net: A cross-layer transmission network for tiny road crack detection. *Remote Sensing*, 15(8), 2023.

[8] Ghada Moussa and Khaled Hussain. A new technique for automatic detection and parameters estimation of pavement crack. 07 2011.

[9] Dennis A. Morian, Douglas Frith, Shelley Stoffels, and Shervin Jahangirnejad. Developing guidelines for cracking assessment for use in vendor selection process for pavement crack data collection/analysis systems and/or services. Technical Report FHWA-RC-20-0005, Federal

Highway Administration, Office of Technical Services, Baltimore, MD, Mar 2020. Prepared by Quality Engineering Solutions, Inc.

[10] Sheng Zhang, Zhenghao Bei, Tonghua Ling, Qianqian Chen, and Liang Zhang. Research on high-precision recognition model for multi-scene asphalt pavement distresses based on deep learning. *Sci. Rep.*, 14(1):25416, October 2024.

[11] Wen-Qing Huang, Liu Feng, and Yuan-Lie He. LTPLN: Automatic pavement distress detection. *PLoS One*, 19(10):e0309172, October 2024.

[12] Hoang Nhat-Duc, Quoc-Lam Nguyen, and Van-Duc Tran. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Automation in Construction*, 94:203–213, 2018.

[13] M. Salman, S. Mathavan, K. Kamal, and M. Rahman. Pavement crack detection using the gabor filter. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 2039–2044, 2013.

[14] Eduardo Zalama, Jaime GómezGarcíaBermejo, Roberto Medina, and José Llamas. Road crack detection using visual features extracted by gabor filters. *Computer-Aided Civil and Infrastructure Engineering*, 29(5):342358, September 2013.

[15] Xiaodong Chen, Dahang Ai, Jiachen Zhang, Huaiyu Cai, and Kerang Cui. Gabor filter fusion network for pavement crack detection. *Chinese Optics*, 2020.

[16] Allen A. Zhang, Q. Li, Kelvin C. P. Wang, and Shi Qiu. Matched filtering algorithm for pavement cracking detection. *Transportation Research Record*, 2367:30 – 42, 2013.

[17] S. Dorafshan, R. Thomas, and Marc Maguire. Benchmarking image processing algorithms for unmanned aerial system-assisted crack detection in concrete structures. *Infrastructures*, 2019.

[18] Dejin Zhang, Qingquan Li, Ying Chen, Min Cao, Li He, and Bailing Zhang. An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection. *Image Vis. Comput.*, 57:130–146, 2017.

[19] Zhongbo Li, Chao Yin, and Xixuan Zhang. Crack segmentation extraction and parameter calculation of asphalt pavement based on image processing. *Sensors (Basel, Switzerland)*, 23, 2023.

[20] Using image processing for automatic detection of pavement surface distress. *Al-Salam Journal for Engineering and Technology*, 2022.

[21] Cheng Peng, Mingqiang Yang, Qinghe Zheng, Jiong Zhang, Deqiang Wang, Ruyu Yan, Jiaxing Wang, and Bangjun Li. A triple-thresholds pavement crack detection method leveraging random structured forest. *Construction and Building Materials*, 2020.

[22] Jia Liang, Xingyu Gu, and Yizheng Chen. Fast and robust pavement crack distress segmentation utilizing steerable filtering and local order energy. *Construction and Building Materials*, 262:120084, 2020.

[23] Abdul Rahim Ahmad, Muhammad Khusairi Osman, Khairul Azman Ahmad, Muhammad Amiruddin Anuar, and Nor Aizam Muhamed Yusof. Image segmentation for pavement crack detection system. In *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pages 153–157, 2020.

[24] Amila Akagic, Emir Buza, Samir Omanovic, and Almir Karabegovic. Pavement crack detection using otsu thresholding for image segmentation. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1092–1097, 2018.

[25] Peggy Subirats, Jean Dumoulin, Vincent Legeay, and Dominique Barba. Automation of pavement surface crack detection using the continuous wavelet transform. In *2006 International Conference on Image Processing*, pages 3037–3040, 2006.

[26] S. Liang, Jianchun Xing, and Zhang Xun. An extraction and classification algorithm for concrete cracks based on machine vision. *IEEE Access*, 6:45051–45061, 2018.

[27] Nhat-Duc Hoang and Quoc-Lam Nguyen. A novel method for asphalt pavement crack classification based on image processing and machine learning. *Engineering with Computers*, 35:487 – 498, 2018.

[28] Abbas Ahmadi, Sadjad Khalesi, and A. Golroo. An integrated machine learning model for automatic road crack detection and classification in urban areas. *International Journal of Pavement Engineering*, 23:3536 – 3552, 2021.

[29] I. Barkiah and Yuslena Sari. Overcoming overfitting challenges with hog feature extraction and xgboost-based classification for concrete crack monitoring. *International Journal of Electronics and Telecommunications*, 2023.

[30] Adrien Müller, N. Karathanasopoulos, C. Roth, and D. Mohr. Machine learning classifiers for surface crack detection in fracture experiments. *International Journal of Mechanical Sciences*, 209:106698, 2021.

[31] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[32] Wenyu Zhang, Zhenjiang Zhang, Dapeng Qi, and Yun Liu. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors (Basel, Switzerland)*, 14:19307 – 19328, 2014.

[33] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[34] Jie Gao, Dongdong Yuan, Zheng Tong, Jiangang Yang, and Di Yu. Autonomous pavement distress detection using ground penetrating radar and region-based deep learning. *Measurement*, 164:108077, 2020.

[35] Suli Bai, Lei Yang, Yanhong Liu, and Hongnian Yu. Dmf-net: A dual-encoding multi-scale fusion network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 25:5981–5996, 2024.

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436444, May 2015.

[37] Blessing Agyei Kyem, Eugene Kofi Okrah Denteh, Joshua Kofi Asamoah, Kenneth Adomako Tutu, and Armstrong Aboah. Advancing pavement distress detection in developing countries: A novel deep learning approach with locally-collected datasets, 2024.

[38] Yann Lecun and Y. Bengio. Convolutional networks for images, speech, and time-series. 01 1995.

[39] Baoxian Li, Kelvin Wang, Allen Zhang, Enhui Yang, and Guolong Wang. Automatic classification of pavement crack using deep convolutional neural network. *International Journal of Pavement Engineering*, 21:1–7, 06 2018.

[40] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3708–3712, 2016.

[41] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 04 2019.

[42] Kai Li, Jie Yang, Siwei Ma, Bo Wang, Shanshe Wang, Yingjie Tian, and Zhiquan Qi. Rethinking lightweight convolutional neural networks for efficient and high-quality pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 25:237–250, 01 2024.

[43] Qiang Zhou, Zhong Qu, and Fang-rong Ju. A lightweight network for crack detection with split exchange convolution and multi-scale features fusion. *IEEE Transactions on Intelligent Vehicles*, 8(3):2296–2306, 2023.

[44] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *ArXiv*, abs/2006.04768, 2020.

[45] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, T. Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. pages 21297–21309, 2021.

[46] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Y. Teh. Set transformer. *ArXiv*, abs/1810.00825, 2018.

[47] Hairui Fang, Jin Deng, Yaoxu Bai, Bo Feng, Sheng Li, Siyu Shao, and Dongsheng Chen. Clformer: A lightweight transformer based on convolutional embedding and linear self-attention with strong robustness for

bearing fault diagnosis under limited sample conditions. *IEEE Transactions on Instrumentation and Measurement*, 71:1–8, 2022.

[48] Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shimin Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:5436–5447, 2021.

[49] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[50] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3708–3712. IEEE, 2016.

[51] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[52] Sangwoo Ham, Soohyeon Bae, Hwiyoung Kim, Impyeong Lee, Gyu-Phil Lee, and Donggyu Kim. Training a semantic segmentation model for cracks in the concrete lining of tunnel. 11 2021.

[53] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[54] Markus Eisenbach, Ronny Stricker, Daniel Seichter, Karl Amende, Klaus Debes, Maximilian Sesselmann, Dirk Ebersbach, Ulrike Stoeckert, and Horst-Michael Gross. How to get pavement distress detection ready for deep learning? a systematic approach. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2039–2047, 2017.

[55] Myeongsuk Pak and Sanghoon Kim. Crack detection using fully convolutional network in wall-climbing robot. In James J. Park, Simon James Fong, Yi Pan, and Yunsick Sung, editors, *Advances in Computer Science*

and Ubiquitous Computing, pages 267–272, Singapore, 2021. Springer Singapore.

[56] Rabih Amhaz, Sylvie Chambon, Jérôme Idier, and Vincent Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2718–2729, 2016.

[57] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[59] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, 39(6):1856–1867, 2020.

[60] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 06 2017.

[61] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing.

[62] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. pages 6392–6401, 06 2019.

[63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017.

[64] Abhishek Chaurasia and Eugenio Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, 2017.

[65] RUI LI, Shunyi Zheng, Ce Zhang, Chenxi Duan, Jianlin Su, Libo Wang, and Peter Atkinson. Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–13, 07 2021.

[66] Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. 05 2018.

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.