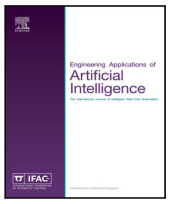
Engineering Applications of Artificial Intelligence 152 (2025) 110697

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Research paper

Efficient crack segmentation with multi-decoder networks and enhanced feature fusion

Ammar M. Okran [a,*], Hatem A. Rashwan [a], Adel Saleh [b], Domenec Puig [a]

[a] Department of Computer Engineering and Mathematics, Rovira i Virgili University, 43007 Tarragona, Spain
[b] Gaist Solutions Ltd., Skipton BD23 2TZ, United Kingdom

ARTICLE INFO

Keywords:
Artificial intelligence
Convolutional neural networks
Deep learning
Self supervised learning
Semantic segmentation
Pavement crack segmentation

ABSTRACT

In infrastructure management, intelligent crack detection is vital, particularly for maintaining crucial elements such as road networks in urban areas. Detecting pavement defects promptly and accurately is essential for timely repairs and hazard prevention. However, this task is challenging due to factors, such as complex backgrounds, micro defects, diverse defect shapes and sizes, and class imbalance issues. Innovative approaches and advanced technologies are needed to address these challenges and effectively manage infrastructure complexities. In this study, we propose a novel framework for crack segmentation, called CrackMaster. CrackMaster utilizes advanced neural network architectures, leveraging the next generation of convolutional networks (ConvNeXt) as an encoder and dual decoders customized for distinct tasks. The first decoder adopts a self-supervised learning paradigm to reconstruct images, thereby enhancing feature extraction capabilities. Meanwhile, the second decoder combines deep labelling network for semantic image segmentation (Deeplabv3+) with a light deep neural network (LinkNet) to facilitate precise segmentation. Notably, we introduce an Enhanced Feature Fusion (EFF) block to improve features quality, enhancing information flow and context preservation, thus boosting segmentation performance. Experimental results conducted on three diverse datasets, including our in-house Road Crack Dataset (RCD), DeepCrack537, and Yang Crack Dataset (YCD) datasets, demonstrate the effectiveness of our framework achieving outstanding Intersection over Union scores (IoU) of 86.0%, 87.8%, and 76.9%, respectively, showing superior accuracy and robustness in crack segmentation tasks. These findings underscore the potential applicability of our framework in real-world infrastructure management scenarios. The code is publicly available at: https://github.com/AmmarOkran/CrackMaster.

1. Introduction

Infrastructure, especially road networks, is crucial for the economic development of countries worldwide. Efficient transportation systems support trade, connect communities, improve access to essential services, and boost national prosperity. However, the integrity of this vital infrastructure faces constant challenges from environmental factors, heavy loads, and natural deterioration, resulting in road surface cracks (Meftah et al., 2024). Cracks in roadways are not just a common maintenance issue; they are early indicators of potential structural failures that can escalate into severe safety hazards, disrupt traffic, and incur significant repair costs. The traditional methods of road inspection, often manual visual assessments conducted by personnel, need to be revised. These methods are labour-intensive, subject to human error, and generally ineffective at early-stage crack detection, especially when the symptoms are not overtly visible (Yang et al., 2022).

With the advent of Artificial Intelligence (AI), road crack detection has relied on digital image processing and traditional handcrafted feature methods, such as threshold-based approaches (Kamaliardakani et al., 2016), percolation models (Qu et al., 2015), and minimal path searching (Amhaz et al., 2016) have been utilized extensively to identify crack patterns. Furthermore, feature extraction methods like Sobel filters (Ayenu-Prah and Attoh-Okine, 2008), Gabor filters (Salman et al., 2013), Histogram of Oriented Gradients (HOG) (Kapela et al., 2015), and Local Binary Patterns (LBP) (Yong and Chun-Xia, 2010). Also, many Machine Learning (ML) classifiers such as support vector machines (Oliveira and Correia, 2012) and random forests (Shi et al., 2016), have been instrumental in the initial stages of crack detection. The traditional AI-based methods, while helpful, often struggle in complex scenarios where background noise is high, or cracks lack clear continuity and contrast.

* Corresponding author.
E-mail address: ammar.okran@urv.cat (A.M. Okran).

https://doi.org/10.1016/j.engappai.2025.110697
Received 19 July 2024; Received in revised form 12 December 2024; Accepted 25 March 2025
Available online 11 April 2025

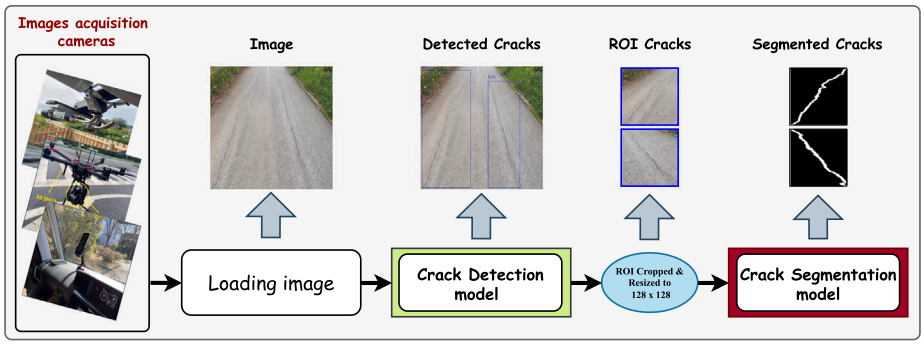


Fig. 1. An overview of our proposed road crack detection and segmentation framework. A road crack image is passed to the crack detection model; our segmentation model then processes the detected cracks (ROI). The outputs (segmented cracks) are represented by the masks of the detected cracks.

Recently, the progress of Deep Learning (DL) has transformed crack detection and segmentation. These DL methods have outperformed traditional image processing techniques, providing unparallelled accuracy in identifying and analysing structural defects. Convolutional neural networks (CNNs) have been widely used in crack segmentation and detection tasks due to their ability to effectively manage the details of diverse crack patterns and changing environmental conditions. The application of object detection methods such as Faster R-CNN (Faster Region-based Convolutional Neural Network) (Ren et al., 2015) YOLO (You Only Look Once) (Redmon et al., 2016), and SSD (Single Shot Detection) (Liu et al., 2016), has been particularly exploited in identifying and localizing cracks within large datasets of roadway images. For instance, studies such as Ding et al. (2022) and Okran et al. (2022b) have demonstrated the efficacy of these models in detecting surface cracks under varying lighting and weather conditions, showcasing significant improvements over manual inspection methods.

While crack detection is vital, comprehensive infrastructure maintenance strategies demand additional attention. Accurate cost estimation for road repair and maintenance requires accurate crack detection and segmentation. Detailed segmentation enables an exhaustive evaluation of crack dimensions, morphology, and extent, guiding decisions on material selection and labour distribution. Therefore, there is a rising emphasis on improving crack segmentation methods to deliver more detailed and practical data for informed decision-making in infrastructure maintenance (Munawar et al., 2022).

Consequently, numerous studies in the literature have been proposed to enhance the segmentation of cracks. One such innovative approach has been presented by DeepCrack (Liu et al., 2019) producing deep features at comparable scales that are then systematically fused in pairs to create a comprehensive representation of the detected crack regions. Another significant contribution comes from Zhou et al. (2022b), which replaces traditional convolution with enhanced convolution to capture long-range dependencies and local details. In turn, Okran et al. (2023) introduced an integrated framework for automatically segmenting road surface cracks, employing a Multi-Attention Network (Fan et al., 2020) (MANet) and a modified U-Net (Ronneberger et al., 2015), synergistically combined through neural network stacking.

Despite advancements, existing segmentation methods struggle in practical scenarios, particularly due to challenges in lighting, texture, noise sensitivity, and background complexity. Class imbalance and feature loss during down-sampling exacerbate the difficulty in detecting fine cracks. These limitations often result in missed cracks or misclassifications, reducing model reliability for real-world infrastructure monitoring (Kheradmandi and Mehranfar, 2022).

To address these challenges, we introduce CrackMaster, a two-stage framework that integrates detection and segmentation in an end-to-end structure. The model first detects regions of interest (ROI) to focus computational resources on crack-prone areas, filtering out non-essential background elements. Following this, precise crack segmentation is applied, leveraging our Enhanced Feature Fusion (EFF) block and Self-Supervised Learning (SSL) to capture intricate crack details and address class imbalance issues. Fig. 1 illustrates this innovative framework, designed for robust crack segmentation under varied conditions.

The key contributions of this work are summarized as follows:

- **An end-to-end crack detection and segmentation framework**: We introduce a novel, automated framework that reduces the need for manual oversight, improving efficiency and scalability in crack detection. The framework's two-stage approach first isolates regions of interest (ROIs) for targeted analysis, followed by precise pixel-level segmentation to enhance detection accuracy under challenging conditions.
- **Enhanced Feature Fusion (EFF)**: The EFF module integrates output from an Atrous Spatial Pyramid Pooling (ASPP) module with feature maps from each encoder layer, combining low-level and high-level information. This design mitigates feature loss during down-sampling and enhances the segmentation of subtle cracks by providing multi-scale context across different feature levels.
- **Self-Supervised Learning (SSL)**: To improve feature extraction, SSL reconstructs the original image during training, allowing the model to learn a comprehensive set of visual features of the input image. This approach addresses challenges related to fine detail preservation and diverse environmental conditions, enabling robust generalization across varied datasets.
- **Mitigation of Class Imbalance:** Our model design, incorporating SSL and EFF, effectively addresses class imbalance, enhancing its sensitivity to small, underrepresented crack regions, which improves segmentation accuracy for minor cracks.
- **Adaptability Across Diverse Datasets:** The proposed framework, CrackMaster, demonstrates robust performance across multiple crack segmentation datasets with varying characteristics, highlighting its potential for broad applicability in real-world scenarios.

The subsequent sections of the paper are structured as follows: Section 2 delves into existing research about crack detection, and segmentation. Section 3 introduces the CrackMaster model. The experimental dataset and setup are elaborated upon in Section 4, while Section 5 presents and analyzes the experimental findings across three datasets. Finally, Section 6 concludes the study and outlines potential avenues for future research.

2. Related works

The capability of detecting and segmenting cracks from infrastructure (roads, bridges, etc.) has undergone many developments. Each traditional method and advanced DL technique has fought the complexity brought by the pattern of cracks and the state of the surface in

Table 1
Summary of traditional crack detection related works.

Literature	Dataset	Models used	Results	Contributions	Limitations
Zhou et al. (2006)	Pavement images	Wavelet Transform	Classification error between 2% to 4%.	Multiscale crack detection	Prone to noise in high-frequency regions
Ayenu-Prah and Attoh-Okine (2008)	Pavement images	Sobel edge detector + Empirical Mode Decomposition	Accuracy of 80%–90%.	Enhanced edge detection with empirical mode decomposition	Sensitive to noise, limited for textured surfaces
Nguyen et al. (2011)	Pavement images	Free-form anisotropy method	Accuracy of 74%.	Anisotropic features for crack detection	Noise sensitivity and computationally intensive
Kapela et al. (2015)	173 images of asphalt pavements	Histogram of Oriented Gradients (HOG)	Accuracy of 70%–90%.	Texture-based features with HOG	Limited effectiveness in low-contrast images
Peng et al. (2015)	100 images of airport runway	Twice-threshold segmentation	Accuracy of 98%	Improved segmentation in controlled lighting	Limited adaptability in varying lighting
Kamaliardakani et al. (2016)	Pavement images (110 samples)	Thresholding-based method	Recall of 87%, Precision of 98%, and Accuracy and 93%	Effective for detecting sealed cracks	Sensitive to illumination changes
Shi et al. (2016)	CFD dataset	Random Structured Forests	Precision of 82.28%, Recall of 89.44%, and F1-score of 85.71%.	Enhanced detection with random forests	Noise sensitivity in complex surfaces
Amhaz et al. (2016)	269 images of asphalt pavements	Minimal path-based crack detection	Precision of 50% and Dice score of ≈%55.	Combines intensity and shape features	Limited in handling high curvature cracks

its day in different ways. The summary of traditional crack detection methods is shown in Table 1, and the overview of the crack detection and segmentation using DL methods are given in Tables 2 and 3 respectively.

2.1. Traditional methods for crack detection and segmentation

Traditional crack detection methods have been dominated by image processing methods and handcrafted features generating the contrast between the crack and background. Such techniques as edge detection, thresholding, and wavelet transforms are common. For example, Sobel edge detection in conjunction with empirical mode decomposition (Ayenu-Prah and Attoh-Okine, 2008) and Wavelet Transforms (Zhou et al., 2006) enable feature extraction that is essential for identifying cracks; these techniques have remained sensitive to noise. Other techniques, including Histogram of Oriented Gradients (HOG) and Laplacian of Gaussian (LoG), have demonstrated incremental improvements, particularly in applications like bridge deck and pavement inspections. However, their performance remains highly constrained under varying lighting conditions. Additionally, these methods demand significant effort in hyperparameter tuning, with parameters needing to be adjusted for each unique scenario and dataset, limiting their adaptability. Table 1 summarizes various traditional crack detection methods, highlighting their contributions and limitations.

2.2. DL-based methods for crack detection and segmentation

Recently, the success of deep learning in computer vision, pioneered by Krizhevsky et al. (2012), has encouraged researchers to apply these techniques to crack detection and segmentation. This blend of innovation in computer vision and the growing demand for reliable infrastructure monitoring has spurred numerous initiatives focused on utilizing DL for crack detection and segmentation. The following subsections review various methods in crack object detection and segmentation, showing a range of approaches and advancements within the deep learning field.

2.2.1. DL-based crack detection

DL-based crack detection has significantly advanced crack detection in infrastructure, particularly for roads and bridges. Leading methods include Faster R-CNN (Ren et al., 2015), YOLO (Redmon et al., 2016), and SSD (Liu et al., 2016), which consistently achieve high accuracy in crack detection and classification tasks. Additionally, Maeda et al. (2018) employed two widely recognized CNN architectures, MobileNet (Howard et al., 2017) and Inception (Ioffe and Szegedy, 2015), to detect defects on road surfaces from smartphone images, balancing accuracy with real-time performance. Wang et al. (2018a) further improved precision using Faster R-CNN, leveraging greater computational resources. Ensemble methods, such as those by Okran et al. (2022b), also showed robust results, with their YOLOv7-based ensemble achieving approximately 72.6% F1-score. However, these detection models often face limitations in accurately identifying small or fine cracks, particularly under variable lighting and environmental conditions. Their high computational demands can hinder real-time applicability, especially in resource-constrained scenarios. Moreover, these models are typically unable to precisely delineate crack boundaries, which is essential for detailed assessment and accurate segmentation in infrastructure monitoring.

Table 2 provides a detailed comparison of these crack detection methods, outlining each technique's unique contributions and inherent limitations, offering a comprehensive overview of DL-based approaches in this domain.

2.2.2. DL-based crack segmentation

Crack segmentation is a pixel-level computer vision approach that accurately differentiates crack regions from non-crack areas using precisely labelled masks. This method identifies the geometry and positioning of cracks within images, making it essential for precision-critical applications such as road and infrastructure assessments. Deep learning, particularly with convolutional neural networks (CNNs), has significantly advanced semantic segmentation for crack detection, effectively addressing challenges in diverse road and surface conditions (Okran et al., 2022a). However, these methods still face several limitations, such as their sensitivity to environmental variations, reliance on large annotated datasets, and computational demands, which hinder their broader deployment in real-world scenarios.

Recently, various models have been proposed for crack segmentation. Early efforts, such as the Fully Convolutional Network (FCN) by Yang et al. (2018), were among the first to implement pixel-level segmentation for cracks, achieving promising results but remaining sensitive to environmental variables. Fan et al. (2018) later enhanced segmentation accuracy using structured prediction techniques, demonstrating strong real-world performance but at the cost of requiring extensively annotated datasets. More recently, models have integrated attention mechanisms and multiscale feature extraction to improve robustness and segmentation detail. For instance, Ma et al. (2024a) introduced an attention-based fusion network that achieves fine-grained segmentation, albeit with higher computational demands. Despite these

Table 2
Overview of DL-based crack detection methods.

Literature	Dataset	Models	Results	Contributions	Limitations
Maeda et al. (2018)	RDD-2018	MobileNet, Inception	Average accuracy of >80% for eight types of road damage.	Real-time detection on mobile devices	Limited for fine cracks
Wang et al. (2018b)	RDD-2018	Faster R-CNN	Mean F1-score of 0.6255	Fine-tuned for surface variation	High computational cost
Maeda et al. (2021)	RDD-2018 & RDD-2019	GAN, Poisson blending	Improved F1-score by 5% on small datasets and 2% on large datasets.	Synthetic data for rare cases	Computationally heavy
Jeong (2020)	RDD-2020	YOLOv5x	F1-score of 0.58	Cross-country training for robustness	Limited to specific regions
Ding et al. (2022)	RDD-2022	YOLO- series models and Faster RCNN	F1-score of 0.7699 for all countries.	Ensemble for small damage	Resource-intensive
Okran et al. (2022b)	RDD-2022	Ensemble YOLO with ASPP	F1-score of 0.7260 for overall test dataset.	Enhanced context capture	Computationally intensive
Li et al. (2024a)	1150 images with clear crack targets, 1024 × 1024 resolution.	CrackTinyNet (CrTNet) using BiFormer, NWD loss function, SPD-Conv.	MAP@0.5 of 0.601, F1-score of 0.61.	High-precision detection of tiny cracks, validated on real vehicles.	Limited to specific dataset, potential overfitting on tiny targets.
He et al. (2024)	RDD-2022	YOLOv7-BiFPN-G with Ghost convolution and knowledge distillation.	Precision of 0.574, Recall of 0.583, and MAP@0.5 of 0.563.	Efficient feature fusion, reduced model size, mobile-friendly deployment.	Limited to specific dataset, potential overfitting on small targets, high dependency on high-quality images.
Li et al. (2024b)	7644 images, augmented with CycleGAN	CycleGAN, YOLOv5 with CBAM and AS-SE modules.	Precision of 0.872, Recall of 0.854, and MAP@0.5 of 0.882.	Enhanced data augmentation, improved feature extraction, real-time detection.	Limited to specific dataset, potential overfitting, high computational complexity.
Lu et al. (2025)	4868 images of dam cracks.	YOLO-DEW (improved YOLOv8s) and Unet.	YOLO-DEW: mAP of 83.5%, Unet: mIoU of 80.73%, mPA of 86.49%.	Enhanced feature extraction, accurate crack localization.	Requires high-res imaging, high computational complexity.

Table 3
Overview of DL-based crack segmentation methods.

Literature	Dataset	Models	Results	Contributions	Limitations
Yang et al. (2018)	YCD (Pavement cracks)	Fully Convolutional Network (FCN)	Precision of 81.7%, Recall of 79.0%, and F1-score of 79.95%	One of the first FCN approaches for crack segmentation	Sensitive to noise and illumination changes
Fan et al. (2018)	CFD and AigleRN	Structured Prediction with CNN	CFD: F1-score of 0.924 - AigleRN: F1-score of 0.895.	Adapts CNN for structured prediction in cracks	Requires high-quality labelled data
Dung et al. (2019)	40 000 images with 227 × 227 pixels	Deep Fully Convolutional Network	AP of 89.3%	Autonomous model for structural health monitoring	Limited adaptability to varying crack textures
Liu et al. (2019)	537 images (DeepCrack dataset)	DeepCrack CNN	mIoU of 85.9% and F1-score of 86.5%	Hierarchical feature learning across scales	Computationally intensive
Ni et al. (2019)	800 images (Concrete cracks)	Feature Fusion CNN	F1-score of 81.9%	Fuses multi-scale features for enhanced clarity	Limited by high memory usage
Ren et al. (2020)	409 images (Tunnel cracks)	Deep FCN (CrackSegNet)	F1-score of 74.55%	Domain-specific enhancements for tunnel cracks	High computational needs for complex environments
Okran et al. (2023)	6246 images (Road surfaces)	Cascade Network	F1-score of 0.895	Stacking model improves efficiency	Computationally demanding
Tao et al. (2023)	Crack500 and DeepCrack	Convolutional-Transformer Network	Crack500: F1-score of 73.3% - DeepCrack: F1-score of 88.3%.	Combines CNN with Transformer for clarity	Potentially slower for large datasets
Ma et al. (2024a)	DeepCrack, Crack500, and CFD	Attention-based Progressive Fusion	mIoU of 0.909, 0.827, and 0.860, respectively.	Uses attention mechanisms for detailed focus	High resource requirements
Bai et al. (2024)	DeepCrack, YCD, and CF D	Dual-Encoder Fusion Network	mIoU of 92.4%, 77.8%, and 79.8%, respectively.	Fusion model enhances crack localization	Limited with faint or complex patterns
Cano-Ortiz et al. (2024)	DeepCrack, CrackSC, CFD, and Crack500	Semantic Diffusion Synthesis Model	F1-score of 0.718, 0.330, 0.323, and 0.707, respectively.	Integrates semantic synthesis for detail	Computationally intensive
Tao (2024)	Crack500, DeepCrack, CFD, Cracktree206, GAPs384, and AEL	Weakly-Supervised Network	OIS (0.739, 0.758, 0.677, 0.728, 0.641, and 0.406)	Dual separation enhances generalization	Limited by weak supervision constraints
Okran et al. (2024)	Crack500	Multi-Dimensional Attention Network	Precision of 73.2%, Recall of 76.3%, and F1-score of 74.7%,	Uses multi-dimensional attention for accuracy	High computational needs

advancements, many state-of-the-art models still struggle with challenges like imprecise boundary delineation, difficulty in handling class imbalance, and limited generalization across different domains, underscoring the need for further innovation in crack segmentation methods.

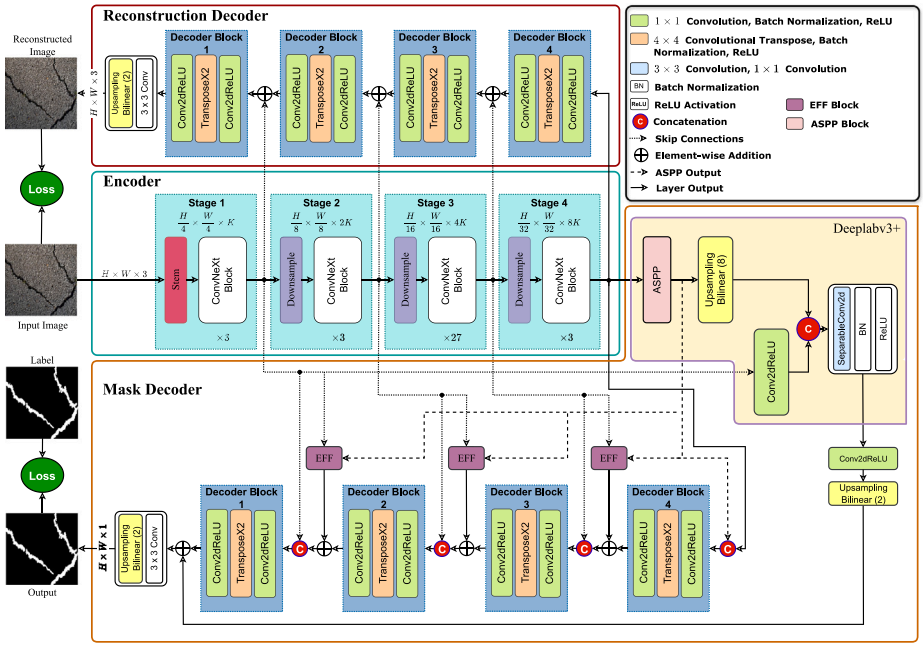


Fig. 2. The architecture of the proposed segmentation model.

Table 3 summarizes several deep learning-based crack segmentation algorithms, detailing the contributions and limitations of each. This table provides a comprehensive review of methodologies and technologies advancing crack segmentation.

Thus, traditional crack detection methods face significant limitations, including sensitivity to environmental variations, reliance on manual hyperparameter tuning, and poor adaptability across diverse surfaces. While DL-based detection models have improved accuracy, they often fail to delineate precise crack boundaries and require substantial computational resources, making real-time deployment challenging. Similarly, deep learning-based segmentation models advance boundary detection but struggle with issues like loss of fine details due to down-sampling, dependency on large annotated datasets, and challenges in handling class imbalance when cracks occupy minimal image areas. To overcome these drawbacks, this paper proposes CrackMaster, a novel dual-decoder framework that combines Enhanced Feature Fusion (EFF) and Self-Supervised Learning (SSL). CrackMaster addresses class imbalance, enhances boundary delineation, and reduces computational overhead by focusing on Regions of Interest (ROIs) for segmentation. Additionally, its robust design ensures adaptability across diverse datasets and environmental conditions, offering a comprehensive solution to the limitations of existing methods.

3. Methodology

In this section, we summarize the complete framework combining the crack detection process of the regions of interest (ROIs) as established in our previous work (Okran et al., 2022b) and the crack segmentation outlining the general architecture of the proposed CrackMaster shown in Fig. 2 and providing a detailed description of each component within the system.

3.1. Crack detection

The initial phase of our system leverages a robust detection model proposed in our previous work (Okran et al., 2022b). This model is specifically employed to identify regions of interest (ROIs) within the larger images. These ROIs are areas where potential cracks are most likely found, allowing the system to focus its computational resources on regions requiring detailed analysis. Using a pre-existing, validated model for this task ensures high reliability and accuracy in the preliminary detection of potential cracks. Once the ROIs are detected, they are meticulously cropped from the original images. Each cropped region is then resized to a standard dimension, ensuring uniformity in the input data for the subsequent segmentation phase. This preprocessing step is crucial as it normalizes the input data, facilitating more consistent performance from the segmentation network.

3.2. Crack segmentation: CrackMaster model

Fig. 2 depicts the CrackMaster model, a specialized architecture designed to detect cracks in images of cracked surfaces. The framework is divided into three main components. Firstly, the encoder, situated at the core of the network, is responsible for extracting robust features from crack images, a crucial step in the initial identification and analysis of damaged areas.

Secondly, The first decoder, positioned above the encoder, employs an SSL technique to reconstruct the original image. This process is vital as it enhances the network's capability to capture and interpret fundamental features of the image. The reconstructed image plays a pivotal role in enabling the network to effectively understand the visual features of the input image that can help capture essential features related to crack regions.

The final component of the network, the second decoder (bottom part in Fig. 2), is for producing accurate crack masks. This decoder utilizes the high-quality features honed by the encoder and further improved by the image reconstruction process of the first decoder. Advanced techniques are integrated into the segmentation decoder to guarantee precise delineation of each crack, which is crucial for practical damage assessment and repair planning.

In our framework, the encoder is fed with the input image and outputs multi-scale features, which are fed into two decoders trained in parallel, optimizing both components simultaneously. This setup allows for efficiently utilizing extracted features, enhancing the learning process. The training of the end-to-end framework involves calculating the total loss, a composite of the losses from both decoders. This total loss is then used to perform backpropagation through the network. By optimizing both decoders together, the network effectively leverages shared encoder features to enhance the reconstruction quality and the accuracy of the segmentation. This method ensures that each decoder

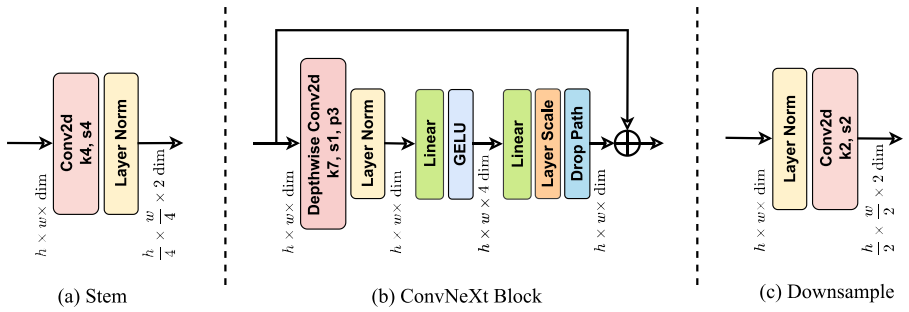


Fig. 3. (a) The internal structure of Stem; (b) The internal structure of ConvNeXt Block; (c) The internal structure of Downsample.

Table 4
The detailed structure of the encoder network.

Stages	Layers	Input	Layer type	Filters	Kernel size	Stride and padding	Groups	Output shape	Bias
Stage1	Stem	128 × 128 × 3	Conv2d + LayerNorm2d	128	4	(4, 4),(0, 0)	1	32 × 32 × 128	–
	(ConvNeXt) × 3	32 × 32 × 128	Conv2d	128	7	(1, 1),(3, 3)	128	32 × 32 × 128	–
		128 × 32 × 32	LayerNorm2d + Linear	–	–	–	–	512 × 32 × 32	True
		512 × 32 × 32	GELU + Linear	–	–	–	–	128 × 32 × 32	True
		32 × 32 × 128	Layer Scale	–	–	–	–	32 × 32 × 128	–
		32 × 32 × 128	DropPath	–	–	–	–	32 × 32 × 128	–
Stage2	Downsample	32 × 32 × 128	LayerNorm2d + Conv2d	256	2	(2, 2),(0, 0)	1	16 × 16 × 256	–
	(ConvNeXt) × 3	16 × 16 × 256	Conv2d	256	7	(1, 1),(3, 3)	256	16 × 16 × 256	–
		256 × 16 × 16	LayerNorm2d + Linear	–	–	–	–	1024 × 16 × 16	True
		1024 × 16 × 16	GELU + Linear	–	–	–	–	256 × 16 × 16	True
		16 × 16 × 256	Layer Scale	–	–	–	–	16 × 16 × 256	–
		16 × 16 × 256	DropPath	–	–	–	–	16 × 16 × 256	–
Stage3	Downsample	16 × 16 × 256	LayerNorm2d + Conv2d	512	2	(2, 2),(0, 0)	1	8 × 8 × 512	–
	(ConvNeXt) × 27	8 × 8 × 512	Conv2d	512	7	(1, 1),(3, 3)	512	8 × 8 × 512	–
		512 × 8 × 8	LayerNorm2d + Linear	–	–	–	–	2048 × 8 × 8	True
		2048 × 8 × 8	GELU + Linear	–	–	–	–	512 × 8 × 8	True
		8 × 8 × 512	Layer Scale	–	–	–	–	8 × 8 × 512	–
		8 × 8 × 512	DropPath	–	–	–	–	8 × 8 × 512	–
Stage4	Downsample	8 × 8 × 512	LayerNorm2d + Conv2d	1024	2	(2, 2),(0, 0)	1	4 × 4 × 1024	–
	(ConvNeXt) × 3	4 × 4 × 1024	Conv2d	1024	7	(1, 1),(3, 3)	1024	4 × 4 × 1024	–
		1024 × 4 × 4	LayerNorm2d + Linear	–	–	–	–	4096 × 4 × 4	True
		4096 × 4 × 4	GELU + Linear	–	–	–	–	1024 × 4 × 4	True
		4 × 4 × 1024	Layer Scale	–	–	–	–	4 × 4 × 1024	–
		4 × 4 × 1024	DropPath	–	–	–	–	4 × 4 × 1024	–

enhances its respective task based on a comprehensive understanding of the image features. This leads to a synergistic improvement in the network's performance on both fronts. This dual-decoder setup not only facilitates the training process but also ensures that both essential tasks – image reconstruction and crack segmentation – are addressed effectively, maximizing the potential of the shared encoder architecture. The experimental findings confirm our hypothesis. The proposed CrackMaster framework, which we outline in the subsequent subsections, is designed with an encoder and a dual-decoder configuration.

3.2.1. Encoder: The model backbone

The heart of our CrackMaster architecture lies a unique ConvNeXt-Base (Liu et al., 2022) model, serving as an encoder. This model fine-tunes from a pre-trained state, a technique that harnesses its advanced feature extraction capabilities honed through extensive preliminary training on diverse images. This fine-tuning approach allows us to adapt these pre-acquired capabilities to the nuanced task of crack segmentation, thereby boosting the efficiency and accuracy of the network. The ConvNeXt-Base encoder is systematically organized into four progressive blocks. The encoder receives an input image with dimensions of 128 × 128 × 3 and processes it to produce a feature map with dimensions of 4 × 4 × 1024. The specific configuration of the encoder is detailed in Table 4. Each block of the encoder network is designed to incrementally refine the features extracted from the input images, which is crucial for the detailed analysis required in the subsequent phases of reconstruction and segmentation. The architecture of each stage comprises two key components:

Stem and downsampling: As shown in Fig. 3(a), the Stem block comprises a convolutional layer followed by a normalization layer. Consider an input image I with dimensions $H \times W \times 3$, where H and W denote the height and width of the image, respectively. The first step in processing this image involves downsampling it to one-quarter of its original dimensions. This downsampling is achieved through a convolutional layer with kernel size 4 × 4 and a stride of 4. Following this convolutional operation, the resulting output features undergo a normalization process via the normalization layer technique, which standardizes the features across the network to improve the stability and speed of the training process. The Stem block is utilized only in the first stage, as depicted in Fig. 2.

$$\hat{I}_{stem} = \eta(\varphi_{4\times4}(I)), \tag{1}$$

where $\hat{I}_{Stem}$ represents the feature map produced by Stem block, I is the input image, $\varphi_{4\times4}$ represents 4 × 4 convolution with stride $s = 4$, and η stands for normalization layer.

For the following stages, we use downsampling, which decreases the resolution of the feature map while simultaneously increasing the channel dimension. This adjustment facilitates the production of hierarchical, multi-scale features. As shown in Fig. 3(c), the downsampling module consists of a normalization layer followed by a convolution

layer. Specifically, the convolution layer employs size 2 × 2 kernels with a stride of 2, as shown in Eq. (2).

$$\hat{I}_{ds} = \varphi_{2\times2}(\eta(\hat{I}_n)), \tag{2}$$

where $\hat{I}_{ds}$ represents the feature map produced by the downsampling block, $\hat{I}_n$ represents the feature map of previous stage, and η represents the normalization layer, and $\varphi_{2\times2}$ denotes the convolution operation with a kernel size of 2 × 2 and stride $s = 2$.

This configuration effectively reduces the spatial dimensions of the feature map by half in both height and width, thus reducing resolution. At the same time, the convolution process can be designed to increase the number of channels, thereby enriching the feature representation with information captured at various scales. This component is employed in the rest of the stages except the first, as shown in 2.

The ConvNeXt block: Fig. 3(b) presents the configuration of the ConvNeXt Block, which is intricately structured to optimize feature processing. The block begins with a depthwise convolution layer, which independently performs spatial filtering by applying a single convolutional filter to each input channel. This method is effective for manipulating spatial features while keeping computational costs low. The resulting output features are then standardized through the normalization layer technique, enhancing stability and accelerating the training process across the network. The architecture incorporates two consecutive pointwise linear layers after the depthwise convolution. These layers play a crucial role in channel-wise feature transformation, adjusting the channel dimensions while the spatial dimensions of the feature map remain unchanged. A significant improvement of ConvNext is to use the Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel, 2016) non-linearity, strategically placed between these two layers. The GELU non-linearity introduces a probabilistic approach to activations, a key factor in capturing more complex patterns in the data than traditional activation functions like ReLU, thereby enhancing the architecture's performance.

Following the second linear layer, a Layer Scale (Touvron et al., 2021) of initial value 1e−6 is applied that adjusts the amplitude of features, potentially stabilizing the network's training phase by controlling the activation scale. Then, ConvNext strategically incorporates the drop path layer to enhance the model's generalization capabilities. This layer randomly omits certain paths in the network during training, similar to dropout, effectively reducing the risk of overfitting and improving the model's ability to generalize from the training data. Moreover, the block includes a residual connection, a critical element that adds the input of the block to its output. This feature is instrumental in preventing the vanishing gradient problem and enabling the training of deeper networks by allowing direct backward flow of gradients. The operations performed within the ConvNeXt block can be formulated as follows:

$$\hat{I}_{ConvNeXt} = \Omega(Y(\alpha(Y(\eta(\Phi_{7\times7}(\hat{I}_b))))) * \gamma) + \hat{I}_b, \tag{3}$$

where $\hat{I}_{ConvNeXt}$ denotes the output of a ConvNeXt block, $\hat{I}_b$ represents the feature map of previous block either $\hat{I}_{Stem}$ or $\hat{I}_{ds}$, and $\Phi_{7\times7}$ represents the depthwise convolution operation with a kernel size of 7 × 7, η represents the normalization layer, Y stands for the full connection layer, α stands for the GELU activation function, γ represents the scale layer, and Ω represents the drop path layer.

The ConvNeXt block is repeatedly used within each model stage, with three repetitions per stage to refine features at multiple levels. However, in the third stage, the block is repeated 27 times, suggesting a focus on deeply processing and enhancing features at a critical network level to ensure detailed and robust feature extraction for accurate downstream performance.

3.2.2. Dual-branch decoder

The proposed CrackMaster model features a dual-branch decoder system, each branch serving a distinct and critical function within the network. This dual-decoder setup is designed to maximize the model's capabilities in understanding the visual features of the input image for reconstructing detailed images and precisely segmenting cracks. Each decoder operates in parallel but targets different aspects of the image processing task, ensuring comprehensive analysis and output generation.

Reconstruction decoder: The first decoder branch of the CrackMaster model is dedicated to image reconstruction, playing a vital role in refining the network's feature extraction capabilities. By reconstructing the original image, this decoder helps the encoder better understand and reproduce the main visual features of the input image, enhancing the quality of features in the crack regions. This reconstruction process is based on self-supervised learning, where the original image serves as both the input and the reference for the output. The detailed architecture of this decoder is provided in Table 5.

The reconstruction decoder in our CrackMaster employs the LinkNet (Chaurasia and Culurciello, 2017) decoder architecture, which consists of four deconvolution blocks, each tailored to refine and upscale the feature maps extracted from the encoder to reconstruct the input image accurately. Each block begins with a convolutional layer using a 1 × 1 kernel size for initial feature refinement, followed by batch normalization and ReLU activation to prepare the features for effective upscaling. The sequence continues with a transposed convolutional layer employing a 4 × 4 kernel with a stride of 2, designed to expand the spatial dimensions of the feature maps. This is complemented by batch normalization and ReLU activation for enhanced continuity and quality. Each block concludes with another 1 × 1 convolutional layer that integrates the processed features, then fine-tuning them to restore detailed features, followed by subsequent normalization and activation. To preserve the spatial information and anatomical structures of the input images extracted by the encoder, skip connections are employed between corresponding layers of the encoder and each reconstruction decoder block. We can formulate these operations of each decoder block as follows:

$$A_n = \varphi_{1\times1}(\psi_{4\times4}(\varphi_{1\times1}(\hat{I}_n))) \oplus \hat{I}_{n-1}, \tag{4}$$

where A_n denotes the output of the decoder block, $\hat{I}_n$ represents the output of the last stage of the encoder or the previous decoder block, $\varphi_{1\times1}$ represents 1 × 1 convolution, $\psi_{4\times4}$ represents the transposed convolution operation with a kernel size of 4 × 4 and stride s = 2, $\hat{I}_{n-1}$ denotes to the corresponding feature maps of the encoder, and $\oplus$ represents the elementwise addition.

The final block in the decoder involves a convolutional layer equipped with a 1 × 1 kernel at the top of the decoder, which finalizes the image reconstruction. This is followed by an upsampling process using bilinear interpolation to refine further and scale the reconstructed image back to its original dimensions, as shown in Eq (5):

$$I_{rec} = \uparrow_2 (\varphi_{1\times1}(A_1)), \tag{5}$$

where I_{rec} denotes the reconstructed image, A_1 represents the output of the last decoder block, and $\uparrow_2$ represents the upsampling operation by factor 2.

Crack segmentation decoder: In the CrackMaster model, the second decoder branch employs a powerful combination of advanced decoding architectures: DeepLabv3+ (Chen et al., 2018) and LinkNet (Chaurasia and Culurciello, 2017). This integration strategically utilizes the individual strengths of each architecture to achieve superior segmentation results, where the final output is a sophisticated fusion of both decoders' outputs. The **DeepLabv3+ network** is exploited in the segmentation decoder that effectively uses the Atrous Spatial Pyramid Pooling (ASPP) module (Chen et al., 2017). The ASPP module utilizes atrous convolution at different rates (12, 24, and 36) to capture contextual information across various scales, critical for handling cracks'

Table 5
The detailed structure of the LinkNet-based SSL reconstruction decoder network.

Layers	Input	Layer type	Filters	Kernel size	Stride and padding	Output shape
(Block4)	4 × 4 × 1024	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1),(0, 0)	4 × 4 × 256
	4 × 4 × 256	ConvTranspose2d + BatchNorm2d + ReLU	256	4	(2, 2),(1, 1)	8 × 8 × 256
	8 × 8 × 256	Conv2d + BatchNorm2d + ReLU	512	1	(1, 1),(0, 0)	8 × 8 × 512
(Block3)	8 × 8 × 512	Conv2d + BatchNorm2d + ReLU	128	1	(1, 1),(0, 0)	8 × 8 × 128
	8 × 8 × 128	ConvTranspose2d + BatchNorm2d + ReLU	128	4	(2, 2),(1, 1)	16 × 16 × 128
	16 × 16 × 128	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1),(0, 0)	16 × 16 × 256
(Block2)	16 × 16 × 256	Conv2d + BatchNorm2d + ReLU	64	1	(1, 1),(0, 0)	16 × 16 × 64
	16 × 16 × 64	ConvTranspose2d + BatchNorm2d + ReLU	64	4	(2, 2),(1, 1)	32 × 32 × 64
	32 × 32 × 64	Conv2d + BatchNorm2d + ReLU	128	1	(1, 1),(0, 0)	32 × 32 × 128
(Block1)	32 × 32 × 128	Conv2d + BatchNorm2d + ReLU	32	1	(1, 1),(0, 0)	32 × 32 × 32
	32 × 32 × 32	ConvTranspose2d + BatchNorm2d + ReLU	32	4	(2, 2),(1, 1)	64 × 64 × 32
	64 × 64 × 32	Conv2d + BatchNorm2d + ReLU	32	1	(1, 1),(0, 0)	64 × 64 × 32
(final)	64 × 64 × 32	Conv2d	3	1	(1, 1),(0, 0)	64 × 64 × 3
	64 × 64 × 3	Upsampling by factor 2	–	–	–	128 × 128 × 3

Table 6
The detailed structure of the Deeplabv3+-based mask decoder network.

Layers	Input	Layer type	Filters	Kernel size	Stride and padding	Dilation and groups	Output shape
ASPP	4 × 4 × 1024	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1), (0, 0)	(1, 1), 1	4 × 4 × 256
	4 × 4 × 1024	Conv2d	1024	3	(1, 1), (12, 12)	(12, 12), 1024	4 × 4 × 1024
	4 × 4 × 1024	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1), (0, 0)	(1, 1), 1	4 × 4 × 256
	4 × 4 × 1024	Conv2d	1024	3	(1, 1), (24, 24)	(24, 24), 1024	4 × 4 × 1024
	4 × 4 × 1024	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1), (0, 0)	(1, 1), 1	4 × 4 × 256
	4 × 4 × 1024	Conv2d	1024	3	(1, 1), (36, 36)	(36, 36), 1024	4 × 4 × 1024
	4 × 4 × 1024	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1), (0, 0)	(1, 1), 1	4 × 4 × 256
	4 × 4 × 1024	AdaptiveAvgPool2d	–	–	–	–	1 × 1 × 1024
	1 × 1 × 1024	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1), (0, 0)	(1, 1), 1	4 × 4 × 256
	5 × (4 × 4 × 256)	Concatenation	–	–	–	–	4 × 4 × 1280
	4 × 4 × 1280	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1), (0, 0)	(1, 1), 1	4 × 4 × 256
	4 × 4 × 256	Dropout	256	–	–	–	4 × 4 × 256
Decoder_block	4 × 4 × 256	Upsampling by factor 8	–	–	–	–	32 × 32 × 256
	32 × 32 × 128	Conv2d + BatchNorm2d + ReLU	48	1	(1, 1), (0, 0)	(1, 1), 1	32 × 32 × 48
	(32 × 32 × 48) (32 × 32 × 256)	Concatenation	–	–	–	–	32 × 32 × 304
	32 × 32 × 304	Conv2d	304	3	(1, 1), (1, 1)	(1, 1), 304	32 × 32 × 304
	32 × 32 × 304	Conv2d + BatchNorm2d + ReLU	128	1	(1, 1), (0, 0)	(1, 1), 1	32 × 32 × 128
	32 × 32 × 128	Conv2d + BatchNorm2d + ReLU	32	1	(1, 1), (0, 0)	(1, 1), 1	32 × 32 × 32
	4 × 4 × 256	Upsampling by factor 2	–	–	–	–	64 × 64 × 32

diverse appearance and size. Following the ASPP, the features undergo further refinement through 1 × 1 convolution, batch normalization, and ReLU activation, enhancing the network's ability to perform detailed segmentation. Additional layers of 3 × 3 and 1 × 1 separable convolutions ensure deeper processing and more precise feature delineation, essential for producing high-accuracy crack masks. More details on the specific configurations and layers within the DeepLabv3+ architecture are provided in Table 6. The specific expression of the DeepLabv3+ is shown in Eq. (6)

$$I_{DeepLabv3+} = \varphi_{1\times1}(\varphi_{3\times3}([\varphi_{1\times1}(\hat{I}_1)\| \uparrow_8 (\zeta(\hat{I}_4))])), \tag{6}$$

where $I_{DeepLabv3+}$ denotes the output of the DeepLabv3+, $\hat{I}_1$ and $\hat{I}_4$ denote to the feature maps output of the first and last encoder block, ζ represents the ASPP module, $\uparrow_8$ represents the upsampling operation by factor 8, $\varphi_{1\times1}$ and $\varphi_{3\times3}$ represent 1 × 1 and 3 × 3 convolutions, respectively, and $\|$ represents the concatenation operation.

Parallel to DeepLabv3+, the **LinkNet network** within the Crack Segmentation Decoder is similar to that used in the reconstruction decoder, as described in the earlier section. The primary distinction in its application here lies in its input; the LinkNet decoder processes a concatenated output comprising the high-level features from the encoder and the enriched feature set from the ASPP module of DeepLabv3+, as shown in Eq. (7).

$$\hat{A}_4 = \varphi_{1\times1}(\psi_{4\times4}(\varphi_{1\times1}(\zeta(\hat{I}_4) \parallel \hat{I}_4))), \tag{7}$$

$$A_4 = (\hat{A}_4 \oplus \hat{I}_{EFF}) \parallel \hat{I}_{n-1}, \tag{8}$$

$$A_n = \varphi_{1\times1}(\psi_{4\times4}(\varphi_{1\times1}(A_{n+1}))), n \in \{3, 2, 1\} \tag{9}$$

where $\hat{A}_4$ denotes the output of the decoder block 4 and A_4 is the input of the next decoder, A_n is the output of the decoder blocks, $\hat{I}_4$ represents the feature maps of the last encoder block, ζ represents the ASPP module, $\varphi_{1\times1}$ represents 1 × 1 convolution, $\psi_{4\times4}$ represents the transposed convolution operation with a kernel size of 4 × 4 and stride s = 2, $\hat{I}_{EFF}$ represents the output of the EFF block, $\hat{I}_{n-1}$ denotes to the corresponding feature maps of the encoder, $\oplus$ represents the elementwise addition, and $\|$ represents the concatenation operation. Detailed information about the LinkNet architecture and its layers can be found in Table 7. Integrating the DeepLabv3+ and LinkNet architectures through the Enhanced Feature Fusion (EFF) block, detailed in the next subsection, CrackMaster produces finely segmented outputs by merging results from both decoders. To prevent overfitting and ensure good generalization across different scenarios, the combined features undergo a dropout process with a rate of 0.2, randomly omitting units during training. These features are then refined with a 1 × 1 convolutional layer and upscaled using bilinear interpolation to match the original image size accurately, as shown in Eq. (10):

$$I_{seg} = \uparrow_2 (\varphi_{1\times1}(A_1 \oplus I_{DeepLabv3+})), \tag{10}$$

where I_{seg} denotes the segmentation image, A_1 represents the output of the last decoder block, $I_{DeepLabv3+}$ represents the output of the deeplabv3+ decoder, and $\uparrow_2$ represents the upsampling operation by factor 2.

This dual-decoder strategy combines DeepLabv3+'s contextual understanding and LinkNet's precision, resulting in high-quality segmentation masks essential for detailed structural analysis. The detailed architecture is provided in Table 8.

Table 7
The detailed structure of the LinkNet-based mask decoder network.

Layers	Input	Layer type	Filters	Kernel size	Stride and padding	Dilation and groups	Output shape
(Block4)	4 × 4 × 1280	Conv2d + BatchNorm2d + ReLU	320	1	(1, 1),(0, 0)	(1, 1), 1	4 × 4 × 320
	4 × 4 × 320	ConvTranspose2d + BatchNorm2d + ReLU	320	4	(2, 2),(1, 1)	(1, 1), 1	8 × 8 × 320
	8 × 8 × 320	Conv2d + BatchNorm2d + ReLU	512	1	(1, 1),(0, 0)	(1, 1), 1	8 × 8 × 512
(Block3)	8 × 8 × 1024	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1),(0, 0)	(1, 1), 1	8 × 8 × 256
	8 × 8 × 256	ConvTranspose2d + BatchNorm2d + ReLU	128	4	(2, 2),(1, 1)	(1, 1), 1	16 × 16 × 256
	16 × 16 × 256	Conv2d + BatchNorm2d + ReLU	256	1	(1, 1),(0, 0)	(1, 1), 1	16 × 16 × 256
(Block2)	16 × 16 × 512	Conv2d + BatchNorm2d + ReLU	128	1	(1, 1),(0, 0)	(1, 1), 1	16 × 16 × 128
	16 × 16 × 128	ConvTranspose2d + BatchNorm2d + ReLU	128	4	(2, 2),(1, 1)	(1, 1), 1	32 × 32 × 128
	32 × 32 × 128	Conv2d + BatchNorm2d + ReLU	128	1	(1, 1),(0, 0)	(1, 1), 1	32 × 32 × 128
(Block1)	32 × 32 × 256	Conv2d + BatchNorm2d + ReLU	64	1	(1, 1),(0, 0)	(1, 1), 1	32 × 32 × 64
	32 × 32 × 64	ConvTranspose2d + BatchNorm2d + ReLU	64	4	(2, 2),(1, 1)	(1, 1), 1	64 × 64 × 64
	64 × 64 × 64	Conv2d + BatchNorm2d + ReLU	32	1	(1, 1),(0, 0)	(1, 1), 1	64 × 64 × 32

Table 8
The detailed structure of the head segmentation block.

Layers	Input	Layer type	Filters	Kernel size	Stride and padding	Dilation and groups	Output shape
(final)	64 × 64 × 32	Dropout	32	–	–	–	64 × 64 × 32
	64 × 64 × 32	Conv2d	2	1	(1, 1),(0, 0)	(1, 1), 1	64 × 64 × 2
	64 × 64 × 2	Upsampling by factor 2	–	–	–	–	128 × 128 × 2

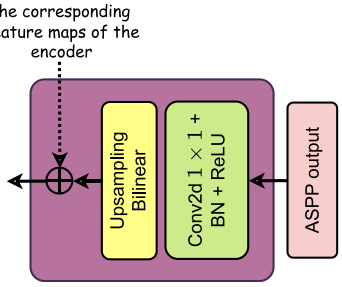


Fig. 4. The internal structure of EFF.

3.2.3. Enhanced Feature Fusion (EFF) block

Fig. 4 illustrates the architecture of the EFF block. The EFF block begins with a 1 × 1 convolution layer that refines channel dimensions by compressing and transforming features without changing their spatial structure. This is followed by batch normalization to stabilize learning by standardizing activations and mitigating internal covariate shifts. A ReLU activation then introduces non-linearity to enhance the model's ability to learn complex patterns.

Before merging with skip connection feature maps, the processed features undergo upsampling to restore spatial dimensions reduced by earlier convolutional compressions. This alignment is crucial for the fusion process, where upsampled features are added to skip connection feature maps. This addition integrates deep, semantic-rich information from the EFF block with detailed spatial information from earlier network layers, enhancing segmentation accuracy and performance by preserving important spatial details and context throughout the process, the specific expression of the EFF block is shown in Eq. (11).

$$\hat{I}_{EFF} = (\uparrow_2 (\varphi_{1\times1}(\hat{I}_{ASPP}))) \oplus \hat{I}_n, \tag{11}$$

where $\hat{I}_{EFF}$ represents the output of EFF block, $\hat{I}_{ASPP}$ represents the output of ASPP module, $\varphi_{1\times1}$ stands for the 1 × 1 convolution, batch normalization and ReLU activation, and $\uparrow_2$ represents the upsampling operation by factor value of 2.

During the training phase, the entire network, which combines the encoder and the dual-decoder networks as depicted in Fig. 2, is optimized. In contrast, only the trained encoder and the crack segmentation decoder are employed during the testing phase to segment cracks in the images.

3.3. Training

The training process for the CrackMaster involves optimizing a composite network structure that integrates an encoder with dual-decoder branches. The encoder processes the input images to extract feature maps, which are then fed to both decoders to perform their respective tasks—image reconstruction and crack segmentation. To effectively train the network, a loss function aggregates the losses from both decoders. Let L_{total} represent the total objective loss for the network, L_{rec} the loss from the reconstruction decoder, and L_{seg} the loss from the segmentation decoder. The total loss is a sum of these two losses:

$$L_{total} = L_{rec} + L_{seg}, \tag{12}$$

The network utilizes the $Huber$ loss function for the image reconstruction decoder, which balances the $L1$ and $L2$ losses, providing robustness against outliers and ensuring stable gradient updates. This loss function is defined as:

$$L_{rec}(\hat{A}, A) = \frac{1}{N}\sum_{i=1}^{N} \text{Huber}(A_i - \hat{A}_i), \tag{13}$$

where A_i is the actual pixel value, $\hat{A}_i$ is the predicted pixel value from the reconstruction decoder, and N is the total number of pixels.

$$\text{Huber}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise}, \end{cases} \tag{14}$$

where x is $A_i - \hat{A}_i$.

We extensively tested various crack segmentation loss functions (L_{seg}) to improve performance and showcase our model's strength in its core architecture, not solely in loss function optimization. By successfully using different functions L_{seg}, including simple ones, we validated our hypothesis that the model's robust segmentation capabilities stem from its sophisticated design.

The initial crack segmentation loss function tested is the cross-entropy (CE) loss, which quantifies the disparity between the predicted and actual probability distributions of crack and non-crack pixels. It penalizes the model according to the inconsistency between its predictions and the ground truth labels. It can be defined as:

$$L_{\text{seg}}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} \left(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right), \tag{15}$$

where y_i is the ground truth label (1 for crack, 0 for non-crack), $\hat{y}_i$ represents the predicted probability of the crack class, and N is the total number of images in the batch.

Table 9
Details of the various crack datasets.

Dataset	Image resolution	Training set	Test set
RCD	128 × 128	5041	1205
DeepCrack537	512 × 512	422	115
YCD	512 × 512	622	154

The second crack segmentation loss function tested is the Dice loss, which is a similarity measure commonly used in segmentation tasks. It quantifies the overlap between the predicted and ground truth segmentation masks. Mathematically, it is defined as:

$$L_{\text{seg}}(y,\hat{y}) = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i + \epsilon}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + \epsilon}, \tag{16}$$

where, ϵ is a smoothing factor that prevents division by zero, stabilizing the loss calculation. In this work, we set $\epsilon = 1e{-}3$.

The third crack segmentation loss function tested is the Focal Tversky loss. This loss function is an extension of the Tversky loss, a variation of Dice loss. The Focal Tversky loss incorporates a focal term inspired by the focal loss used in object detection tasks. This focal term down-weights well-classified pixels and emphasizes complex examples, helping to address the class imbalance and focusing more on challenging regions in the segmentation task. The Focal Tversky loss is defined as:

$$L_{\text{seg}}(y,\hat{y}) = \frac{\sum_{i=1}^{N} T_{\beta}(y_i,\hat{y}_i)\left(1 - \frac{T_{\alpha}(y_i,\hat{y}_i)}{T_{\beta}(y_i,\hat{y}_i)}\right)^{\gamma}}{\sum_{i=1}^{N} T_{\beta}(y_i,\hat{y}_i)}, \tag{17}$$

where the components of the Tversky index, $T_{\alpha}(y_i,\hat{y}_i)$ and $T_{\beta}(y_i,\hat{y}_i)$, are involved, while γ acts as the focusing parameter, set to a value between 0 and 1.

The entire network, including both decoders, is optimized using the AdamW optimizer, an extension of the Adam optimizer that includes decoupled weight decay regularization.

Empirically, the most effective objective loss function for training the entire model is cross-entropy for segmentation, complemented by $Huber$ loss as the reconstruction loss function.

4. Experimental dataset and setup

This section introduces the implementation details, datasets utilized, evaluation metrics employed, and experiments conducted to assess the CrackMaster model's performance.

4.1. Datasets

To demonstrate the effectiveness of the proposed CrackMaster in pixel-wise crack segmentation tasks, we conducted experiments using a diverse set of datasets, comprising one private dataset, called the RCD dataset, and two publicly available datasets known for the crack segmentation tasks: the DeepCrack537 dataset, and the YCD dataset. These datasets were selected to comprehensively evaluate the CrackMaster's performance across different crack segmentation scenarios. Fig. 5 depicts sample images from each dataset, showing various crack types and background complexities. Notably, the images exhibit a range of crack sizes, from minor cracks occupying small-scale pixels to more extensive crack patterns, presenting challenges such as class imbalance and complex background interference.

These datasets offer valuable opportunities for evaluating and validating the CrackMaster network's performance under diverse real-world conditions. Further details regarding the specifications of these datasets are provided in Table 9.

(1) ***RCD*** (Okran et al., 2023): Derived from the RDD2022 (Arya et al., 2022) dataset, this private dataset is a critical resource for crack detection research, offering 5041 training images and 1205 testing images. The dataset includes images with scale and size variations to reflect real-world crack scenarios. To ensure uniformity and efficient model training, all images were standardized to a resolution of 128 × 128 pixels. Despite the sub-optimal image quality, which presents challenges for accurate annotation and model training, the dataset is valuable for its diverse representation of crack scenarios. Annotation was conducted using a polygonal region approach for cracks resembling alligators, considering image quality, the time-consuming nature of annotation, and the variability of real-world cracks. This nuanced annotation enriches the dataset, providing researchers valuable data for developing robust crack detection algorithms.

(2) ***DeepCrack537*** (Liu et al., 2019): Compiled by Liu et al. comprises a diverse collection of crack images to facilitate robust crack detection and segmentation tasks. With 537 complex crack images carefully selected to represent various scenarios and scales, including bare, rough, and dirty types, the dataset provides comprehensive coverage of crack characteristics. While the images' original size is 544 × 384 pixels, we standardized them to 512 × 512 pixels to train and evaluate our model performance, ensuring consistency across the dataset and enabling efficient model training and evaluation. The dataset is further partitioned into training and test sets, with a division ratio of 8:2. Specifically, the training dataset consists of 422 images, and the test dataset comprises 115 images, facilitating rigorous evaluation of model performance. This dataset is a valuable resource for advancing crack detection and segmentation.

(3) ***YCD*** (Yang et al., 2018): Assembled by Yang et al. it represents a significant resource in crack detection research, comprising a comprehensive collection of 776 crack images. Notably, these images exhibit variations in size and scale, presenting a diverse range of crack scenarios and complexities. The images were adjusted to a standardized dimension of 512 × 512 pixels to ensure uniformity and facilitate robust model training and evaluation. This pre-processing step enables consistent analysis and comparison across the dataset. The dataset is divided into a training set comprising 622 images and a test set of 154 images. This structured division allows for rigorous evaluation and validation of model performance across different scenarios and scales, enhancing the reliability and generalizability of crack detection methodologies developed using this dataset.

(4) ***Tarragona Road Crack Dataset (TRCD)***: For the pilot deployment of our framework, we collected a real dataset from Tarragona City in Spain. The dataset comprises 310 images captured using a Smartphone and GoPro camera to ensure a variety of perspectives and resolutions. This diverse collection of photographs includes various types of road surfaces and crack patterns, providing a robust basis for evaluating our model's performance in real-world conditions. Different devices allowed us to test the framework's adaptability to varying image qualities and environmental conditions, such as lighting (e.g., day and night) and weather variations (e.g., sunny and cloudy). This dataset is critical in validating our proposed framework's practical applicability and effectiveness in detecting and segmenting road surface cracks in urban infrastructure.

4.2. Data augmentation

Our methodology leveraged various augmentation techniques to enhance our dataset's diversity and improve our models' generalization ability. These techniques, available through the MMsegmentation framework (Contributors, 2020), include random rotation, horizontal

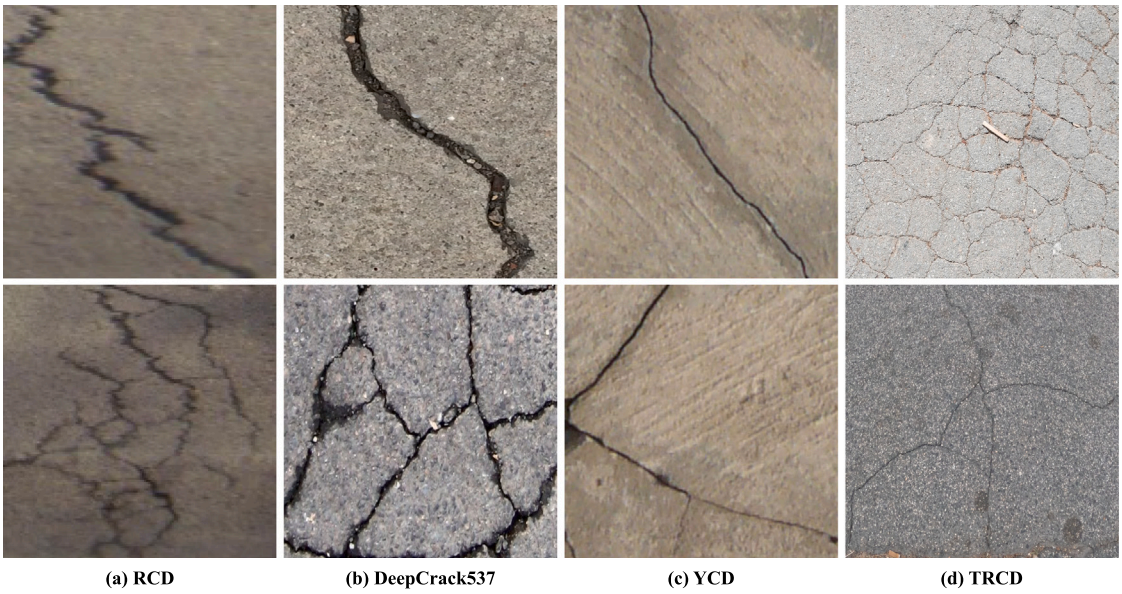


Fig. 5. Sample crack images from the datasets used in the experiments.

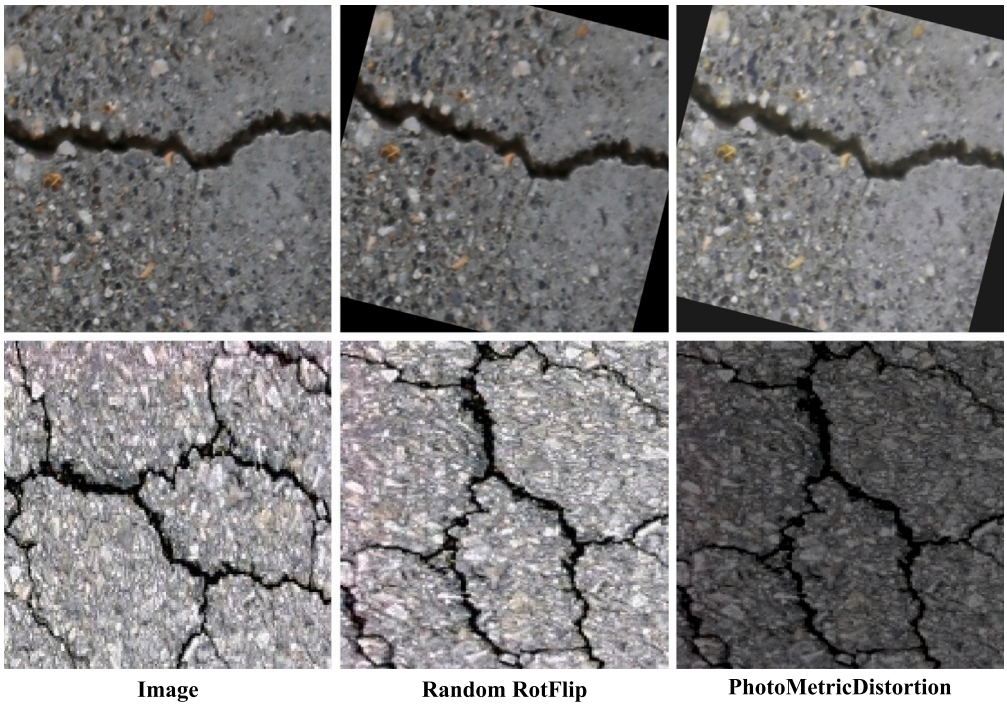


Fig. 6. Examples of data augmentation transformation applied to the images in the training sets.

and vertical flipping, and photometric distortion. The latter encompasses a range of transformations such as random brightness, contrast adjustment, colour space conversion from BGR to HSV, random saturation and hue changes, and conversion back to BGR. Each augmentation was sequentially applied to the images, with a probability of occurrence set at 0.5. By incorporating these techniques, we diversified our dataset and augmented its richness, enhancing our models' capacity to generalize across different crack detection scenarios. Fig. 6 illustrates several examples showing the transformations applied to each input image within the dataset.

4.3. Implementation details

The MMsegmentation framework (Contributors, 2020) was employed for model implementation, and all computations were conducted on a 64-bit Core i7-9700K CPU operating at 3.60 GHz with 64 GB of memory alongside an NVIDIA GeForce RTX 3080 GPU with 10 GB on Windows 10 Pro. AdamW optimizer (Loshchilov and Hutter, 2017) was utilized through experimentation. We set the batch size to 16 for the RCD dataset based on two factors: the memory size of the

GPU card and the image size, which is 128 × 128—the total number of rounds set to 20 000 iterations.

Building upon the previous setup, we fine-tuned the best-performing model from the RCD dataset experiments to tailor it for the DeepCrack537 and YCD datasets. Given the larger image size of 512 × 512 pixels, the batch size was adjusted to 2, balancing the GPU's memory limitations with sufficient gradient estimation. The rigorous training regimen consisted of 20 000 iterations to ensure comprehensive learning and model stability.

4.4. Hyperparameter tuning

Hyperparameter (HP) fine-tuning is a critical component in DL models to achieve high model performance, stability, and generalization across a variety of datasets. Essential parameters like learning rate, batch size, optimizer settings, and loss functions are carefully chosen to optimize model efficiency and accuracy (Simon et al., 2022). In our work, unlike traditional crack segmentation methods, we applied tailored HP settings to meet the needs of the tested datasets and maintain general consistency across our experiments. The only parameter to change from one dataset to another is the size of the input image. For the DeepCrack537 and YCD datasets, the image size is 512 × 512 pixels, while for the RCD dataset, it is 128 × 128.

We began with an initial learning rate of $1e{-}04$ for all datasets and used a scheduled adjustment strategy to stabilize the training process. Specifically, we employed a LinearLR scheduler with a starting factor of 1e−6 over the initial 3000 iterations, transitioning to a PolyLR scheduler from 3000 to 20 000 iterations, with a power of 1.0 and an eta_min of 0.0. This combination allowed the model to warm up gradually and then adaptively reduce the learning rate, which helped achieve effective convergence. Batch sizes were adjusted according to GPU memory limits, especially when dealing with larger images, as seen in the DeepCrack537 and YCD datasets, where we set the batch size to 2.

4.5. Evaluation metrics

In this study, we employed eight metrics to gauge the efficacy of the CrackMaster model for crack segmentation. Intersection over Union (IoU), Precision, Recall, Accuracy, Dice Coefficient, Mean Precision (mPrecision), Mean Recall (mRecall), and Mean Intersection over Union (mIoU) metrics offer quantitative assessments of the model's performance in accurately delineating crack regions.

5. Experimental results and discussion

This section comprehensively assesses the proposed crack segmentation network's performance by conducting ablation and comparison experiments with the RCD, DeepCrack537, and YCD datasets. Our evaluation employs eight key metrics to deliver precise quantitative analysis, complemented by visualizations for qualitative insights into the segmentation capabilities of CrackMaster and other state-of-the-art segmentation networks. In addition to the primary evaluation metrics, we plotted the Receiver Operating Characteristic (ROC) curves and computed the Area Under the Curve (AUC) values to further assess the model's segmentation performance. The AUC-ROC is widely recognized as a robust metric for evaluating classification performance across varying thresholds, particularly in imbalanced datasets (Bradley, 1997).

5.1. Ablation study

In this section, we detail the construction of our proposed framework, CrackMaster, and evaluate the effectiveness of various networks used for both the encoder and decoder. Initially, we selected U-Net (Ronneberger et al., 2015) as the decoder and conducted experiments with different state-of-the-art backbones to determine the best encoder. We trained our model with each backbone using the RCD dataset (Okran et al., 2023), employing the CE loss function. The encoder achieving the highest mean Intersection over Union (mIoU) was chosen. The tested backbones included ResNet18, ResNet34, ResNet50, ResNet101 (He et al., 2016), HRNet-w48, HRNet-w48-ssld, HRNet-w64 (Wang et al., 2019), SeNet154 (Hu et al., 2018), EfficientNet-b5 (Tan and Le, 2019), VGG16 (Simonyan and Zisserman, 2014), MobileNetv2 (Sandler et al., 2018), and ConvNeXt-Base (Liu et al., 2022). Through this rigorous selection process, we ultimately identified the optimal encoder for CrackMaster. Table 10 shows the quantitative results for all backbones networks employed, among all tested backbones, ConvNeXt-Base achieved the best result with the most metrics; an IoU of 83.1%, Accuracy of 92.54%, Precision of 87.52%, F1 score (Dice) of 90.77%, mPrecision of 91.84%, mRecall of 92.85%, and mIoU of 85.65%. However, the ResNet50 network fulfilled the best Recall of 94.53% among all tested backbones, slightly different from what the ConvNeXt-Base achieved. ConvNeXt-Base's architecture learns a broader representation of features, enhancing overall accuracy, precision, and other metrics. It may effectively balance detecting true positives while minimizing false positives, leading to better overall metrics except for recall. ResNet50 generates more false positives while having higher recall, reducing its precision.

On the other hand, to choose the decoder network, we took the ConvNeXt-Base network, which was chosen as the best backbone among all backbones from the former step, made it the encoder network, and conducted several experiments with the state-of-the-art decoder networks. Specifically, we tested U-Net (Ronneberger et al., 2015), U-Net++ (Zhou et al., 2018), FPN (Kirillov et al., 2017), PSPNet (Zhao et al., 2017), LinkNet (Chaurasia and Culurciello, 2017), MANet (Fan et al., 2020), UPerNet (Xiao et al., 2018), DeepLabv3 (Chen et al., 2017), and DeepLabv3+ (Chen et al., 2018). RCD dataset and CE loss function were used for training the models, we selected the decoder with the best mIoU metric. As shown in Table 11, DeepLabv3+ yielded the best results for most metrics, achieving an Accuracy of 92.67%, IoU of 83.37%, F1 score (Dice) of 90.93%, mPrecision of 91.98%, mRecall of 92.97%, and mIoU of 85.89%. However, the best Precision value of 87.93% was achieved by LinkNet, while U-Net++ achieved the best Recall value of 94.73%; in turn, LinkNet achieved the same mPrecision of 91.98% as DeepLabv3+. We ended up with the best configuration consisting of ConvNeXt-Base as an encoder and DeepLabv3+ as a decoder, as a **Baseline**.

To assess the efficacy of each component within the CrackMaster framework, we conducted ablation experiments utilizing the RCD dataset. We aimed to refine segmentation precision on crack images by introducing specific network enhancements, including the proposed LinkNet component, EFF component, and the SSL "reconstruction" network. The baseline configuration represents the CrackMaster architecture without these additional components. Subsequently, various configurations were devised for the ablation experiments to evaluate the performance of each component:

(1) Baseline (ConvNeXt-Base as encoder and DeepLabv3+ as decoder),
(2) Baseline + LinkNet,
(3) Baseline + LinkNet + EFF,
(4) Baseline + LinkNet + EFF + SSL,
(5) Proposed CrackMaster. Configuration (4) + skip connection concatenation between the encoder features and the corresponding features in the decoder.

Table 10
The ablation experiments on the RCD dataset for choosing the best encoder. The red bold text indicates the best results, while the blue bold text represents the second-best results.

Encoder	Decoder	IoU	Accuracy	Precision	Recall	Dice	mPrecision	mRecall	mIoU
ResNet18	U-Net	80.27%	91.08%	85.29%	81.20%	89.06%	90.33%	91.46%	83.14%
ResNet34		80.82%	91.54%	***87.32%***	91.56%	89.39%	90.88%	91.54%	83.83%
ResNet50		80.38%	91.02%	84.31%	***94.53%***	89.13%	90.26%	91.65%	83.08%
ResNet101		81.20%	91.66%	86.97%	92.45%	89.62%	90.97%	91.81%	84.09%
HRNet-w48		74.74%	88.47%	83.56%	87.63%	85.54%	87.71%	88.32%	78.62%
HRNet-w48-ssld		74.73%	88.44%	83.35%	87.85%	85.54%	87.66%	88.33%	78.58%
HRNet-w64		75.61%	88.86%	83.69%	88.68%	86.11%	88.09%	88.83%	79.30%
SeNet154		81.37%	91.67%	86.29%	93.45%	89.73%	90.94%	91.99%	84.14%
EfficientNet-b5		***82.06%***	***92.02%***	86.86%	93.69%	***90.15%***	***91.31%***	***92.33%***	***84.75%***
VGG16		81.98%	91.98%	86.73%	93.73%	90.10%	91.26%	92.29%	84.67%
MobileNetv2		80.69%	91.28%	85.43%	93.56%	89.31%	90.53%	91.69%	83.48%
ConvNeXt-Base		***83.10%***	***92.54%***	***87.52%***	***94.27%***	***90.77%***	***91.84%***	***92.85%***	***85.65%***

Table 11
The ablation experiments on the RCD dataset for choosing the best decoder. The red bold text indicates the best results, while the blue bold text represents the second-best results.

Encoder	Decoder	IoU	Accuracy	Precision	Recall	Dice	mPrecision	mRecall	mIoU
ConvNeXt-Base	U-Net	83.10%	92.54%	87.52%	94.27%	90.77%	91.84%	92.85%	85.65%
	U-Net++	82.78%	92.33%	86.77%	***94.73%***	90.58%	91.60%	92.76%	85.31%
	FPN	83.23%	92.60%	87.60%	94.36%	90.85%	***91.90%***	***92.92%***	85.77%
	PSPNet	77.70%	89.79%	83.87%	91.35%	87.45%	89.01%	90.08%	80.93%
	LinkNet	***83.29%***	***92.65%***	***87.93%***	94.05%	***90.88%***	***91.98%***	92.91%	***85.85%***
	MANet	82.35%	92.13%	86.72%	94.23%	90.32%	91.41%	92.51%	84.96%
	UPerNet	83.15%	92.55%	87.48%	***94.37%***	90.80%	91.85%	92.88%	85.68%
	DeepLabv3	77.87%	89.87%	83.87%	91.60%	87.56%	89.09%	90.18%	81.06%
	DeepLabv3+	***83.37%***	***92.67%***	***87.74%***	94.36%	***90.93%***	***91.98%***	***92.97%***	***85.89%***

Table 12
The ablation experiments on the RCD dataset.

Method	IoU	Accuracy	Precision	Recall	Dice	mPrecision	mRecall	mIoU
Baseline	83.37%	92.67%	87.74%	94.36%	90.93%	91.98%	92.97%	85.89%
Baseline+LinkNet	84.57%	93.29%	88.93%	94.53%	91.64%	92.64%	93.51%	86.97%
Baseline+LinkNet+EFF	85.26%	93.63%	89.56%	94.66%	92.04%	93.02%	93.81%	87.58%
Baseline+LinkNet+EFF+SSL	85.36%	93.66%	89.44%	***94.93%***	92.10%	93.04%	93.89%	87.65%
Proposed	***86.03%***	***94.04%***	***90.71%***	94.34%	***92.49%***	***93.50%***	***94.09%***	***88.30%***

As presented in Table 12, our ablation study meticulously dissects the enhancements introduced to the baseline crack segmentation model and their impact on performance within the RCD dataset. The Baseline model set the initial standard, achieving an IoU of 83.37%, Accuracy of 92.67%, and Precision of 87.74%, alongside notable Recall, Dice, mPrecision, mRecall, and mIoU scores.

Combining the LinkNet architecture with the DeepLab network as a decoder in the Baseline significantly improved the model's IoU to 84.57%, indicating a more accurate overlap in segmentation. Accuracy increased to 93.29% and Precision to 88.93%, with the Dice score rising to 91.64%. This indicates that LinkNet enhances the Precision of predictions and the overall segmentation fidelity.

By incorporating EFF, the model saw further improvements, particularly with IoU rising to 85.26%, illustrating that EFF successfully enriches the feature space, leading to more precise segmentation. Accuracy also increased slightly to 93.63% and Precision to 89.56%, demonstrating the positive impact of EFF on the model's ability to classify pixels as crack or non-crack correctly. Including the reconstruction decoder advanced the model's Recall to 94.93% and boosted the Dice score to 92.10%. These enhancements underscore SSL's critical role in improving the model's sensitivity and Accuracy in identifying actual crack areas.

Our Proposed model, which synergizes all the enhancements above, achieved the most substantial improvements across all metrics. The IoU saw a significant rise to 86.03%, Accuracy reached 94.04%, and Precision climbed to 90.71%. The model recorded the highest Dice score at 92.49% and showcased remarkable mPrecision and mRecall, indicative of its overall segmentation prowess. The proposed model's mIoU of 88.30% reflects the synergistic effect of the incorporated components, signalling an overall superior segmentation capability. This thorough ablation study verifies that each enhancement, e.g., adding LinkNet to the decoder, EFF, and the reconstruction decoder, selectively contributes to elevating the performance of the crack segmentation model. In addition, in our ablation study, we examine the performance of the CrackMaster model using various combinations of segmentation and reconstruction loss functions on the RCD dataset. Segmentation losses include Cross-Entropy (CE), Dice, and Focal Tversky (FT), while Huber is used for reconstruction. As we can see in Table 13, the results reveal that the CE loss for the segmentation task and Huber for the reconstruction task achieve the highest Intersection over Union (IoU) of 86.03%, along with an accuracy of 94.04% and precision of 90.71, indicating robust segmentation performance. The Huber loss is shared across all segmentation loss combinations, providing a consistent framework for comparison. When using the Dice loss, the model achieves a notable recall of 95.08%, although it slightly underperforms in precision and IoU compared to CE. The FT loss shows balanced performance metrics with an accuracy of 93.63% and a precision of 89.47%. Combinations of segmentation losses such as CE + Dice, CE + FT, and Dice + FT produce varied results. The combination of CE + Dice achieves the highest recall value of 96.02%, however, underperforms in precision. Furthermore, the combination of CE + FT achieves a notable precision of 89.98% and recall of 94.33%. However, combining all three segmentation losses (CE + Dice + FT) results in the lowest IoU of 84.75%, though it maintains a solid precision of 89.55%. Overall, the study indicates that while CE loss alone provides the best IoU and balanced metrics, different combinations of segmentation and reconstruction losses can optimize specific performance aspects, such as precision or recall, depending on the application's requirements.

Table 13
Comparison of CrackMaster performance using different loss functions on RCD.

Seg. Loss	Rec. Loss	IoU	Accuracy	Precision	Recall	Dice	mPrecision	mRecall	mIoU
CE		***86.03%***	***94.04%***	***90.71%***	94.34%	***92.49%***	***93.50%***	***94.09%***	***88.30%***
Dice		85.64%	93.79%	89.61%	95.08%	92.26%	93.17%	94.02%	87.89%
FT		85.29%	93.63%	89.47%	94.81%	92.06%	93.01%	93.85%	87.60%
CE + Dice	Huber	85.14%	93.47%	88.25%	***96.02%***	91.97%	92.78%	93.93%	87.36%
CE + FT		85.36%	93.70%	89.98%	94.33%	92.10%	93.12%	93.81%	87.70%
Dice + FT		85.19%	93.57%	89.17%	95.02%	92.00%	92.93%	93.83%	87.49%
CE + Dice + FT		84.75%	93.41%	89.55%	94.06%	91.75%	92.82%	93.53%	87.18%

Table 14
Comparative results across various methods on RCD dataset. (–) Dashed cells indicate instances where no results were provided. The red bold text indicates the best results, while the blue bold text represents the second-best results.

Method	Type	IoU	Precision	Recall	Dice	mIoU
U-Net (Ronneberger et al., 2015)	CNN	80.4%	84.3%	***94.5%***	89.1%	83.1%
FPN (Kirillov et al., 2017)		81.4%	86.8%	92.8%	89.7%	84.2%
PSPNet (Zhao et al., 2017)		73.7%	82.3%	87.6%	84.8%	77.6%
LinkNet (Chaurasia and Culurciello, 2017)		81.6%	87.4%	92.5%	89.9%	84.5%
DeepLabv3 (Chen et al., 2017)		74.8%	82.7%	88.7%	85.6%	78.5%
DeepLabv3+ (Chen et al., 2018)		79.7%	84.8%	93.0%	88.7%	82.6%
U-Net++ (Zhou et al., 2018)		81.1%	85.4%	94.1%	89.6%	83.8%
UPerNet (Xiao et al., 2018)		81.6%	87.5%	92.3%	89.9%	84.4%
MANet (Fan et al., 2020)		82.1%	87.5%	93.1%	90.2%	84.9%
Okran et al. (2023)		81.7%	***88.8%***	–	89.5%	–
SwinT (Liu et al., 2021)	Transformer	***83.1%***	87.9%	93.8%	***90.8%***	***85.7%***
MobileVIT (Mehta and Rastegari, 2021)		82.2%	85.9%	***94.9%***	90.2%	84.7%
CT-CrackSeg (Tao et al., 2023)		80.6%	84.6%	***94.5%***	89.3%	83.3%
Proposed	CNN	***86.0%***	***90.7%***	94.3%	***92.5%***	***88.3%***

5.2. Comparative studies

This subsection presents a comparative analysis to demonstrate the effectiveness of our proposed CrackMaster model for pixel-level crack segmentation. To assess its performance relative to state-of-the-art methods, we evaluate CrackMaster across three datasets: the private RCD dataset and two widely recognized benchmark datasets, DeepCrack537 and YCD. CrackMaster is compared to 29 methods. Some of them are based on CNN (Ronneberger et al., 2015; Liu et al., 2019; Okran et al., 2023), and others are based on transformers (Liu et al., 2021; Tao et al., 2023; Han et al., 2024), as shown in Tables 14–16. The experiments are designed to evaluate CrackMaster's robustness, adaptability, and segmentation accuracy across different datasets, particularly with focusing on precision in boundary detection and fine cracks. Performance is quantified using five distinct evaluation metrics: : Intersection over Union (IoU), Precision, Recall, Dice Coefficient, and mean IoU (mIoU), offering a comprehensive assessment of the proposed network's capabilities.

Evaluation on RCD: Table 14 highlights the five metrics with the RCD dataset. Our proposed CrackMaster achieves top results in several key metrics, though it does not have the highest Recall. The MobileVIT model attains the best Recall, indicating its strong ability to identify true positive crack pixels. Despite this, CrackMaster demonstrates superior overall performance, with the highest Precision among all models at 90.7%. It also achieves an IoU of 86.0%, indicating the model's strong ability to accurately segment cracks with significant overlap between the predicted regions and the ground truth, highlighting its effectiveness in minimizing false positives and false negatives. Additionally, CrackMaster's Dice score is 92.5%, with mIoU values of 88.30%, reinforcing the model's consistent and high-quality segmentation capabilities. Although CrackMaster's Recall is not the highest, its performance at 94.34% is only slightly less than MobileVIT, which has the highest Recall at 94.9%.

Evaluation on DeepCrack537: The DeepCrack537 dataset serves as a critical benchmark in pixel-wise crack detection, providing a standardized platform for comparing the efficacy of various segmentation networks. The CrackMaster network has been rigorously tested against a comprehensive suite of CNN-based and transformer-based networks as shown in Table 15. Evaluation metrics, including IoU, Precision, Recall, Dice score, and mIoU are utilized to gauge performances. In the experimental results presented in Table 15, CrackMaster demonstrates outstanding performance on the DeepCrack537 dataset, achieving an IoU of 87.8%, Precision of 94.3%, Recall of 92.7%, Dice score of 93.5%, and mIoU of 93.5%. While CrackMaster's Precision is slightly lower than some comparison models – 1.3% less than U-Net3+ and 3.4% less than CrackW-Net's highest Recall – its other evaluation metrics far surpass those of both CrackW-Net and U-Net3+. This highlights CrackMaster's overall superior performance.

Evaluation on YCD: Table 16 provides a comparative analysis of various segmentation methods applied to the YCD dataset, highlighting key performance metrics. The proposed CrackMaster model achieves the highest IoU at 76.9%, demonstrating its superior ability to segment areas of interest compared to the ground truth accurately. Additionally, the model records a Precision of 86.4%, which is competitive but slightly lower than the top Precision of 89.4% achieved by MST-Net. CrackMaster demonstrates superior performance in Recall, achieving a score of 87.6% compared to all other tested models. This exceptional Recall highlights the model's ability to identify true positive crack areas, minimizing missed cracks accurately. This underscores the robustness of CrackMaster in ensuring comprehensive crack detection within the YCD dataset. Additionally, the Dice score is notable at 86.9%, indicating a well-maintained balance between Precision and Recall, crucial for practical applications.

Furthermore, CrackMaster yields the best mean metric, with an mIoU of 87.6%, solidifying its top-tier status and confirming the model's overall segmentation quality across all types of cracks.

Overall, the proposed model demonstrates a comprehensive suite of top-ranking metrics, including the leading IoU, which is exclusive to it in this comparison. Its robust performance across the board positions it as a highly competent solution for crack detection tasks on the YCD dataset, making it a prime candidate for real-world infrastructure maintenance and safety assessment applications.

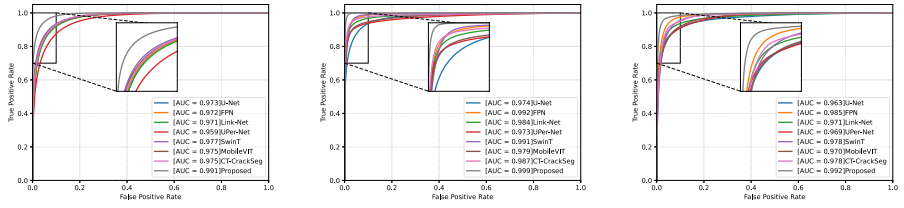
Based on the three datasets, Fig. 7 shows the ROC curve comparison and illustrates the performance of various crack segmentation models in terms of their True Positive Rate (TPR) against False Positive Rate (FPR), with the AUC values serving as a quantitative measure of

Table 15
Comparative results across various methods on DeepCrack537 dataset. (–) Dashed cells indicate instances where no results were provided. The red bold text indicates the best results, while the blue bold text represents the second-best results (Mehta and Rastegari, 2021; Huang et al., 2020; Han et al., 2021; Zhou et al., 2022a; Al-Huda et al., 2023; Yang et al., 2023; Bai et al., 2024; Ju et al., 2022; Guo et al., 2023; Zhang et al., 2023; Liu et al., 2023; Ma et al., 2024b).

Method	Type	Year	IoU	Precision	Recall	Dice	mIoU
U-Net (Ronneberger et al., 2015)		2015	–	84.9%	93.1%	77.7%	81.3%
FPN (Kirillov et al., 2017)		2017	68.7%	87.6%	76.2%	81.5%	83.6%
LinkNet (Chaurasia and Culurciello, 2017)		2017	68.0%	87.1%	75.6%	80.9%	83.2%
UPerNet (Xiao et al., 2018)		2018	79.2%	95.4%	82.3%	88.4%	88.9%
DeepCrack (Liu et al., 2019)		2019	–	91.9%	78.3%	84.6%	86.0%
U-Net3+ (Huang et al., 2020)		2020	–	***95.6%***	86.5%	83.0%	83.2%
CrackW-Net (Han et al., 2021)		2021	–	***97.7%***	65.7%	75.1%	79.1%
TCDNet (Zhou et al., 2022a)		2022	–	94.3%	82.7%	88.3%	89.2%
KTCAM-Net (Al-Huda et al., 2023)		2023	–	88.7%	88.2%	89.2%	88.6%
MST-Net (Yang et al., 2023)		2023	–	94.9%	***95.8%***	90.1%	91.1%
APF-Net (Ma et al., 2024a)		2024	–	89.7%	90.9%	90.9%	90.3%
DEHF-Net (Bai et al., 2024)		2024	–	95.1%	***96.7%***	***92.1%***	***92.4%***
SwinT (Liu et al., 2021)		2021	79.7%	89.7%	87.7%	88.7%	89.2%
MobileVIT (Mehta and Rastegari, 2021)		2021	***80.0%***	92.4%	85.7%	88.9%	89.4%
TransMF (Ju et al., 2022)		2022	–	82.2%	88.1%	85.1%	86.5%
STA (Guo et al., 2023)		2023	–	89.1%	89.1%	89.1%	89.8%
CT-CrackSeg (Tao et al., 2023)	Transformer	2023	79.9%	89.6%	88.1%	88.8%	89.3%
UTCD-Net (Zhang et al., 2023)		2023	–	90.4%	83.2%	86.7%	86.7%
Crackformer II (Liu et al., 2023)		2023	–	89.9%	87.6%	88.7%	89.3%
TFCF-Net (Ma et al., 2024b)		2024	–	91.4%	90.5%	90.9%	91.4%
Proposed	CNN	2024	***87.8%***	94.3%	92.7%	***93.5%***	***93.5%***

Table 16
Comparative results across various methods on YCD dataset. (–) Dashed cells indicate instances where no results were provided. The red bold text indicates the best results, while the blue bold text represents the second-best results (Chen and Lin, 2021; Yang et al., 2023; Bai et al., 2024; Liu et al., 2023; Han et al., 2024).

Method	Type	Year	IoU	Precision	Recall	Dice	mIoU
U-Net (Ronneberger et al., 2015)		2015	–	84.7%	73.5%	60.0%	66.9%
FPN (Kirillov et al., 2017)		2017	63.8	83.5%	72.9%	77.9%	80.4%
LinkNet (Chaurasia and Culurciello, 2017)		2017	60.7	74.4%	76.8%	75.5%	78.6%
FCN (Yang et al., 2018)		2018	–	81.7%	79.0%	80.0%	–
UPerNet (Xiao et al., 2018)	CNN	2018	63.9	76.9%	79.0%	78.0%	80.4%
U-Net3+ (Huang et al., 2020)		2020	–	84.1%	79.1%	67.9%	70.7%
HACNet-D (Chen and Lin, 2021)		2021	–	79.6%	62.5%	70.0%	75.2%
MST-Net (Yang et al., 2023)		2023	–	***89.4%***	***84.8%***	75.5%	78.7%
DEHF-Net (Bai et al., 2024)		2024	–	***87.8%***	84.6%	73.6%	77.8%
SwinT (Liu et al., 2021)		2021	***69.9%***	85.6%	79.3%	82.3%	83.8%
MobileVIT (Mehta and Rastegari, 2021)		2021	68.4%	81.2%	81.2%	81.2%	82.9%
Crackformer II (Liu et al., 2023)	Transformer	2023	–	82.6%	80.1%	79.5%	82.4%
CT-CrackSeg (Tao et al., 2023)		2023	67.8%	79.5%	82.1%	80.8%	82.5%
MambaCrackNet (Han et al., 2024)		2024	–	85.0%	84.6%	***84.1%***	***85.6%***
Proposed	CNN	2024	***76.9%***	86.4%	***87.6%***	***86.9%***	***87.6%***



(a) RCD dataset (b) DeepCrack537 dataset (c) YCD dataset

Fig. 7. Comparison of ROC curves across different datasets.

model effectiveness. CrackMaster Achieves the highest AUC score of 0.99, indicating near-perfect classification performance with minimal false positives. Transformer-based models demonstrate excellent performance with AUC of 0.98 that are slightly lower than the proposed model. Other models, such as U-Net (0.97), UPer-Net (0.97), and CT-CrackSeg (0.98), perform well but are outperformed by the proposed method. The results validate the contributions of the proposed model, CrackMaster, particularly in addressing limitations such as boundary delineation and class imbalance.

5.3. Qualitative analysis of results

Fig. 8 presents a comparative visual analysis of segmentation results from different models on three crack datasets: RCD, DeepCrack537, and YCD. This comparison provides a clear visual representation of each model's ability to identify and delineate cracks against various surface textures.

As shown in Fig. 8, from top to bottom, the rows represent the raw images, the ground truth annotations, and the segmentation masks

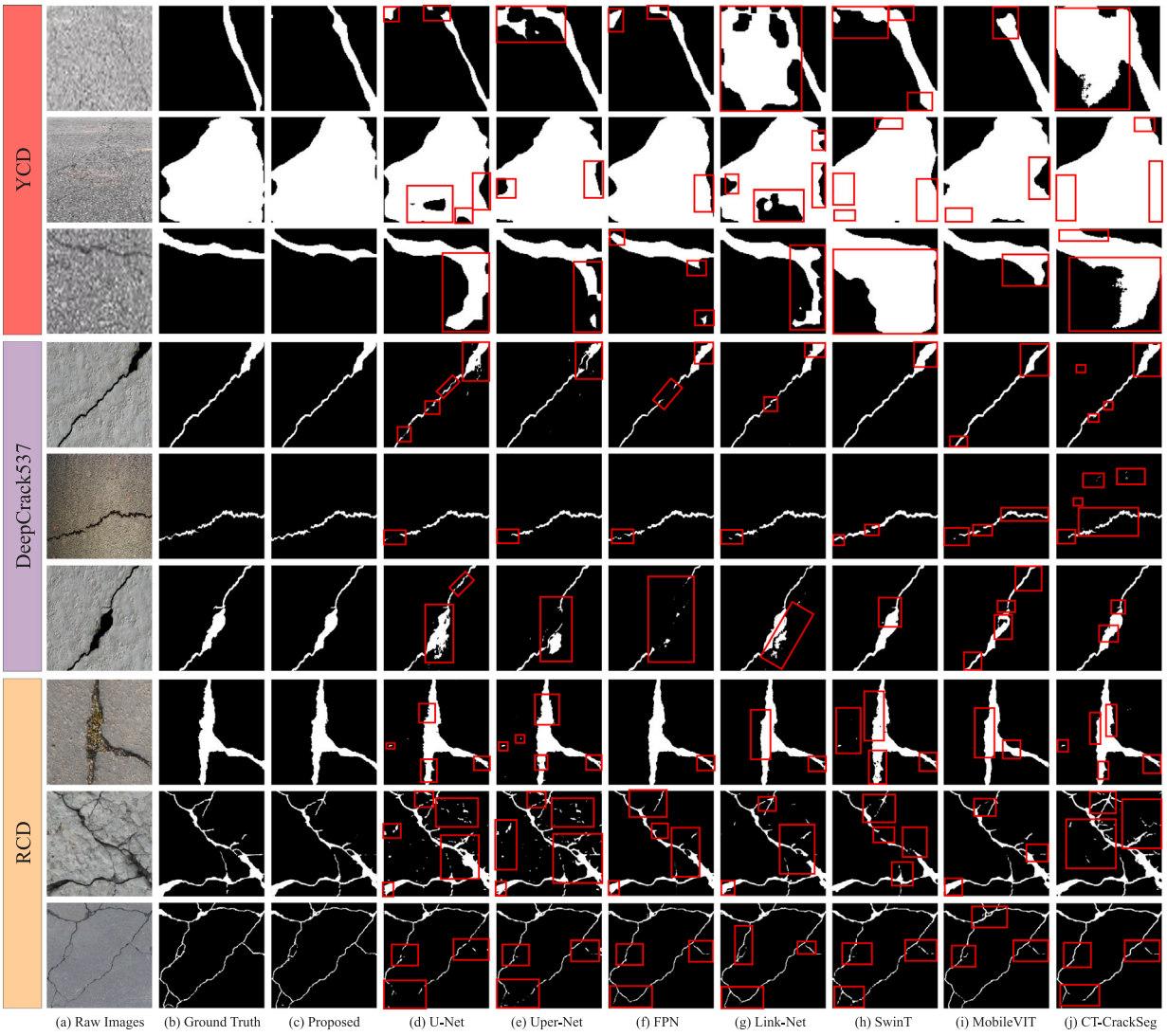
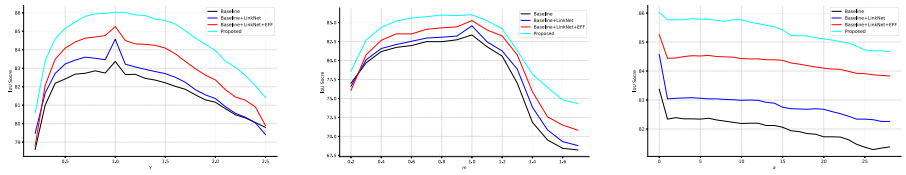


Fig. 8. Visualization results of different methods with the three datasets, RCD, DeepCrack537 and YCD.

generated by the proposed CrackMaster network, followed by the results from U-Net, UPerNet, FPN, LinkNet, SwinT, MobileVIT and CT-CrackSeg, respectively. Each row offers a glimpse into the segmentation effectiveness of each model, with the raw images serving as the baseline for comparison, the ground truth as the segmentation target, and the subsequent rows revealing how closely each model's segmentation masks align with the ground truth. CrackMaster demonstrates precise segmentation for the RCD dataset, closely mirroring the ground truth with minimal noise and false positives. It outperforms other models such as U-Net and UPerNet, which exhibit instances of under-segmentation, as indicated by the red boxes, signifying missed cracks. While FPN also shows under-segmentation to a lesser extent, LinkNet's results include some over-segmentation, where non-crack areas are incorrectly identified as cracks. SwinT shows moderate performance on the RCD dataset, capturing most crack details but occasionally missing finer structures. In the third image of the RCD dataset, SwinT exhibits over-segmentation of the transverse crack, incorrectly identifying non-crack areas as part of the crack. CT-CrackSeg performs worse than SwinT, exhibiting similar issues, including over-segmentation in certain instances, such as in the first and third images of the RCD dataset. CT-CrackSeg falls short compared to MobileVIT, which provides a significant improvement by capturing intricate crack patterns and minimizing both under-segmentation and over-segmentation.

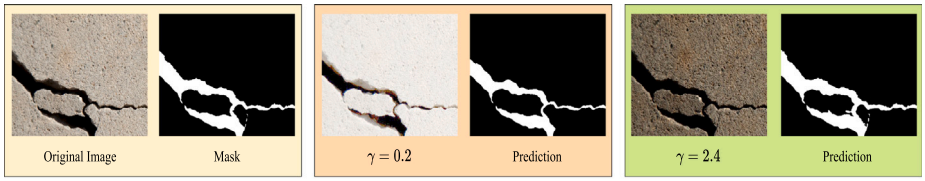
As shown in Fig. 8, On the DeepCrack537 dataset, the proposed model again shows superior performance, capturing finer details and subtler crack features overlooked by models like U-Net, UPerNet, and SwinT. U-Net results illustrate significant under-segmentation and noise, as indicated by the red boxes. Although UPerNet improves upon U-Net, it still struggles to capture all crack details. FPN exhibits both over-segmentation and intermittent detection, missing continuity in crack paths. While LinkNet performs better, the red boxes highlight areas where it fails to adhere to narrow and winding crack paths as effectively as the proposed model. SwinT also shows some improvement and performs better than MobileVIT and CT-CrackSeg on the DeepCrack537 dataset, capturing finer crack details and maintaining better crack continuity. MobileVIT shows moderate performance on the DeepCrack537 dataset, missing some pixels and resulting in incomplete crack detection, similar to CT-CrackSeg. Both models struggle to capture finer crack details, leading to less accurate segmentation compared to other models. This leads to less complete crack detection compared to CrackMaster.

For the YCD dataset, CrackMaster maintains high accuracy with clear and continuous crack delineation. In contrast, U-Net struggles with noise and disjointed crack paths. While UPerNet improves upon U-Net's performance, it can detect the complete crack structure more effectively. FPN exhibits patchy and broken detections, with the red

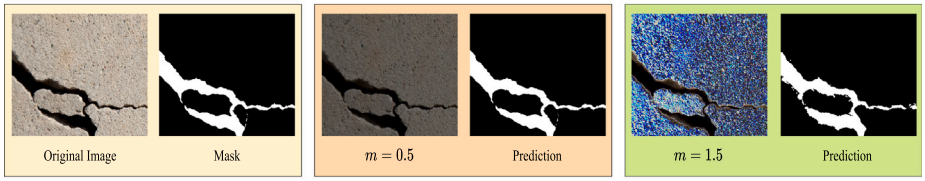


(a) (b) (c)

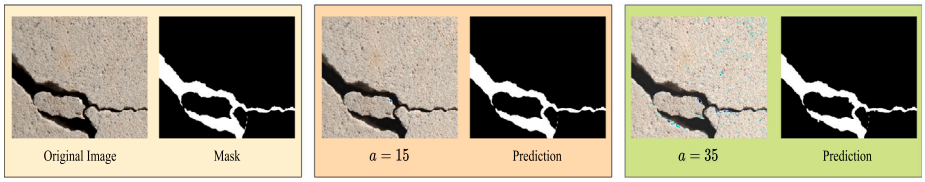
Fig. 9. IoU with four variations of the proposed model for changing (a) γ, (b) m, and (c) a, respectively.



(a) Values of $m = 1.0$, $a = 0$ while γ is with different values of γ of 0.2 and 2.4, respectively.



(b) Values of $\gamma = 1.0$, $a = 0$ while m is with different values of 0.5 and 1.5, respectively.



(c) Values of$\gamma = 1.0$, $m = 1.0$ while a is with different values of a of 15 and 35, respectively.

Fig. 10. Visualization of the effect of changing parameters γ, m, and a on the prediction results for an image of the DeepCrack dataset under different illumination conditions.

boxes highlighting segmentation gaps. LinkNet's performance is inconsistent, with both over-segmentation and under-segmentation present, as highlighted by the red boxes. SwinT and MobileVIT show under-segmentation, slightly missing some points and finer crack pixels, leading to incomplete crack detection. CT-CrackSeg, while better at detecting cracks, can incorrectly classify some non-crack pixels as cracks, resulting in false positives and missing finer crack details.

In all cases, the proposed CrackMaster model's segmentation masks are more accurate than other models. This visual analysis underscores the proposed model's superior capability in accurately detecting and segmenting cracks across diverse datasets, demonstrating its potential as a robust tool for real-world infrastructure monitoring and maintenance applications.

5.4. Robustness to illumination changes and noise

The proposed CrackMaster model and its variations (i.e, Baseline, Baseline+LinkNet, Baseline+LinkNet+EEF (CrackMaster without SSL) and CrackMaster) were tested with the RCD dataset (Okran et al., 2023) by changing the illumination of images and adding noise according to the formula (18), as visualized in Fig. 10. Despite changes in noise and illumination parameters, the prediction results remained stable, indicating the model's robustness.

$$I_o = \text{uint8}\left(255\left(\frac{mI_i + a}{255}\right)^{\gamma}\right), \tag{18}$$

where I_i and I_o are the input and output images, respectively. $m > 0$ is a multiplicative factor, a is an additive change factor, and $\gamma > 0$ is the gamma correction. The function *uint*8 quantifies the values to an eight-bit unsigned integer format so that they seem like normal image values without floating numbers.

As shown in Fig. 9, the inclusion of SSL in our proposed model demonstrates significant advantages in handling various noise conditions. When subjected to gamma noise (Fig. 9-(a)), all models show an initial increase in performance up to an optimal point ($\gamma \approx 1.0$) before performance declines. Fig. 10(a) further shows that despite variations in γ, the model's prediction results remain consistent, demonstrating resilience to illumination changes. The proposed CrackMaster model exhibits superior performance, peaking at an IoU score of ≈86.0 and maintaining higher scores across all γ values than other models. This indicates that SSL equips the model with better feature extraction capabilities, allowing it to adapt more effectively to varying noise and illumination in the input images.

In the case of multiplicative noise (Fig. 9-(b)), the IoU scores for the Baseline and Baseline+LinkNet models remain significantly lower than the proposed model. Fig. 10(b) illustrates that varying m has minimal effect on the prediction quality, highlighting the model's robustness against multiplicative noise. The Baseline+LinkNet+EFF model maintains a relatively higher IoU score, but the proposed CrackMaster model with SSL consistently outperforms all other tested variations. The CrackMaster model achieves an IoU score of 85 at the lowest noise

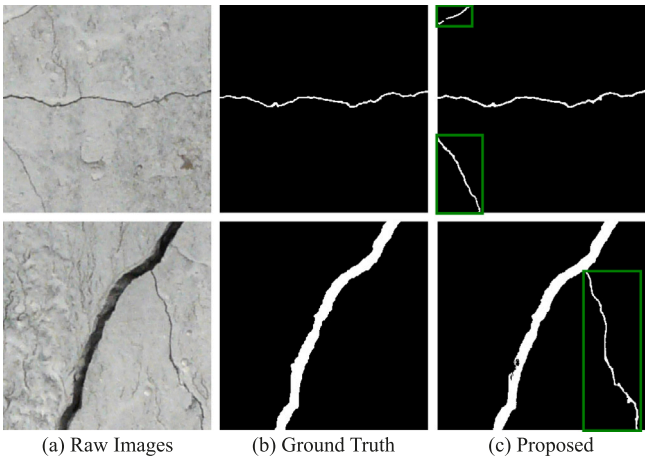


Fig. 11. Examples of detected cracks not present in the ground truth. It comprises three columns: Column (a) shows the raw images of the surfaces, column (b) displays the corresponding ground truth annotations, and column (c) presents the segmentation results from the proposed model, with correctly identified unannotated cracks highlighted by green boxes.

levels. It maintains robustness with a slight decline as noise increases, indicating its enhanced ability to generalize and resist noise disruption. In turn, regarding additive noise shown in (Fig. 9-(c)), the proposed model's performance is similarly superior, Fig. 10(c) shows that while a is adjusted, there is no significant change in the predicted results, suggesting robustness against additive noise. While all models experience a performance decline as noise intensity increases ($a > 1.0$), the proposed model consistently achieves higher IoU scores, degrading more gracefully than Baseline, Baseline+LinkNet, and Baseline+LinkNet+EFF models.

The consistent performance improvements observed with the proposed CrackMaster model justify its use, as shown in Fig. 10, where the visual results confirm the robustness under different noise and illumination conditions. SSL helps the model learn more robust feature representations, which is crucial for maintaining high performance under diverse and challenging conditions. This robustness is critical for real-world applications where environmental noise and lighting conditions vary unpredictably. By leveraging SSL, the proposed model achieves higher accuracy and ensures stability and reliability, making it a superior choice for practical implementations.

5.5. Identification of unannotated crack features

Fig. 11 sheds light on the proposed model's capability to detect crack features not marked in the ground truth annotations. These highlighted areas within the green boxes indicate the presence of actual cracks that are not included in the ground truth annotations. This may occur for various reasons, such as the subtle nature of early-stage cracks that can be overlooked during manual annotation or inconsistencies in the annotation process. Our model's ability to detect these features demonstrates its sensitivity and precision, transcending the limitations of the provided ground truth. This highlights the robustness of the proposed model in identifying true positives and underscores the potential necessity for re-evaluating the ground truth in specific datasets.

The inclusion of these results serves a dual purpose. It validates the model's effectiveness in a real-world scenario where all cracks may not be annotated and provides an opportunity to refine the datasets' annotations. Therefore, this qualitative analysis contributes to the ongoing development and improvement of automated crack detection systems and the datasets on which they are trained and evaluated.

5.6. Visual insights into the model interpretability

Fig. 12 offers a revealing window into the inner layers of the proposed model's crack detection.

In Fig. 12-c and -d, we explore the analytical insights into the proposed model's feature interpretation and segmentation decision-making process. The Feature Focus Heatmaps (column c), generated from the last encoder stage via GradCAM (Selvaraju et al., 2017), highlight the regions deemed crucial by the model for extracting meaningful features. The heatmaps' spectrum, ranging from cool blues to intense reds, visually represents the model's attention, emphasizing areas of highest activation crucial for subsequent segmentation tasks.

The Segmentation Decision Heatmaps (column d) offer a strategic perspective, revealing where the model makes its final segmentation decisions. This step is pivotal since the model consolidates its learned features to identify and segment cracks. Again, the heatmaps are colour-coded, with warmer colours indicating regions more substantially influencing the model's segmentation outputs. The alignment of these areas with the ground truth underscores the model's ability to identify true positives and its precision in segmenting cracks.

5.7. Trustworthy and reliability analysis

The results presented in Table 17 highlight the superior performance of the evaluated models across the RCD, DeepCrack537, and YCD datasets, emphasizing metrics such as Rank Graduation Box Accuracy (RGA) (Babaei et al., 2025) and Risk Probability Below Threshold (RPBT) (Singpurwalla, 2006), which gauge the trustworthiness and reliability of the models.

The proposed model achieved the highest reliability, with RGA scores of 81.41%, 88.65%, and 84.90% on the RCD, DeepCrack537, and YCD datasets, respectively. These scores indicate the model's ability to produce accurate and consistent predictions across diverse datasets. In turn, the second-best RGA scores were achieved by CT-CrackSeg for the DeepCrack537 and YCD datasets and SwinT for the RCD dataset.

Regarding RPBT, the proposed model consistently outperformed others, achieving the lowest risk probabilities at $t = 0.95$: 0.1941, 0.0000, and 0.0179 for the RCD, DeepCrack537, and YCD datasets, respectively. These outcomes highlight the robustness of our model under stringent confidence thresholds, affirming its reliability. Noteworthy is that while most models yielded perfect RPBT scores of 0.0000 at $t = 0.95$ on the DeepCrack537 dataset, SwinT and FPN displayed commendable performances in RPBT across varying thresholds for the RCD and YCD datasets, respectively.

Integrating metrics from the SAFE AI framework (Babaei et al., 2025) and Bayesian modelling principles (Singpurwalla, 2006) provided a holistic assessment of the trustworthiness of our proposed model. These metrics account for both prediction accuracy and uncertainty quantification, essential for deploying safe and reliable AI in high-stakes environments. thus, the dominant performance of our proposed model across both pivotal metrics reinforces its standing as the most reliable and trustworthy solution for crack segmentation tasks, particularly in safety-critical applications.

5.8. Complexity analysis

For the complexity analysis, we explore the comprehensive evaluation of CrackMaster beyond its precision in pixel-wise pavement crack detection. We used three key indicators: model parameters, inference time, and FLOPs, to achieve this. These metrics provide valuable insights into the computational demands of the model. Additionally, we conduct comparison experiments with other typical segmentation networks to provide context for our findings. Our results, summarized in Table 18, show that CrackMaster yields comparable FLOPs, inference time, and number of trained parameters to the tested models.

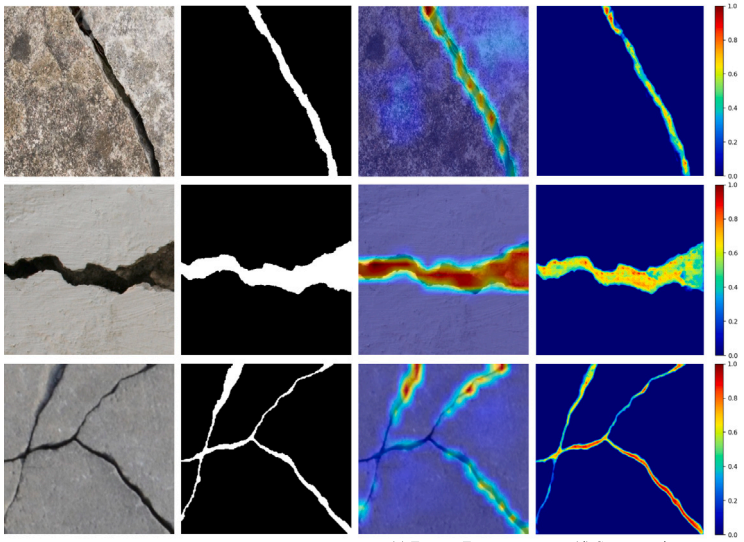


Fig. 12. Visualization of heatmaps for the proposed model. This figure is arranged into four distinct columns: (a) Raw Images, (b) Ground Truth, (c) Feature Focus Heatmaps of the Encoder, and (d) Segmentation Decision Heatmaps.

Table 17
Comparison of different models across RCD Dataset, DeepCrack537 Dataset, and YCD Dataset. Metrics include Rank Graduation Box Accuracy (RGA) and Risk Probability Below Threshold (RPBT) at various thresholds. The red bold text indicates the best results, while the blue bold text represents the second-best results.

Model	RCD Dataset				DeepCrack537 Dataset				YCD Dataset			
	RGA% ↑	RPBT% ↓			RGA% ↑	RPBT% ↓			RGA% ↑	RPBT% ↓		
		t = 0.85	t = 0.9	t = 0.95		t = 0.85	t = 0.9	t = 0.95		t = 0.85	t = 0.9	t = 0.95
U-Net	0.7323	0.0257	0.5531	0.8724	0.7137	0.0000	0.0000	0.0009	0.7089	0.0000	0.0014	0.1746
FPN	0.7800	0.0018	0.1851	0.5293	0.7786	0.0000	0.0000	0.0000	0.7073	0.0000	***0.0004***	***0.0874***
Link-Net	0.7841	0.0014	0.1621	0.4922	0.7741	0.0000	0.0000	0.0000	0.7225	0.0000	0.0009	0.1326
UPer-Net	0.7358	0.0301	0.5817	0.8874	0.7537	0.0000	0.0000	0.0000	0.7082	0.0000	0.0009	0.1326
SwinT	***0.7988***	***0.0004***	***0.0908***	***0.3514***	0.7116	0.0000	0.0000	0.0001	0.7140	0.0000	0.0008	0.1273
MobileVIT	0.7879	0.0012	0.1511	0.4733	0.6601	0.0000	0.0000	0.0007	0.6592	0.0000	0.0020	0.2067
CT-CrackSeg	0.7712	0.0039	0.2662	0.6386	***0.7951***	0.0000	0.0000	0.0000	***0.7350***	0.0000	0.0017	0.1904
Proposed	***0.8141***	***0.0001***	***0.0367***	***0.1941***	***0.8865***	0.0000	0.0000	0.0000	***0.8490***	0.0000	***0.0000***	***0.0179***

Table 18
Comparative analysis of FLOPs, inference time and the number of trained parameters of different segmentation methods on the DeepCrack537 test set with ConvNeXt-Base backbone for all methods.

Method	FLOPs	Inference time (s)	Parameters (M)
U-Net (Ronneberger et al., 2015)	92.37	0.0203	92.59
UPerNet (Xiao et al., 2018)	291.12	0.0356	120.74
FPN (Kirillov et al., 2017)	89.01	0.0204	89.62
LinkNet (Chaurasia and Culurciello, 2017)	85.30	0.0195	89.43
Proposed	97.91	0.0211	93.32

Despite the complexity of the model architecture, our model can manage the trade-off of increasing crack segmentation performance while maintaining a comparable inference time.

Particularly, the proposed model has 97.91 FLOPs, which is slightly higher than U-Net (92.37), FPN (89.01), and LinkNet (85.30) but significantly lower than UPerNet (291.12). This indicates that CrackMaster is more complex than some models but far more efficient than UPerNet. In terms of inference time, CrackMaster demonstrates an inference time of 0.0211 s, which is comparable to U-Net (0.0203 s), FPN (0.0204 s), and LinkNet (0.0195 s) and notably faster than UPerNet (0.0356 s). This shows that our model maintains efficient real-time performance despite its complexity. Additionally, with 93.32 million parameters, CrackMaster has a slightly higher parameter count than U-Net (92.59M), FPN (89.62M), and LinkNet (89.43M), but considerably fewer than UPerNet (120.74M). This suggests that our model strikes a balance between complexity and performance.

5.9. Analysis of failure results

In Fig. 13, the results of the proposed CrackMaster exhibit instances of inaccurate segmentation. For example, certain cases (highlighted with red rectangles) depict the model's failure to detect finer cracks, possibly due to sensitivity settings not finely tuned to capture subtle variations in pixel intensity that fine cracks represent. Additionally, some segments display over-segmentation, where non-crack areas are incorrectly identified as cracks. This is particularly noticeable in areas with complex textures or shadows that the model might misinterpret as cracks.

These discrepancies highlight areas for potential improvement in the model's performance. Enhancing feature extraction capabilities or adjusting the threshold value in the model's algorithm may be necessary to address the failure to detect fine cracks. Over-segmentation issues could be mitigated by refining the model's ability to differentiate between actual cracks and similar visual patterns, possibly

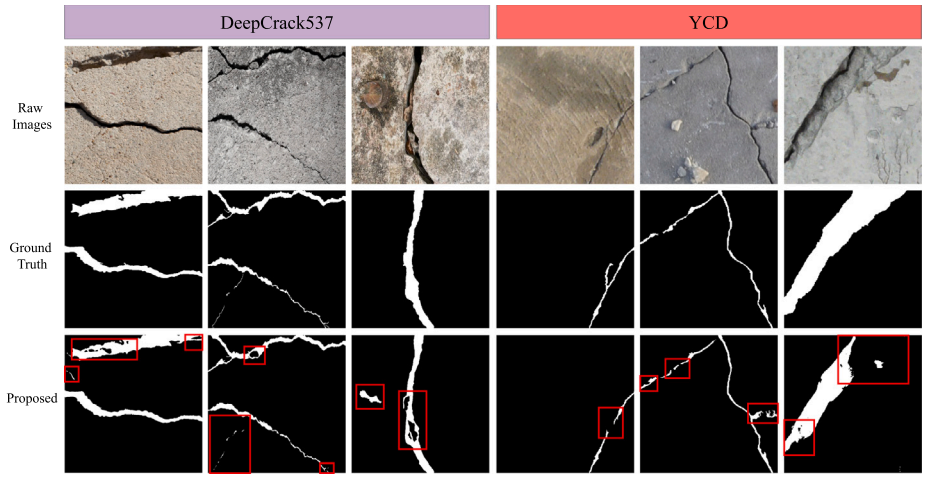


Fig. 13. Failure cases of our proposed model. The raw images show diverse surface conditions and crack types, ranging from distinct and broad to delicate and subtle cracks. The ground truth row accurately delineates these cracks, marking them in white against a black background, representing the ideal results expected from an effective segmentation model.

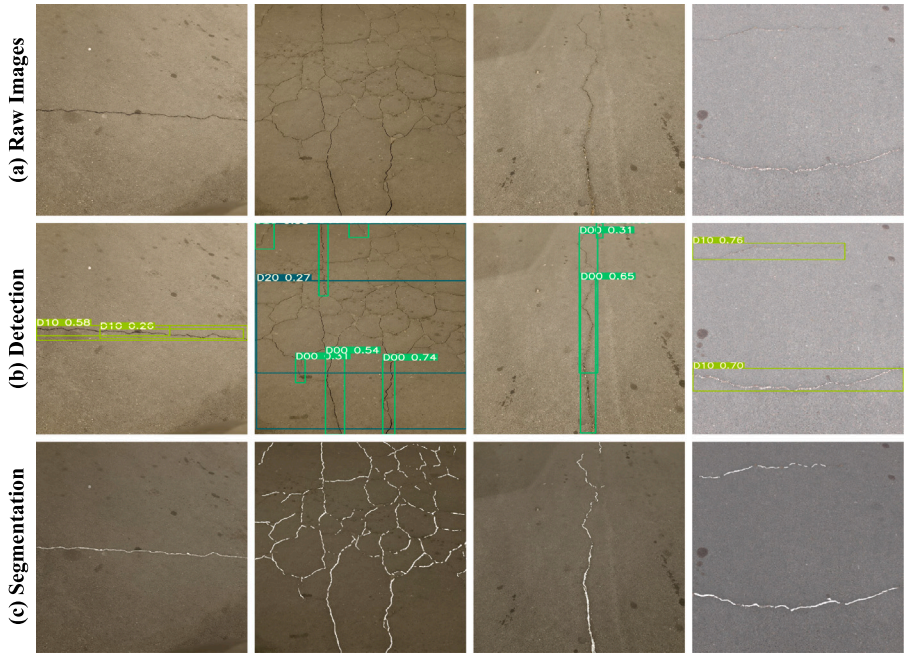


Fig. 14. Real-world case study results using the Tarragona City Road Crack Dataset. (a) Raw images of road surfaces showing various crack patterns. (b) The detection phase results in predicted bounding boxes (ROIs) indicating detected cracks. (c) Segmentation phase results are where the ROIs are processed to delineate the exact boundaries of the cracks, and the segmented results are merged into a single mask.

by adjusting the balance of precision and recall in the model's loss function. Additionally, training with augmented data or implementing techniques like spatial regularization could improve the model's ability to generalize from training data to real-world conditions, reducing false positives and enhancing crack detection accuracy. Furthermore, using high-resolution images could also help the model to detect the tiny crack regions.

5.10. Real-world case study

The efficacy of our proposed framework was validated through a pilot deployment using real-world images collected from Tarragona City, Spain. Some examples of the collected images are shown in Fig. 14-(a). Fig. 14 illustrates the results with the complete workflow of CrackMaster that comprises three stages: detection, cropping, and segmentation. In the first stage (Fig. 14-(b)), the detector predicts the bounding boxes representing the regions of interest (ROIs) containing cracks. As shown, the detection model can detect all cracks appearing in the examples. These ROIs are then cropped and resized to a fixed size to ensure uniform input dimensions for the subsequent stage. In the second stage, the cropped ROIs are fed into our proposed model, which accurately segments the cracks within these regions. Finally, the segmented results are merged into a single mask, comprehensively delineating all detected cracks (Fig. 14-(c)). This multi-step process effectively combines detection and segmentation to produce precise and reliable crack identification, demonstrating the robustness of our framework in real-world scenarios. The qualitative results shown in Fig. 14 confirm that our approach can handle diverse environmental conditions (e.g., day and night), ensuring high accuracy in detecting and segmenting road surface cracks. This proves its practical utility for urban infrastructure monitoring and maintenance.

6. Conclusion and future work

This work extensively investigates pixel-level crack detection using advanced segmentation techniques and introduces CrackMaster, a robust and effective framework for crack segmentation and detection. Through intensive experimentation on three diverse datasets – RCD, DeepCrack537, and YCD – CrackMaster demonstrated superior performance, consistently outperforming existing segmentation models. By incorporating the Enhanced Feature Fusion module, Self-Supervised Learning, and Skip Connections, the framework effectively addresses challenges such as class imbalance, feature loss, and environmental variability. Comparative analyses and qualitative visualizations, including GradCAM heatmaps, reinforced CrackMaster's robustness, adaptability, and real-world applicability, providing valuable insights into its decision-making process. These findings highlight the significance of advanced segmentation methodologies in tackling critical infrastructure maintenance challenges.

Ongoing work focuses on optimizing CrackMaster for real-time processing on edge devices or embedded systems that will enable on-site applications, critical for infrastructure monitoring and maintenance. Also, we will focus on implementing innovative regularization methods that could further improve the model's generalizability and reduce overfitting on specific datasets. Furthermore, future work concerns leveraging additional data sources such as LiDAR, infrared imaging, or hyperspectral data could enhance the framework's performance under extreme environmental conditions or on complex surfaces. In addition, we will explore methods to improve domain generalization, such as domain adaptation or transfer learning, which would extend CrackMaster's usability across diverse applications, including non-road surfaces and other structural materials. As well as, incorporating Bayesian approaches or confidence estimation methods could provide uncertainty quantification, enhancing reliability and interpretability in critical decision-making scenarios.

CRediT authorship contribution statement

Ammar M. Okran: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hatem A. Rashwan:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization. **Adel Saleh:** Writing – review & editing. **Domenec Puig:** Supervision.

Declaration of competing interest

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

Acknowledgements

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

Data availability

Data will be made available on request.

References

Al-Huda, Z., Peng, B., Algburi, R.N.A., Al-antari, M.A., Rabea, A.-J., Zhai, D., 2023. A hybrid deep learning pavement crack semantic segmentation. Eng. Appl. Artif. Intell. 122, 106142.

Amhaz, R., Chambon, S., Idier, J., Baltazart, V., 2016. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. IEEE Trans. Intell. Transp. Syst. 17 (10), 2718–2729.

Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Sekimoto, Y., 2022. Rdd2022: A multi-national image dataset for automatic road damage detection. arXiv preprint arXiv:2209.08538.

Ayenu-Prah, A., Attoh-Okine, N., 2008. Evaluating pavement cracks with bidimensional empirical mode decomposition. EURASIP J. Adv. Signal Process. 2008, 1–7.

Babaei, G., Giudici, P., Raffinetti, E., 2025. A rank graduation box for SAFE AI. Expert Syst. Appl. 259, 125239.

Bai, S., Ma, M., Yang, L., Liu, Y., 2024. Pixel-wise crack defect segmentation with dual-encoder fusion network. Constr. Build. Mater. 426, 136179.

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. 30 (7), 1145–1159.

Cano-Ortiz, S., Sainz-Ortiz, E., Iglesias, L.L., del Árbol, P.M.R., Castro-Fresno, D., 2024. Enhancing pavement crack segmentation via semantic diffusion synthesis model for strategic road assessment. Results Eng. 23, 102745.

Chaurasia, A., Culurciello, E., 2017. Linknet: Exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing. VCIP, IEEE, pp. 1–4.

Chen, H., Lin, H., 2021. An effective hybrid atrous convolutional network for pixel-level crack detection. IEEE Trans. Instrum. Meas. 70, 1–12.

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 801–818.

Contributors, M., 2020. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation.

Ding, W., Zhao, X., Zhu, B., Du, Y., Zhu, G., Yu, T., Li, L., Wang, J., 2022. An ensemble of one-stage and two-stage detectors approach for road damage detection. In: 2022 IEEE International Conference on Big Data (Big Data). IEEE, pp. 6395–6400.

Dung, C.V., et al., 2019. Autonomous concrete crack detection using deep fully convolutional neural network. Autom. Constr. 99, 52–58.

Fan, T., Wang, G., Li, Y., Wang, H., 2020. Ma-net: A multi-scale attention network for liver and tumor segmentation. IEEE Access 8, 179656–179665.

Fan, Z., Wu, Y., Lu, J., Li, W., 2018. Automatic pavement crack detection based on structured prediction with the convolutional neural network. arXiv preprint arXiv:1802.02208.

Guo, F., Liu, J., Lv, C., Yu, H., 2023. A novel transformer-based network with attention mechanism for automatic pavement crack detection. Constr. Build. Mater. 391, 131852.

Han, C., Ma, T., Huyan, J., Huang, X., Zhang, Y., 2021. CrackW-Net: A novel pavement crack image segmentation convolutional neural network. IEEE Trans. Intell. Transp. Syst. 23 (11), 22135–22144.

Han, C., Yang, H., Yang, Y., 2024. Enhancing pixel-level crack segmentation with visual mamba and convolutional networks. Autom. Constr. 168, 105770.

He, J., Wang, Y., Wang, Y., Li, R., Zhang, D., Zheng, Z., 2024. A lightweight road crack detection algorithm based on improved YOLOv7 model. Signal, Image Video Process. 1–14.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hendrycks, D., Gimpel, K., 2016. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141.

Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y.-W., Wu, J., 2020. Unet 3+: A full-scale connected unet for medical image segmentation. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 1055–1059.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pmlr, pp. 448–456.

Jeong, D., 2020. Road damage detection using YOLO with smartphone images. In: 2020 IEEE International Conference on Big Data (Big Data). IEEE, pp. 5559–5562.

Ju, X., Zhao, X., Qian, S., 2022. TransMF: Transformer-based multi-scale fusion model for crack detection. Mathematics 10 (13), 2354.

Kamaliardakani, M., Sun, L., Ardakani, M.K., 2016. Sealed-crack detection algorithm using heuristic thresholding approach. J. Comput. Civ. Eng. 30 (1), 04014110.

Kapela, R., Śniatała, P., Turkot, A., Rybarczyk, A., Pożarycki, A., Rydzewski, P., Wyczałek, M., Błoch, A., 2015. Asphalt surfaced pavement cracks detection based on histograms of oriented gradients. In: 2015 22nd International Conference Mixed Design of Integrated Circuits & Systems. MIXDES, IEEE, pp. 579–584.

Kheradmandi, N., Mehranfar, V., 2022. A critical review and comparative study on image segmentation-based techniques for pavement crack detection. Constr. Build. Mater. 321, 126162.

Kirillov, A., He, K., Girshick, R., Dollár, P., 2017. A unified architecture for instance and semantic segmentation.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25.

Li, H., Peng, T., Qiao, N., Guan, Z., Feng, X., Guo, P., Duan, T., Gong, J., 2024a. CrackTinyNet: A novel deep learning model specifically designed for superior performance in tiny road surface crack detection. IET Intell. Transp. Syst..

Li, Z., Xiao, X., Xie, J., Fan, Y., Wang, W., Chen, G., Zhang, L., Wang, T., 2024b. Cycle-YOLO: A efficient and robust framework for pavement damage detection. arXiv preprint arXiv:2405.17905.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, pp. 21–37.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: 2021 IEEE/CVF International Conference on Computer Vision. ICCV, IEEE Computer Society, Los Alamitos, CA, USA, pp. 9992–10002. http://dx.doi.org/10.1109/ICCV48922.2021.00986.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022. A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986.

Liu, H., Yang, J., Miao, X., Mertz, C., Kong, H., 2023. Crackformer network for pavement crack segmentation. IEEE Trans. Intell. Transp. Syst. 24 (9), 9240–9252.

Liu, Y., Yao, J., Lu, X., Xie, R., Li, L., 2019. DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing 338, 139–153.

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Lu, X., Li, Q., Li, J., Zhang, L., 2025. Deep learning-based method for detection and feature quantification of microscopic cracks on the surface of concrete dams. Measurement 240, 115587.

Ma, M., Yang, L., Liu, Y., Yu, H., 2024a. An attention-based progressive fusion network for pixelwise pavement crack detection. Measurement 226, 114159.

Ma, M., Yang, L., Liu, Y., Yu, H., 2024b. A transformer-based network with feature complementary fusion for crack defect detection. IEEE Trans. Intell. Transp. Syst..

Maeda, H., Kashiyama, T., Sekimoto, Y., Seto, T., Omata, H., 2021. Generative adversarial network for road damage detection. Comput.- Aided Civ. Infrastruct. Eng. 36 (1), 47–60.

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H., 2018. Road damage detection and classification using deep neural networks with smartphone images. Comput.- Aided Civ. Infrastruct. Eng. 33 (12), 1127–1141.

Meftah, I., Hu, J., Asham, M.A., Meftah, A., Zhen, L., Wu, R., 2024. Visual detection of road cracks for autonomous vehicles based on deep learning. Sensors 24 (5), 1647.

Mehta, S., Rastegari, M., 2021. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178.

Munawar, H.S., Hammad, A.W., Waller, S.T., Islam, M.R., 2022. Modern crack detection for bridge infrastructure maintenance using machine learning. Human- Centric Intell. Syst. 2 (3), 95–112.

Nguyen, T.S., Begot, S., Duculty, F., Avila, M., 2011. Free-form anisotropy: A new method for crack detection on pavement surface images. In: 2011 18th IEEE International Conference on Image Processing. IEEE, pp. 1069–1072.

Ni, F., Zhang, J., Chen, Z., 2019. Pixel-level crack delineation in images with convolutional feature fusion. Struct. Control. Heal. Monit. 26 (1), e2286.

Okran, A.M., Abdel-Nasser, M., Rashwan, H.A., Puig, D., 2022a. A curated dataset for crack image analysis: Experimental verification and future perspectives. In: Artificial Intelligence Research and Development. IOS Press 225–228.

Okran, A.M., Abdel-Nasser, M., Rashwan, H.A., Puig, D., 2022b. Effective deep learning-based ensemble model for road crack detection. In: 2022 IEEE International Conference on Big Data (Big Data). IEEE, pp. 6407–6415.

Okran, A.M., Rashwan, H.A., Puig, D., 2024. Enhanced crack segmentation network: Leveraging multi-dimensional attention. In: Artificial Intelligence Research and Development. IOS Press, pp. 94–96.

Okran, A.M., Saleh, A., Puig, D., Rashwan, H.A., 2023. Stacking up for success: A cascade network model for efficient road crack segmentation. In: Artificial Intelligence Research and Development. IOS Press, pp. 38–47.

Oliveira, H., Correia, P.L., 2012. Automatic road crack detection and characterization. IEEE Trans. Intell. Transp. Syst. 14 (1), 155–168.

Peng, L., Chao, W., Shuangmiao, L., Baocai, F., 2015. Research on crack detection method of airport runway based on twice-threshold segmentation. In: 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control. IMCCC, IEEE, pp. 1716–1720.

Qu, Z., Lin, L.-D., Guo, Y., Wang, N., 2015. An improved algorithm for image crack detection based on percolation model. IEEJ Trans. Electr. Electron. Eng. 10 (2), 214–221.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 28.

Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L., Shen, X., 2020. Image-based concrete crack detection in tunnels using deep fully convolutional networks. Constr. Build. Mater. 234, 117367.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597.

Salman, M., Mathavan, S., Kamal, K., Rahman, M., 2013. Pavement crack detection using the gabor filter. In: 16th International IEEE Conference on Intelligent Transportation Systems. ITSC 2013, IEEE, pp. 2039–2044.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 618–626.

Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z., 2016. Automatic road crack detection using random structured forests. IEEE Trans. Intell. Transp. Syst. 17 (12), 3434–3445.

Simon, S., Schwarz, G.M., Aichmair, A., Frank, B.J., Hummer, A., DiFranco, M.D., Dominkus, M., Hofstaetter, J.G., 2022. Fully automated deep learning for knee alignment assessment in lower extremity radiographs: A cross-sectional diagnostic study. Skelet. Radiol. 1–11.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Singpurwalla, N.D., 2006. Reliability and Risk: A Bayesian Perspective. John Wiley & Sons.

Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114.

Tao, H., 2024. Weakly-supervised pavement surface crack segmentation based on dual separation and domain generalization. IEEE Trans. Intell. Transp. Syst..

Tao, H., Liu, B., Cui, J., Zhang, H., 2023. A convolutional-transformer network for crack segmentation with boundary awareness. In: 2023 IEEE International Conference on Image Processing. ICIP, IEEE, pp. 86–90.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H., 2021. Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 32–42.

Wang, Y.J., Ding, M., Kan, S., Zhang, S., Lu, C., 2018a. Deep proposal and detection networks for road damage detection and classification. In: 2018 IEEE International Conference on Big Data (Big Data). IEEE, pp. 5224–5227.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B., 2019. Deep high-resolution representation learning for visual recognition. TPAMI.

Wang, W., Wu, B., Yang, S., Wang, Z., 2018b. Road damage detection and classification with faster R-CNN. In: 2018 IEEE International Conference on Big Data (Big Data). IEEE, pp. 5220–5223.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J., 2018. Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 418–434.

Yang, L., Bai, S., Liu, Y., Yu, H., 2023. Multi-scale triple-attention network for pixelwise crack segmentation. Autom. Constr. 150, 104853.

Yang, L., Fan, J., Huo, B., Li, E., Liu, Y., 2022. A nondestructive automatic defect detection method with pixelwise segmentation. Knowl.-Based Syst. 242, 108338.

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., Yang, X., 2018. Automatic pixel-level crack detection and measurement using fully convolutional network. Comput. Aided Civ. Infrastruct. Eng. 33 (12), 1090–1109.

Yong, H., Chun-Xia, Z., 2010. A local binary pattern based methods for pavement crack detection. J. Pattern Recognit. Res. 5 (1), 140–147.

Zhang, E., Shao, L., Wang, Y., 2023. Unifying transformer and convolution for dam crack detection. Autom. Constr. 147, 104712.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2881–2890.

Zhou, J., Huang, P.S., Chiang, F.-P., 2006. Wavelet-based pavement distress detection and evaluation. Opt. Eng., Bellingham 45 (2), 027007–027007.

Zhou, Q., Qu, Z., Li, Y.-X., Ju, F.-R., 2022a. Tunnel crack detection with linear seam based on mixed attention and multiscale feature fusion. IEEE Trans. Instrum. Meas. 71, 1–11.

Zhou, Q., Qu, Z., Wang, S.-Y., Bao, K.-H., 2022b. A method of potentially promising network for crack detection with enhanced convolution and dynamic feature fusion. IEEE Trans. Intell. Transp. Syst. 23 (10), 18736–18745.

Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J., 2018. Unet++: A nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4. Springer, pp. 3–11.