DOI: 10.1111/mice.12577

ORIGINAL ARTICLE

A Rendering-based Lightweight Network for Segmentation of High-resolution Crack Images

Honghu Chu[1, 2] | Diran Yu[2] | Weiwei Chen[2, *] | Jun Ma[3] | Lu Deng[1]

[1] College of Civil Engineering, Hunan University, Changsha, China

[2] Bartlett School of Sustainable Construction, University College London, London, UK

3 Faculty of Architecture, The University of Hong Kong, Hong Kong, China

Correspondence

Weiwei Chen, Bartlett School of Sustainable Construction, University College London, London, WC1E 7HB, UK.

Email: weiwei.chen@ucl.ac.uk

Funding information

National Natural Science Foundation of China, Grant/Award Number: 52278177; China Scholarship Council, Grant/Award Number: 202206130068; Hunan Provincial Innovation Foundation for Postgraduate, Grant/Award Number: QL20210106

ABSTRACT

High-resolution (HR) crack images provide detailed structural assessments crucial for maintenance planning. However, the discrete nature of feature extraction in mainstream deep learning algorithms and computational limitations hinder refined segmentation. This study introduces a rendering-based lightweight crack segmentation network (RLCSN) designed to efficiently predict refined masks for HR crack images. The RLCSN combines a deep semantic feature extraction architecture—merging Transformer with a super-resolution boundary-guided branch—to reduce environmental noise and preserve crack edge details. It also incorporates customized point-wise refined rendering for training and inference, focusing computational resources on critical areas, and an efficient sparse training method to ensure efficient inference on commercial mobile computing platforms. Each RLCSN's components are validated through ablation studies and field tests, demonstrating its capability to enable unmanned aerial vehicle (UAV)-based inspections to detect cracks as narrow as 0.15 mm from a distance of 3 meters, thereby enhancing inspection safety and efficiency.

1 INTRODUCTION

Cracks are key indicators in mechanical performance tests of concrete and are crucial for routine bridge inspections (Deng et al., 2022; Tian et al., 2022). For concrete bridge structures, the presence of cracks can lead to the spalling of protective layers and corrosion of steel reinforcements. These degradations not only diminish the structure's durability but also directly compromise its strength and stability. Furthermore, severe through-cracks may even lead to significant structural damage and greatly endanger the structure's safety (Chu et al., 2023). Accurate crack detection results can intuitively reflect the damage level of concrete structures and further reveal the bridge structure's mechanism of force, which holds substantial importance for maintenance staff in safeguarding the structural integrity of bridges throughout their operational lifespan (Chun et al., 2021).

Conventional methods for detecting cracks in bridges have largely relied on manual inspections, which is not only inefficient and imprecise but also subject to variability due to the inspectors' expertise and the quality of inspection equipment, resulting in inconsistent detection outcomes (Ellenberg et al., 2016; Jeong et al., 2020). Moreover, the effectiveness of human visual inspection is significantly impacted by lighting conditions and is incapable of assessing areas like bridge towers and tall piers, thereby posing challenges in the representation of complete and objective crack information (Yeum and Dyke, 2015; Abdallah et al., 2022). These drawbacks make current manual inspections unable to meet the requirements of numerous bridge crack detections in terms of economy, efficiency, accuracy, and data management (Sacks et al., 2018).

In recent years, segmentation networks based on encoder-decoder architecture have made significant progress and are expected to become the paradigm for high-precision intelligent crack detection. However, as shown in Figure 1, most typical encoder-decoder architectures such as FCN (Long et al., 2015) and UNet (Ronneberger et al., 2015) still face a major limitation in segmenting cracks from HR images: the segmentation results show ambiguous boundary predictions.

© 2020 Computer-Aided Civil and Infrastructure Engineering

The underlying reason is that the grayscale values of pixels at the crack edges are typically similar to those of adjacent pixels, which, coupled with the segmentation algorithms' inadequate emphasis on boundary area pixels, frequently results in inaccurate automated segmentation. Imprecise segmentation in boundary areas significantly impedes the application of automatic segmentation algorithms in bridge inspections. Given that crack width is an essential metric for quantitative analysis and is distinctly defined as a vital indicator within detection protocols, the foundational requirement for extracting accurate crack width data lies in the boundary precision of the crack segmentation mask (Alipour et al., 2019; Ni et al., 2019). To address the issue of blurred boundaries, researchers have dedicated efforts to devising numerous methods that concentrate specifically on boundary information, such as manually adding parameters for post-processing (Mohan and Poobal, 2018), and adding boundary constraints in the network (Takikawa et al., 2019; Lee et al., 2020).

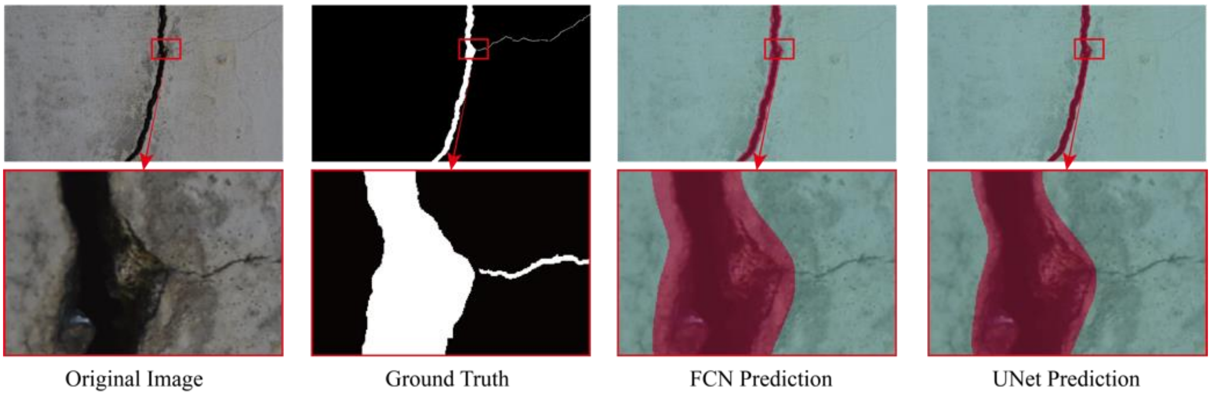

FIGURE 1 Coarse Segmentation Performance Exhibited by the FCN and Unet Architectures at the Crack Boundaries when Dealing with High-resolution (HR) Crack Images.

Crack segmentation presents a long-standing challenge in the dense prediction tasks of computer vision. Although numerous studies have been conducted to enhance crack segmentation quality (Liang et al., 2020; Yuan et al., 2020; Li et al., 2022), the challenges of refined crack boundary prediction remain inadequately addressed.

In fact, the issue of unclear boundary areas in crack image segmentation parallels the issue of aliasing artifacts that arise in the realm of computer graphics when images or graphics become pixelated on display devices. Researchers in the field of computer graphics chiefly concentrate on developing rendering technologies for achieving edge anti-aliasing, extensively applied within the realms of gaming and film special effects (Cole et al., 2021; Hu et al., 2023). These efforts, to a significant degree, symbolize the zenith of technological advancement in the field of computer vision. Thus, the authors seek inspiration from the rendering-based anti-aliasing technique for performing segmentation with refined and unabridged edges for HR crack images. The authors noted that in computer graphics, the recently proposed edge-aware rendering is an effective anti-aliasing technique (Barron et al., 2021; Chen et al., 2023), primarily concentrating on dense computation in the edge areas of the entire image, aiming to reduce aliasing, blurring, and other artifacts in the target edge areas. Based on importance sampling and dynamic resolution adjustment techniques, the edge-aware rendering can even ensure output at 4K resolution, simulating the refraction and reflection of light at the edges of transparent objects in real time.

Therefore, the authors aim to introduce this advanced edge-aware rendering technique from computer graphics into the task of refined segmentation of HR crack images. However, crack semantic segmentation focuses on the semantic understanding of crack pixels, while image rendering emphasizes generating realistic 2D images from the entire 3D scene or 3D model, posing fundamental differences in the objectives of the two tasks. Therefore, directly applying image rendering, which belongs to the image pre-processing method, to the post-processing crack segmentation task poses several challenges:

1. Firstly, unlike most objects in natural scenes, cracks are found in environments and backgrounds characterized by high levels of non-uniformity. These complex backgrounds, along with various environmental noises, substantially impede fine edge segmentation.

2. Secondly, segmentation tasks have more specific objectives compared to image rendering tasks, in that they focus solely on segmenting crack pixels, rather than all pixels as in rendering tasks. Indiscriminately incorporating a rendering head into the segmentation architecture would burden the model with unnecessary computational resources, resulting in computational redundancy.

3. Finally, incorporating rendering methods might significantly increase the model's complexity and computational cost. Because rendering methods in computer graphics usually involve complex lighting and shadow computations and need to consider material and texture information, which would increase the parameter count and operations in deep learning (DL) models. Therefore, considering the practicality and widespread applicability, it is necessary to design efficient model structures and training strategies.

Thus, a specialized algorithm based on rendering representation is required for accurate boundary segmentation of crack images.

To this end, this study proposes a rendering-based lightweight crack segmentation network (RLCSN), inspired by the refined rendering graphic representation architecture in computer graphics. It is noteworthy that the RLCSN is the first network that applies the emerging refined representation theory from computer graphics – rendering technology – to the segmentation of high-resolution (HR) crack images, an area that has not been fully explored. It specifically addresses the three issues described above with three targeted improvements on the basis of the high-precision rendering head, allowing the advantages of rendering methods to be fully utilized in accurate segmentation of crack images. The network architecture is shown in Figure 2.

The first innovation is the design of a crack deep semantic feature extraction architecture that combines Transformer with a super-resolution boundary-guided branch, reducing environmental noise interference while effectively preserving crack edge details. Second, two types of point-wise refined rendering point sampling methods were customized for training and inference stages, enabling effective concentration of computational resources on ambiguous crack edges and tiny crack areas. Thirdly, for the initially built RLCSN, an efficient sparse training method was designed, incorporating an L1 norm of weights into the loss function and performing pruning at the weight level, achieving lightweight deployment of the model. With these customizations, the refined scene rendering methods originally used in computer graphics can be effectively applied to the fine segmentation of crack images. This customized adaptation maintains the high precision and GPU-friendly attributes characteristic of rendering representation techniques.

The main contributions of this paper are as follows:

1. An HR crack image fine segmentation architecture named RLCSN is proposed. As the first network architecture to employ rendering technology for crack segmentation, the RLCSN achieves fine segmentation of crack images with dispersed topological structure distribution on a low-cost GPU.
2. When integrated with the unmanned aerial vehicle (UAV) equipped with the HR imaging device, the RLCSN offers a safer and more efficient method for the crack detection of real bridges.
3. Through comprehensive ablation studies, visualization, and comparative analysis, this paper thoroughly investigates the operational mechanism of the RLCSN. It demonstrates the network's exceptional performance, attributable to its computational approach of executing predictions for crack edges in a dense representation fashion, facilitated by boundary point rendering technology.

2 LITERATURE REVIEW

2.1 DL-based Crack Detection

As summarized above, semantic segmentation algorithms can extract crack features more precisely. However, most of the semantic segmentation algorithms used in existing crack recognition research are based on Convolutional Neural Network (CNN) architecture (Bang et al., 2019; Zou et al., 2019). The essence of feature extraction in CNNs lies in the convolution kernels, which capture local spatial information accurately due to their translational invariance and local sensitivity (Li et al., 2022; She et al., 2023). Nevertheless, convolution kernels lack a global understanding of the image, making it difficult to establish dependencies between features, resulting in challenges for CNN-based segmentation architectures to maintain the integrity of global crack segmentation (Ni et al., 2019; Alzubaidi et al., 2021). Following this, Vaswani et al. (2017) introduced self-attention, achieving notable results in overcoming the dependency features of long-distance words in machine translation tasks, which garnered widespread attention from computer vision researchers. On this basis, Dosovitskiy et al. (2020) proposed the Vision Transformer (VIT), applying self-attention to image classification tasks for the first time. Tests on multiple large-scale open-source datasets (such as ImageNet and Cifar-100) demonstrated the effectiveness and potential of self-attention in capturing global features for image processing. Subsequently, Liu et al. (2021) introduced Swin Transformer based on sliding window self-attention, achieving better results in semantic segmentation tasks than most CNNs while significantly reducing the model parameters. Recently, network architectures composed of concatenated Transformer blocks have been applied to crack recognition tasks.

Zhou et al. (2023) , building upon the DeepLabv3+ encoder-decoder architecture, introduced Swin Transformer and CNN inverse residual blocks, enhancing the model's capability to capture global information of cracks while preserving the performance of local detail representation. Xiang et al. (2023) proposed a dual-encoder network integrating transformer and CNN, improving segmentation results of crack pixels in complex backgrounds and demonstrating robustness across multiple open-source datasets while maintaining high inference speed. To effectively address the challenge of incomplete segmentation results for slender cracks in complex backgrounds, Guo et al. (2023) proposed an encoder-decoder architecture based on Swin Transformer and UperNet, enhancing the segmentation result's completeness by learning global and remote semantic features of crack pixels. Quan et al. (2023) introduced a crack pixel-level segmentation architecture incorporating ViT, leveraging ViT's captured global context information to establish global dependencies for dispersed crack pixels, thus significantly improving segmentation accuracy. However, the prediction heads at the end of the decoders in the above methods treat edge and main body pixels equally, resulting in insufficient computational resource allocation in the hard sample areas (i.e., crack edge pixels) and leading to ambiguous prediction results at the edges of the predicted masks.

2.2 Boundary-aware Semantic Segmentation and Lightweight Models

Shen et al. (2022) and Cheng (2020) observed that the deep receptive fields and downsampling processes in traditional DL architectures tend to smooth out sharp edges in feature spaces (Rafiei et al., 2017; Hassanpour et al., 2019; Martins et al., 2020), making segmentation predictions more errors at the boundaries. To address these issues, one of the most common methods is to combine boundary prediction with segmentation tasks for multi-task training. Yu et al. (2018) designed a segmentation network called DFNet, which features a boundary prediction branch. This branch enhances supervision for binary boundaries, thereby correcting ambiguous boundary predictions. Xu et al. (2018) proposed PAD-Net, which includes a contour detection branch. This multi-task network improves depth prediction and scene parsing performance through multimodal information sharing. Additionally, some works have attempted to utilize the predicted boundary masks in the forward process, rather than simply adding boundary detection as an auxiliary task. Takikawa et al. (2019) proposed a dual-stream network, Gated-SCNN, which uses a shape stream to extract boundary semantic information for each target independently, combining the extracted features with the output of the regular segmentation stream for prediction, to achieve refined representation of target boundary areas. Li et al. (2020) proposed DepairSegNets, which include a feature distortion sub-network branch capable of independently decoupling edge features for enhanced representation of target boundary details. Ding et al. (2020) introduced CGBNet, which utilizes a Boundary Deteched Result (BDR) module to recover lost boundary information. The BDR module effectively suppresses low-level features far from boundaries while enhancing details in high Signal-to-Noise Ratio boundary regions. However, these works always classify boundary and interior pixels as two distinct classes when optimizing the auxiliary pixel boundary classification task. Since boundary pixels are forcibly distinguished from interior pixels, intra-class consistency is disrupted. As different class boundary pixels are classified into the same boundary class, these methods also reduce inter-class differences, particularly at the boundaries.

There are also works based on pairwise affinity or graphs to improve boundary segmentation quality. Ding et al. (2019) proposed an affinity-based boundary-aware feature propagation module, which uses a directed acyclic graph to propagate semantic information within boundaries and maintain object-internal consistency. Bertasius et al. (2017) introduced the Random Walk Network to jointly optimize pairwise affinities, capturing semantic relations between objects through a random walk layer to enhance boundary performance in segmentation results. Chen et al. (2021) adopted GALD, which employs a local affinity matrix with global priors in an adaptive manner for reinforced representation of boundary information. Building on this, Ke et al. (2018) proposed the concept of Adaptive Affinity Field to capture and match relationships between adjacent pixels in semantic label space, using adversarial loss for reinforced learning of boundaries. Borse et al. (2021), based on the homography transformation associating boundary hypotheses, proposed InverseForm to refine target boundaries by measuring and supervising the similarity between predicted and ground-truth boundaries. However, as these types of methods treat all homogeneous pixels equally and do not select, they not only propagate discriminative information but also noise.

Other works have improved boundary segmentation results through post-processing methods. Krähenbühl and Koltun (2011) introduced DenseCRF, which refines coarse segmentation results' boundaries through Conditional Random Fields. Bertasius et al. (2016) proposed the Boundary Neural Field, globally optimizing segmentation results based on edge maps and boundary-based pixel affinity functions, which assign low similarity between pixels separated by strong boundaries. Replacing BNF's boundary map with the output edge map from CED (Wang et al., 2018) further improves BNF's refined segmentation results.

Recently, Li et al. (2022) adopted an additional boundary-aware branch to enhance mask feature boundary perception, which could, to some extent fix optimization biases, but the increased computational cost of the boundary branch restricted its implementation to low-resolution images. Liu et al. (2023) proposed a Dual-Stream Boundary-Aware Crack segmentation network, achieving fine recognition of edges through dynamic feature fusion, but it was also limited by computational resources and could only conduct inference on low-resolution images. Additionally, the authors noted that in the field of computer vision, PolyTransform (Liang, Homayounfar, Ma, Xiong, Hu and Urtasun, 2020) and SegFix (Yuan, Xie, Chen and Wang, 2020) could serve as post-processing solutions to improve the quality of crack boundary segmentation. Specifically, Liang et al. (2020) developed PolyTransform, which uses a deformation network with clipped instance patches to predict polygon vertex offsets, but it incurs a high computational cost. Yuan et al. (2020) developed SegFix, which uses a deformation network with clipped instance patches to predict polygon vertex offsets. However, the aforementioned post-processing methods cannot be used to conduct end-to-end inference. In contrast, end-to-end approaches provide a streamlined process that inherently learns to correct errors, which can be advantageous for achieving higher accuracy in boundary segmentation tasks.

In addition, several studies have addressed the computational efficiency and deployment constraints of DL models by implementing lightweight architectures for crack segmentation. For instance, a streamlined network named MiniCrack has been developed for detecting narrow cracks under resource-limited conditions, utilizing PixelShuffle and PixelUnshuffle to counteract the drawbacks of pooling (Lan and Dong, 2022). Additionally, Kim et al. (2021) introduced a hierarchical CNN-based approach that reduces inference time by 65.90% compared to traditional segmentation networks. Reference (Xie et al., 2022) presents a crack segmentation model that combines sparse sensing encoders and superpixel decoders, surpassing other models in accuracy and efficiency. Wang and Su (2021) devised a lightweight crack segmentation model using a bilateral segmented network and a contextual path for rapid downsampling of feature maps. Chen et al. (2023) managed the knowledge distillation process with a temperature parameter, introducing high-temperature and non-isothermal distillation strategies for efficient training of lightweight models. However, their method's requirement for strict alignment between the teacher and student networks imposes constraints that are overly restrictive for semantic segmentation tasks.

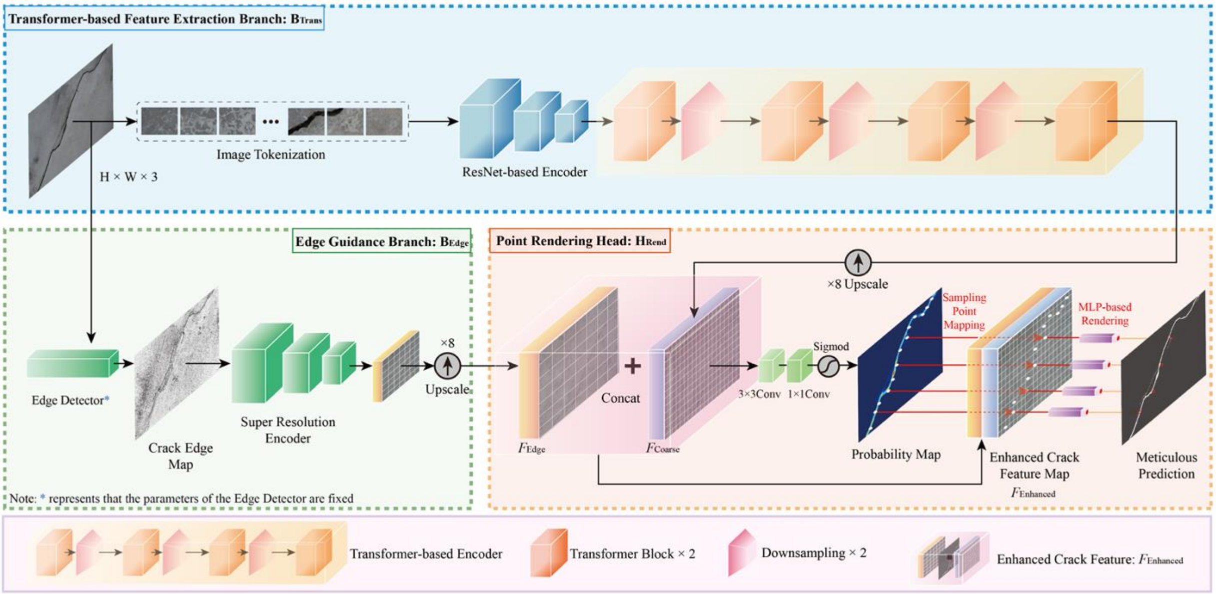

FIGURE 2. Implementation Process of the Rendering-based Lightweight Crack Segmentation Network (RLCSN)

3 Rendering-based Lightweight Crack Segmentation Network

The RLCSN proposed in this study consists of three main components: the main backbone for coarse crack feature extraction B_{Trans}, the boundary-guided branch B_{Edge}, and the refined rendering head H_{Rend}. B_{Trans} is composed of an encoder architecture built on Transformer, B_{Edge} is based on a fixed-parameter edge detector and a super-resolution encoding structure, and H_{Rend} is constructed as a lightweight Multi-Layer Perceptron (MLP) based on point-wise refined rendering. The deep coarse semantic features F_{Coarse} captured by B_{Trans} are concatenated with the fine semantic features of crack boundaries F_{Edge} outputted by B_{Edge} to form an enhanced crack feature map F_{enhanced}. F_{enhanced} serves as the information source for H_{Rend} to execute refined rendering, and refined crack mask prediction is carried out based on the specialized rendering points sampling method. Figure 2 visually presents certain algorithmic details and computational logic of the proposed RLCSN.

3.1 Crack Feature Encoding Architecture

3.1.1 Coarse Crack Feature Extraction Backbone Based on Transformer

The coarse crack feature extraction backbone based on Transformer comprises a Patch embedding layer and an encoder. Initially, the Patch embedding layer divides the input crack image into non-overlapping image blocks with a resolution of 4×4 pixels, and then a convolutional layer maps the channel number of each image block to a specified dimension. The encoder sends the divided image block sequence into stacked Transformer modules to learn contextual feature representations, merging image blocks through downsampling layers to reduce the resolution of the feature map and increase the channel dimension. Since multiple Transformer encodings and downsampling are required, the encoder learns multi-scale hierarchical feature representations, which will help the network to enhance the representation of tiny crack details.

Specifically, the Patch embedding layer's function is to divide the input HR crack image $x \in \mathbb{R}^{H \times W \times 3}$ into a set of non-overlapping image block sequences $x_p \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times (3P^2)}$. Here, P represents the size of the image block. It is evident that there are $\frac{H}{P} \times \frac{W}{P}$ image blocks, and each image block, when unfolded, has a channel number of $3P^2$, which after a further linear transformation, maps to the specified dimension C, as $x_p \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times C}$. In practice, $P = 4$ is chosen, and a convolution operation is performed on the input image using a convolution kernel of size 4×4, stride 4, and output channel C. Through this process, the generated image blocks, combined together, form an initial feature map with a resolution of 1/4 of the original input and a channel dimension of C.

The Transformer module consists of Layer Normalization (LN), Multi-Head Self-Attention (MSA), residual connections, and Multi-Layer Perceptron (MLP). In MSA, to reduce computational load, the study adopts the sliding window method of Swin Transformer to perform self-attention calculations within non-overlapping local windows of each block. The details of Swin Transformer can be found in the reference (Liu et al., 2021). It should be noted that each self-attention head models attention from different representational spaces, enabling accurate extraction of richer crack semantic information from crack images with complex backgrounds. After the multi-head self-attention computation, the block sequence features maintain their original scale and are input into the refined rendering head H_{Rend} after 8× upsampling.

3.1.2 Boundary-Guided Branch

To ensure that the point-wise refined rendering head, built on shared-weight MLP, is effectively guided to the crack edge areas, it is essential to preserve refined boundary details within the deep semantic features extracted by the encoding architecture. Therefore, the Boundary-Guided Branch B_{Edge} is proposed.

During the boundary guidance process, a morphological edge detector built on the basis of the Sobel operator with fixed parameters is initially used as an edge-guided encoder to preliminarily extract crack contours from the original HR crack image (Nhat-Duc et al., 2018). The extracted binary crack contour image, along with the original crack image, is then input into a custom-designed super-resolution encoding architecture based on residual attention. This architecture refines the encoding of crack edge features, enabling them to guide the implicit function in performing refined pixel restoration in subsequent lower-dimensional latent spaces.

The super-resolution model based on residual attention, serving as the core architecture for achieving the above objectives, primarily consists of two parts: a shallow feature extraction module and a deep feature extraction module. The shallow feature extraction module uses 3×3 convolution to extract 64-channel features from the concatenated RGB crack image and edge binary image, followed by nonlinear rectification through a PReLU activation function. The deep feature extraction module is composed of several concatenated residual attention modules, with skip connections between the outputs of each module to facilitate more effective information transfer. The computational process is represented by the following equations:

$$F_1 = F_L + H_1(F_L) \tag{1}$$

$$F_2 = F_1 + H_2\big(H_1(F_L)\big) \tag{2}$$

Where $H_1(F_L)$ is the output of the first residual attention module applied to the shallow features F_L, which is then added to F_L as input F_1 for the second residual attention module. Similarly, the output of the second residual attention block is $H_2\big(H_1(F_L)\big)$, where the input to each residual attention module is the sum of the input and output of the previous module. This process continues until the output of the last layer of the deep feature extraction module, F_D, is obtained.

It is important to note that, to improve the transmission efficiency of tiny crack feature information in HR images within the network, the authors customized each residual attention module. Firstly, the original residual module was replaced with a lightweight WDSR-B residual module, which, while sharing network weights, increased the number of feature channels before activation and the utilization of information in the network, effectively mitigating the feature transmission barrier caused by the ReLU activation function in the original residual module. Additionally, the feature extraction network in the WDSR-B residual module uses 1×1 convolution instead of 3×3 convolution, significantly reducing computational costs. When performing operations in the residual attention module, the input feature map first undergoes 1×1 convolution with 256-channel dimensions, allowing more high-frequency features containing tiny crack details to be extracted. Then, after nonlinear rectification, it is followed by 1×1 convolution to compress the feature channels. Subsequently, 1×1 convolution is used to expand the channels to match the input feature channels, allowing more shallow features to be conveyed within the network and reducing the loss of feature information. Moreover, a coordinate attention branch is introduced to enhance the representation of refined crack features using spatial location coordinate information. This coordinate attention branch performs feature enhancement along both the width and height directions of the input feature map. Specifically, it involves decomposing the feature matrix into aggregated features along the x and y spatial axes, compressing the channels with 1×1 convolution, then batch normalization and nonlinear regression to encode crack spatial information along the x and y axes, ultimately achieving a refined representation of tiny crack details in the form of aggregated channel parameters. Finally, the boundary features F_{Edge} , extracted by the super-resolution reconstruction encoding architecture, are upsampled by 8 times to match the size of the original HR input image. They are then concatenated with the coarse crack feature map F_{Coarse} extracted by the Transformer backbone to form the crack-enhanced feature $F_{Enhanced}$ used for refined segmentation.

3.2 Refined Rendering Head

To perform a GPU-friendly refined decoding for the previously extracted crack deep semantic features with boundary details, the authors, inspired by the PointRend model (Kirillov et al., 2020), developed a point-rendering-based refined prediction head for crack targets. This refined prediction head, necessary for effective decoding of the enhanced crack features, requires two steps: firstly, the selection of point-wise refined rendering points along the boundaries, and then the point-by-point refined rendering based on MLP. For the MLP-based point-by-point refined rendering operation, the authors adopted the same architecture as PointRend, because MLPs, compared to traditional CNNs, offer computational efficiency and predictive accuracy advantages due to their shared weights and point-by-point prediction. In terms of the strategy for selecting refined rendering points along the boundaries, considering that the PointRend architecture is primarily designed for traditional large-size natural scene targets and is not suitable for small crack targets with elongated topological structures, the study customizes it for crack targets.

Training Phase Rendering Point Sampling Method: Each input image has refined labels to effectively supervise the learning process of the model. Extracting boundary information from these refined labels has advantages in computational efficiency and avoiding cumulative errors compared to the uncertain point method used in the PointRend architecture. Therefore, these existing refined labels can be directly used to guide the network, focusing on the crack boundary areas. Specifically, during the training phase, boundary information is directly extracted from the refined labels of crack images to provide guidance for sampling points. The edge detection algorithm is utilized to extract the edges of the refined labels, and some of the sampling points originally uniformly distributed across the background and cracks are concentrated in the extracted boundary areas. It is important to note that to avoid a decline in model performance due to an imbalance in the ratio of positive and negative samples, and to ensure efficient training, the total number of sampling points on each image is set to $N = \frac{H \times W}{20}$, with all sampling points randomly distributed at a ratio of 0.3:0.4:0.3 respectively in the main crack area, the crack boundary (as the accuracy in the boundary area is fifty-fifty), and the background area. For a more intuitive demonstration of this refined label-based training phase sampling point guidance method, Figure 3 visually presents the point sampling method for a randomly selected training sample. Finally, all the sampling points determined on the label are mapped to the corresponding enhanced crack feature map for model training.

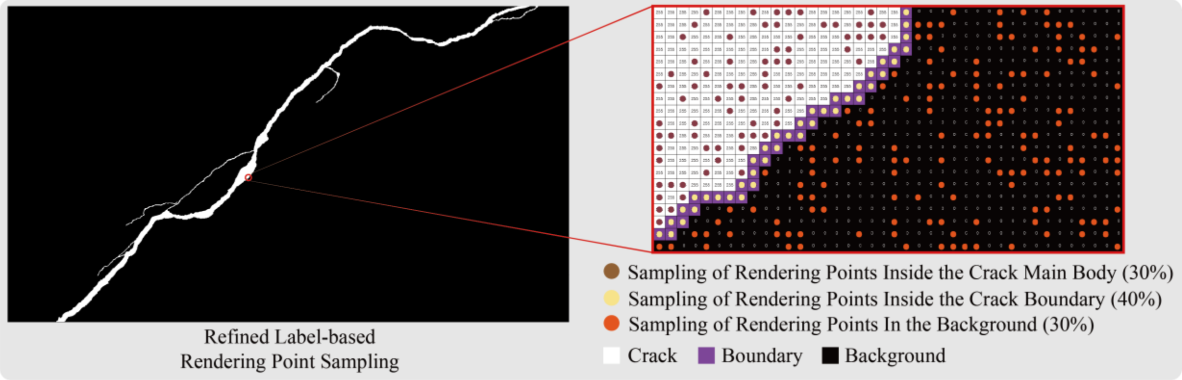
Refined Label-based
Rendering Point Sampling
● Sampling of Rendering Points Inside the Crack Main Body (30%)
● Sampling of Rendering Points Inside the Crack Boundary (40%)
● Sampling of Rendering Points In the Background (30%)
□ Crack
■ Boundary
■ Background

FIGURE 3. Rendering Point Sampling Strategy during the Training Phase

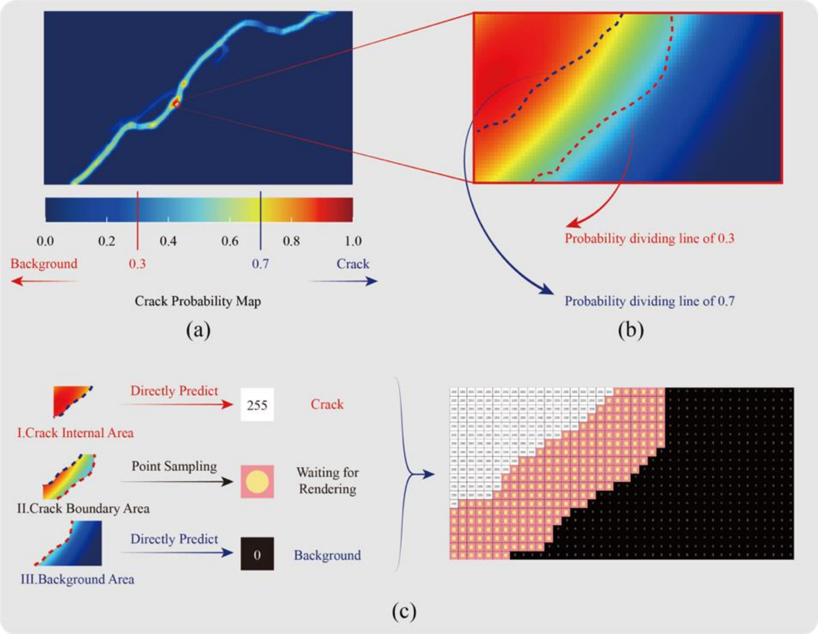

FIGURE 4. Rendering Point Sampling Strategy during the Inference Phase

Inference Phase Rendering Point Sampling Method: Refined labels are available only after the inference is completed; therefore, it is not feasible to use refined labels for guiding sampling points during the inference process. To address this, a boundary-guided rendering point sampling strategy based on a probability heatmap was specifically designed for the inference phase, enabling the model to effectively concentrate computational resources on difficult-to-predict tiny cracks and crack boundary areas. Specifically, the refined probability heatmap was adopted to achieve efficient boundary rendering point guidance. As illustrated in Figure 2, this probability heatmap is primarily obtained by adding two convolution blocks after the crack feature map, which is a concatenation of outputs from the encoder and the boundary-guided branch. Compared to the original PointRend's strategy of guiding rendering point sampling based on coarse segmentation masks, the generation process of probability heatmaps is computationally more direct and efficient. More importantly, probability heatmaps can reflect the likelihood of each pixel belonging to each category, rather than merely assigning it to a simple category. This probabilistic information provides a continuous confidence measure for each pixel, rather than categorizing pixels in a binary manner like in coarse masks, thereby preserving more uncertainty and subtle differences about image areas, which is crucial for understanding refined structures and complex scenes within the image.

Specifically, this study divides the areas on the probability map into three parts based on the differences in prediction probabilities: the background area with a probability close to 0, the crack area with a probability close to 1, and the ambiguous area with a probability around 0.5. The area with probability near 0.5 is considered the region requiring point-wise refined rendering point guidance. Following this principle of area division, and to avoid excessive inference on simple samples (main crack and background), during the inference process, point-wise refined rendering point sampling is performed only in the areas of the probability map where the probability fluctuates around 0.5, focusing solely on these ambiguous prediction areas for refined rendering. For areas on the probability map with probabilities close to 0 and 1, direct mapping to background pixels and crack pixels will be performed on the prediction results, avoiding redundant computation. Notably, the reason for not using the coarse segmentation guidance from the original PointRend is that coarse segmentation, which involves multiple downsampling processes, results in a significant loss of tiny cracks and crack edge details in the crack image. In contrast, the probability heatmap undergoes only one downsampling based on the original image, preserving as much crack detail as possible while consuming less computational resources. To visually represent the refined rendering point sampling method during the inference phase, Figure 4 provides a visualization example of a randomly selected probability heatmap. It is evident that on the probability heatmap, the probabilities in the background and main crack areas are concentrated around 0 and 1 respectively, while in the boundary area, due to manual annotation errors and insignificant color differences, pixel probabilities on the heatmap fluctuate around 0.5. The study sets the hard-to-recognize pixel probability range at 0.3-0.7, and in the subsequent refined rendering phase, only hard-to-recognize samples with probabilities between 0.3-0.7 undergo refined inference. The parameter settings for the sampling points during the training and inference phases will be detailed in Section 4.4.2.

4 EXPERIMENTS

4.1 Datasets

A sufficient number of image samples is a prerequisite for obtaining high-performance DL models. In fact, researchers in the field have already open-sourced several datasets with pixel-level annotations, facilitating the performance testing of models. Table 1 provides a statistical overview of the relevant information on the current mainstream crack segmentation datasets.

TABLE 1. Summary of Some Commonly used Open-source Datasets in The Field of Crack Segmentation

Dataset	Capture Device	Resolution	Scenes	No.Images
CrackForest Dataset	Iphone5	480 × 320	Road surface	118
CrackLS315	Line-array camera	512 × 512	Road surface	315
Cracktree200	Area-array camera	800 × 600	Road surface	206
FIND	Line-array camera	256× 256	Bridge deck, Road surface	2500
DeepCrack	Unknown	544 × 384	Asphalt & concrete	537
Stone331	Area-array camera	512 × 512	Stone Surface	331
Bochum Crack DataSet	Smartphones	512 × 512	Concrete building	370

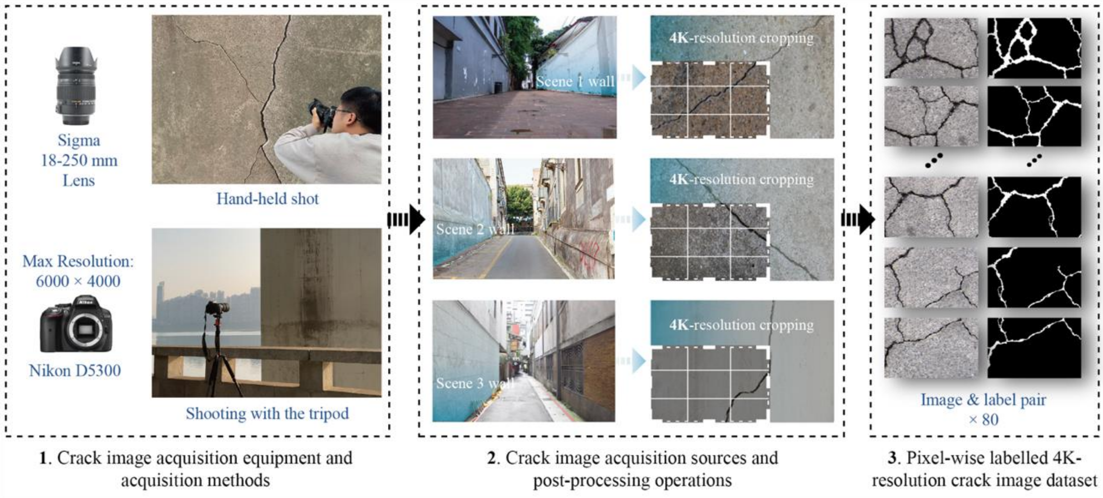

FIGURE 5. Acquisition Process for HR Crack Images and Their Corresponding Refined Masks for Model Performance Evaluation

As shown in Table 1, two clear phenomena can be observed regarding these datasets: Firstly, most crack image datasets originate from roads; secondly, the resolution of almost all crack images does not exceed 600×800 pixels, which are considered low-resolution images. Regarding the former, considering that the method proposed in this study primarily targets concrete structure cracks, which differ in data distribution types from road crack images. Hence, directly using open-source datasets for model training cannot fully exploit the model's potential. As for the latter, training with lower-resolution crack images requires fewer computational resources and is easier for model convergence. After systematically considering two aspects' factors, some low-resolution open-source road crack image datasets were selected for preliminary model training, followed by fine-tuning the model with a small number of collected concrete cracks, as detailed in Section 4.3.1. It is noteworthy that the preference for road crack images over concrete crack images for preliminary training stems not only from the greater availability of road crack images but also from their highly detailed pixel-level annotations. These annotations are essential for training the proposed fine-grained crack segmentation model. In contrast, the available concrete crack datasets typically do not offer this degree of detailed annotation, rendering them less suitable for the specific requirements of this research. Specifically, for the road crack data used in preliminary model training, to ensure enough number of images, three open-source crack image datasets including EdmCrack600 (Mei et al., 2020), Aigle-RN (Chambon and Moliard, 2011), and Crack500 (Yang et al., 2019) were selected as training data sources. All images were uniformly resized to 256 × 256 pixels for training on commercial GPUs. In total, 860 crack images with various background textures and crack patterns were obtained, of which 620 were used for training, 120 for validation, and 120 for preliminary testing.

Regarding the collection process of onsite captured images, as shown in Figure 5, three different buildings' concrete walls in Changsha city were selected as the sources for crack image collection, using a Nikon D5300 camera to capture 300 original 6K resolution crack images from various parts of the walls. Among the images, 220 original HR images were randomly selected and cropped to produce 800 low-resolution images, each with a resolution of 256×256 pixels. These images were used to fine-tune the performance of a model that was initially trained on open-source crack datasets. Additionally, 80 4K HR crack images were selected and cropped from the remaining 250 original HR images for further testing of the proposed method's performance with HR crack images. The reason of using HR crack images for further testing is that higher resolution crack images require higher downsampling in the inference process, making the boundary areas more prone to ambiguous predictions, thus refined processing for HR crack images more significantly highlights the importance of this study. Moreover, it is important to note that for accurate evaluation results, all onsite collected crack images were annotated with precision by the author. The following three measures were mainly taken to ensure accurate and reliable annotation results:

1. HR Image Source: To ensure annotators could confidently and comprehensively annotate, the crack boundaries in the images had to be very clear. Therefore, after image collection, two rounds of selection by different professionals were arranged to remove blurred crack images due to imprecise focus. Additionally, during image collection, the camera's photo storage mode was adjusted to the maximum size mode (6K resolution). By using these HR images, small target details could be clearly presented. And more pixel-level information was provided, aiding in more accurate marking of small targets.

2. Professional Annotation Tools: Professional open-source annotation tool LabelMe was used, which allowing annotators to easily draw pixel-level labels. This annotation tool offers a convenient interface and tools to improve the accuracy of annotations.

3. Training Multiple Annotators and Mutual Verification: Prior to the start of the marking process, annotators were trained, especially in terms of accuracy and consistency when annotating small targets. Trained personnel were guided to

annotate correctly based on the sample images and label references provided in Figure 5. To avoid annotation errors due to individual subjective judgments, labels after each annotation were reviewed by another trained professional annotator for timely correction of subjective errors.

4.2 Evaluation Method

For the quantitative evaluation of the experimental outcomes, we employed two widely recognized metrics: the Intersection over Union (IoU) and the Dice Similarity Coefficient (Dice). Additionally, to underscore the efficacy of our proposed approach in delineating boundaries, we utilized the Mean Boundary Accuracy (mBA) as a metric, a concept pioneered in the study by CascadePSP (Cheng, Chung, Tai and Tang, 2020). The essence of mBA lies in determining the IoU specifically within the boundary region, comparing the Ground Truth (GT) with the predicted segmentation mask. Figure 6 provides a depiction of the procedure for calculating mBA.

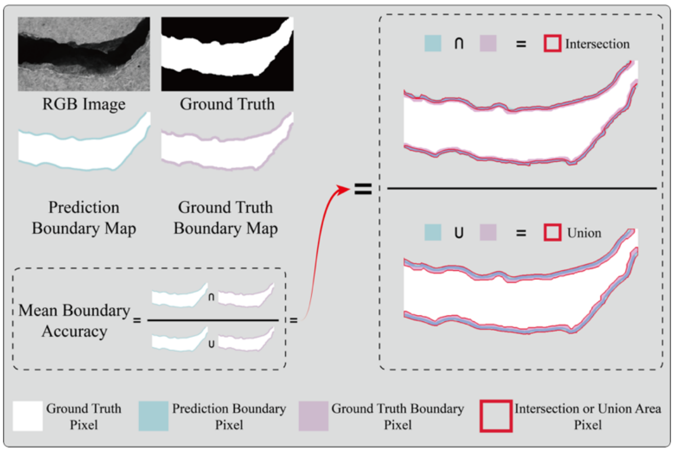

FIGURE 6. Visualization of Mean Boundary Accuracy Calculation Method

4.3 Implementation Details

4.3.1 Hardware Equipment and Hyperparameters

The data processing and all semantic segmentation inference experiments mentioned in this paper were conducted and deployed on a desktop workstation in the laboratory. The hardware configuration of the workstation, including the Central Processing Unit (CPU), Graphics Processing Unit (GPU), and Random Access Memory (RAM), is detailed in Table 2. Additionally, a virtual Python environment was established using Anaconda, configured with the Pytorch DL framework. The specific versions of the related software are also listed in Table 2.

During the training phase, the Adam optimizer was utilized as the optimizer for network training. the Adam optimizer was chosen for its well-documented efficiency in handling sparse gradients and its adaptability to large datasets with HR images. This choice is particularly beneficial for DL tasks involving detailed feature recognition such as in crack segmentation. Compared to other common optimizers like SGD (Stochastic Gradient Descent) and RMSprop, Adam combines the advantages of adaptive gradient algorithm and root mean square propagation, providing an automated adjustment of learning rates (Reyad et al., 2023). This leads to better handling of noise and faster convergence in training deep neural networks, which is crucial for the high variability seen in crack images. Furthermore, Adam's robustness against the vanishing learning rate problem often observed with SGD, coupled with its capability to stabilize the updates due to its momentum component, makes it especially suitable for our application. These characteristics ensure more effective training outcomes in scenarios requiring high precision, as is the case with the segmentation of fine details in crack images.

In addition, a hybrid loss function combining Binary Cross-Entropy (BCE) loss (Zhang and Sabuncu, 2018) and Dice loss (Sun and Li, 2022) was adopted to effectively balance the segmentation accuracy and model sensitivity. The final chosen loss function can be formulated as: $Loss = 0.9 \times BCE + 0.1 \times Dice$. This weighting strategy was meticulously designed to optimize the segmentation of HR crack images using rendering technology, prioritizing precision in identifying crack boundaries while maintaining general segmentation integrity.

The maximum number of training iterations on the low-resolution open-source crack image dataset was set to 800. The batch size was set at 8, with an initial learning rate of 0.007, and a decay of 0.0001 after every 10 training cycles. After completing preliminary training, the same hyperparameter configuration was continued, using the onsite collected concrete crack images to fine-tune the model for 200 cycles, obtaining the final model for subsequent crack segmentation. Training 100 epochs on the workstation used in this study took less than 4 hours. Figure 7 visually displays the changes in loss during the model training process.

TABLE 2. Visualization of Edge Accuracy Calculation Method

Hardware/Software	Parameters/Version
CPU	Inteli7-8700k
GPU	GeForce RTX 3090 24GB
RAM	64GB
System	Ubuntu 18.04
Anaconda	3-4.4.10
Python	3.6.5
PyTorch	1.8.0

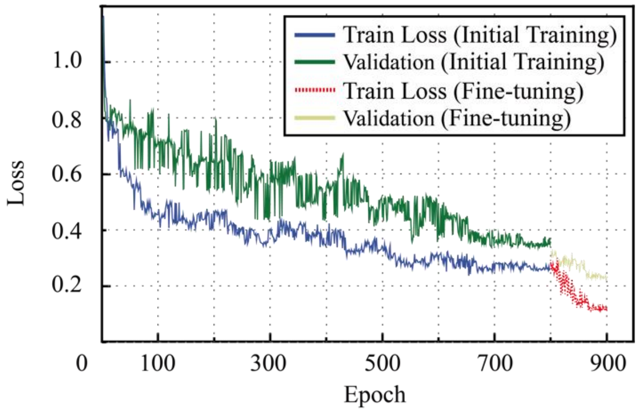

FIGURE 7. Training and Validation Loss Curves for the RLCSN

4.3.2 Sparse Training Method

To further enhance the model's prediction speed, reduce computational load, and minimize model size, sparse training and pruning operations were implemented. Given that the complexity of parameters mainly stems from the Transformer architecture within the encoding structure, sparse training and pruning were focused on the Transformers in both the coarse crack feature backbone and the boundary guidance branch. Sparse training involves adding an L1 norm of weights into the loss function during the model's training process. The L1 norm tends to favor a smaller number of significantly large features during weight updates, penalizing the complexity of the model. The improved model's loss function, as shown in Equation (3), incorporates the added L1 regularization term. The hyperparameter is α and the weights of the convolution kernel is ω. The numeric value of the hyperparameter will affect the sparsity of the final weight.

$$L = -\sum_{j=1}^{T} y_i log p_i + \alpha\|\omega\| \quad (3)$$

Pruning the model primarily involves weight pruning, channel pruning, and convolution kernel pruning. Weight pruning calculates the importance of neurons based on certain rules and prunes them according to an importance threshold. The pruned network is then fine-tuned until the target accuracy is reached, but weight pruning still occupies runtime memory. Channel pruning focuses on feature channels and typically prunes channels based on their effectiveness evaluated by an importance factor. Channel pruning does not rely on specific libraries for computational acceleration. Convolution kernel pruning introduces a regularizing term during weight updates and prunes weights within the convolution kernel based on a threshold. It does not change the output's channel count and does not affect the structure of the next layer's input.

In this study, the weight pruning of convolution kernels was based on the magnitude of connections. After sparse training, the weights of each convolution kernel were sorted by their absolute values, assuming that larger absolute values indicate higher importance and smaller weights contribute less to the output. During sparse training, weights continually reduce to zero. The degree of model sparsity can be determined by setting different levels of sparsity during training. The number of parameters to be retained is decided by setting a threshold. The weights are sorted by their absolute values, and if the pruning rate is set at 30%, the top 70% of the weights are retained.

4.4 Ablation Study

4.4.1 Ablation Study for Crack Feature Encoding Architecture

In this section's ablation experiments, five representative CNN-based feature extraction architectures were selected as the backbone for crack feature extraction. All model experiments were conducted on the crack image dataset collected as mentioned in Section 4.1, to validate the advantages of the Transformer architecture introduced in the crack feature extraction backbone. Specifically, ResNet50 (He et al., 2016), ShuffleNet (Zhang et al., 2018), DenseNet-121 (Huang et al., 2017), and HRNet (Wang et al., 2020) were chosen for experimental analysis. Additionally, the effectiveness of the Boundary Guidance Branch (B_{Edge}) was comparatively examined on each set of models using different crack feature extraction backbones. The results of these comparative experiments are shown in Table 3. They provide a comparative analysis of the models built with different encoding architectures across three aspects: model inference speed, model parameter quantity, and accuracy.

TABLE 3. Ablation Experiment Results on the Selection of Backbone Networks in Crack Feature Architecture and the Presence or Absence of Boundary Guidance Branch

Coarse Crack Feature Enhancement Extraction Architecture	Inference Speed (FPS)		Para. (M)		IoU(%)		mBA(%)		Dice(%)	
	W/ B_{Edge}	W/O B_{Edge}	W/ B_{Edge}	W/O B_{Edge}	W/ B_{Edge}	W/O B_{Edge}	W/ B_{Edge}	W/O B_{Edge}	W/ B_{Edge}	W/O B_{Edge}
ResNet50	12.35	15.67	25.64	21.29	78.20	77.45	80.11	76.35	82.97	81.58
ShuffleNet	25.82	31.63	8.32	3.97	76.55	75.29	79.10	75.12	82.38	80.97
DenseNet-121	10.16	13.76	14.37	10.02	78.76	77.17	81.69	77.06	84.13	83.43
HRNet	6.54	8.61	31.53	27.18	81.32	80.33	84.53	80.07	85.16	83.76
SWIN Transformer	12.76	15.76	29.15	24.80	83.21	81.98	85.68	81.31	86.74	85.33

Through parallel comparison of crack recognition models constructed using five different feature extraction architectures, it is observed that the encoder employing the SWIN Transformer model to achieve the best balance in terms of inference speed, model parameter quantity, and recognition accuracy. It can achieve recognition results with IoU, mBA, and Dice scores of 83.21%, 85.68%, and 86.74%, respectively, at an inference speed of 12.76 FPS on 4K resolution crack images. This represents an average improvement of 4.50%, 4.32%, and 3.08%, compared with the remaining four groups of architectures based on CNNs in the three accuracy metrics. This is because the Transformer model, compared to CNN, incorporates an additional multi-head self-attention mechanism, which gives the model the ability to establish long-distance dependencies for crack targets dispersed throughout the global image. This enables the model to perform self-attention operations on multi-scale feature maps, effectively eliminating the interference of background noise. This makes the Transformer model stronger in recognition performance and robustness when dealing with cracks against complex backgrounds than traditional architectures built with CNNs. When considering the quantity of the model's parameter and inference speed, in practical engineering applications, improving the accuracy and recall rate of crack detection is often more of a concern for inspection departments than the speed and efficiency of recognition. Of course, this fact does not mean that the model can indiscriminately sacrifice efficiency for improved recognition accuracy. Instead, the inference speed and model parameter size of the architecture built using SWIN Transformer meet the requirements of practical engineering crack detection tasks, which will be validated in the UAV-based field experiments in Section 5. Additionally, comparing all models with the added edge guidance branch, it is observed that the edge guidance branch, with a parameter size of 4.35M, brings an average improvement of 1.16%, 4.24%, and 1.26% in IoU, mBA, and Dice, respectively. The most significant improvement is in mBA, further illustrating the effectiveness of B_{Edge} in improving crack boundary detail information in the deep semantic feature maps of cracks.

4.4.2 Ablation Study for Point Rendering Head

To fully demonstrate the effectiveness of the decoding architecture proposed in this study in refining crack details, a parallel comparison was made to compare it with the most advanced self-attention-based decoding architectures at first. To ensure the validity of the experimental results, the comparison decoding architectures used the same training parameters as the method proposed in this study, ensuring that the models converged to their optimal state. The corresponding experimental results on the test set are shown in Table 4. From the first two rows of Table 4, it can be seen that the point rendering-based decoding architecture achieves a certain degree of improvement over traditional decoding architectures in IoU, Dice, and mBA, with the most noticeable improvement in mBA, reaching 86.98%. This is because the MLP in the point rendering-based decoding architecture is position-sensitive, calculating the prediction value independently for each pixel. Therefore, it can flexibly capture details and spatial relationships in crack images. In contrast, although the self-attention-based decoding architecture can enhance the mutual representation of global tiny cracks through self-attention during decoding, it struggles to capture local crack details due to the discrete feature sampling method.

After validating the effectiveness of the proposed decoding architecture, it is necessary to conduct parameter performance experiments on the architecture to obtain the most suitable parameter configuration for the model structure. As detailed in Section 3.2, two different types of point-wise refined rendering point sampling methods were adopted for the training and inference phases. Therefore, two sets of parameter performance experiments will be conducted next, to obtain relatively optimal point sampling parameters for the training and inference processes.

Training Phase Point Sampling Parameter Experiment: Before conducting the parameter experiment, it can be known from the probability heatmap that the areas most likely to produce incorrect predictions are mainly distributed around the crack boundary, not just a single-pixel-wide crack boundary contour. This is due to the unavoidable errors between the real boundary and the label boundary caused by the subjectivity of manual annotation. Therefore, if only the boundary training point sampling method shown in Figure 3 is used, which samples only the boundary with a width of one pixel, it cannot effectively avoid the above-mentioned bias guidance, negatively impacting the model's ability to refine boundaries. To eliminate the negative impact of this incorrect guidance on the model's boundary recognition, an effective method is to expand sampling over the boundary and its adjacent areas. The expansion of the refined point sampling area to encompass regions prone to subjective errors effectively eliminates these errors through dense resampling. To implement boundary expansion, the study first used a Sobel operator-based edge detector to extract crack boundaries with a width of one pixel. Then, with this boundary as the dilation center, uniform dilation is carried out towards both the crack interior and the background areas, according to the preset dilation coefficient. Figure 8 shows four sets of different parameter boundary expansions. Considering the size of the crack images and the pixel width, the expanded boundary widths are 3, 5, 7, and 9 pixels, respectively. Finally, the coordinates within the dilation area are mapped onto the crack feature map for feature sampling during training. The total number of sampling points on each training image is $N = \frac{H \times W}{20}$, with 30%, 40%, and 30% of the points randomly distributed in the background, expanded crack edge, and crack interior regions, respectively.

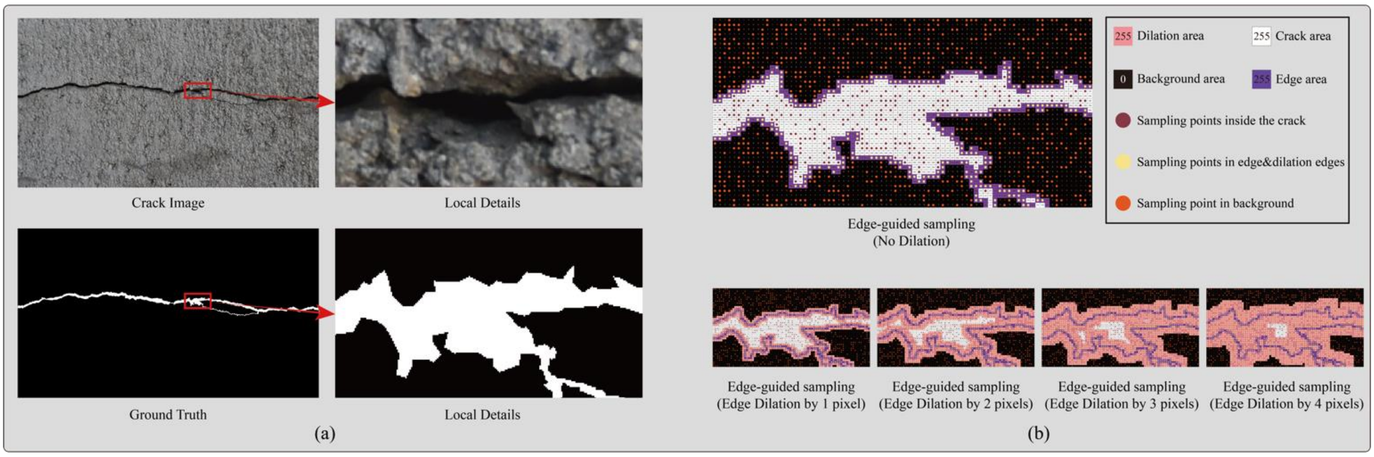
Crack Image
Local Details
Ground Truth
Local Details
(a)
255 Dilation area
255 Crack area
0 Background area
255 Edge area
Sampling points inside the crack
Sampling points in edge&dilation edges
Sampling point in background
Edge-guided sampling (No Dilation)
Edge-guided sampling (Edge Dilation by 1 pixel)
Edge-guided sampling (Edge Dilation by 2 pixels)
Edge-guided sampling (Edge Dilation by 3 pixels)
Edge-guided sampling (Edge Dilation by 4 pixels)
(b)

FIGURE 8. Visualization of the Label Boundary-guided Rendering Point Sampling on a Randomly Selected Crack Image at Different Dilation Coefficients

Table 4 provides a detailed statistical analysis of the performance of models trained using different widths of expanded boundary guidance. Observation of Table 4 reveals that when the width of the expanded boundary is 7 pixels (i.e., a dilation coefficient of 3), the model's performance is relatively superior, with the average IoU, mBA, and Dice exceeding those of models trained using the other four expansion methods by at least 2.13%, 3.36%, and 1.49%, respectively. This is because the subjective annotation errors at the crack boundaries in the crack training dataset used in this study are precisely within the range of ±2 pixels from the boundary. A comprehensive analysis of the visualized results of Table 4 and the detailed visualization of the boundary heatmap in Figure 3 indicates that too narrow a boundary expansion area results in insufficient coverage of the subjective error area, leading to less significant performance improvement. Conversely, too wide an expansion area reduces performance enhancement because computational resources are scattered by simple sample areas outside the error range. Specifically, when the dilation coefficient is 1 or 2, the post-dilation boundary area is insufficient to encompass the biases generated by manual annotation near the boundary area. When the dilation coefficient is 4, the main crack area and excessive background areas without manual annotation errors are included as ambiguous boundary regions requiring refined sampling. These unnecessary simple sample areas divert computational power that should be allocated to ambiguous boundary regions, thereby reducing the model's learning and representation capacity for such areas and limiting the performance improvement brought by boundary-guided sampling.

Inference Phase Point Sampling Parameter Performance Experiment: To enable the model to achieve an effective balance between inference accuracy and efficiency, it is necessary to determine a reasonable probability range on the probability heatmap for areas with uncertain prediction results around 0.5. A larger probability range means more points require refined rendering, which increases accuracy but also significantly raises computational redundancy in the inference process. Conversely, a smaller probability range, while speeding up inference, may fail to render tiny cracks and boundary details effectively, severely impacting the final recognition accuracy. Therefore, choosing an appropriate range for refined rendering probability becomes a key issue to address. Specifically, two probability values are selected, namely, the critical probability value α between the background and boundary areas, and the critical probability value β between the boundary areas and crack pixels.

For the critical probability value α, this study set three different probability parameters: 0.2, 0.3, and 0.4. similarly, for β, three different probability parameters were set: 0.6, 0.7, and 0.8. These six critical probability values (α and β) collectively define nine different boundary regions with varying probability ranges. Table 5 statistically analyzes the inference results on the test set for the model that applied sampling using these nine different probability ranges.

TABLE 4. Performance Comparison of Models Trained Using Different Feature Point Sampling Strategies

Sampling point extraction method for the training phase	No.	Dilating coefficient	Width of the boundary area after dilating	IoU(%)	mBA(%)	Dice(%)
Uniform sampling	1	/	/	83.21	85.68	86.74
Boundary guided sampling	2	1	3	83.78	86.98	87.65
	3	2	5	84.77	87.68	87.93
	4	3	7	86.09	90.26	89.11

	5	4	9	83.32	86.03	87.27

TABLE 5. Performance Comparison of Models Guided by Different Boundary Probability Range Sampling Strategies During the Inference Phase

Set No.	Background area probability range	Crack edge area probability range	Crack body area probability range	IoU	Dice	mBA
1	（0.0,0.2）	（0.2,0.6）	（0.6,1.0）	86.09	90.26	89.11
2	（0.0,0.2）	（0.2,0.7）	（0.7,1.0）	86.25	90.78	89.34
3	（0.0,0.2）	（0.2,0.8）	（0.8,1.0）	86.97	91.29	90.06
4	（0.0,0.3）	（0.3,0.6）	（0.6,1.0）	86.40	90.61	89.37
5	（0.0,0.3）	（0.3,0.7）	（0.7,1.0）	88.37	94.06	93.25
6	（0.0,0.3）	（0.3,0.8）	（0.8,1.0）	87.65	92.36	91.07
7	（0.0,0.4）	（0.4,0.6）	（0.6,1.0）	85.03	87.32	87.66
8	（0.0,0.4）	（0.4,0.7）	（0.7,1.0）	86.60	90.83	89.98
9	（0.0,0.4）	（0.4,0.8）	（0.8,1.0）	87.35	91.16	92.30

TABLE 6. Performance Comparison of Models Equipped with Coarse Crack Feature Extraction Backbones of Different Sparsity Levels

Encoder type	Sparsity	IoU （%）	mBA （%）	Dice （%）	Inference speed （FPS）	The number of model parameters （M）
Transformer	-	88.37	94.06	93.25	4.76	29.15
Prunning-Transformer	20	87.57	93.66	92.78	5.14	23.84
	40	87.13	93.45	91.34	8.76	16.53
	60	78.67	89.76	84.31	11.92	11.22
	80	70.36	85.93	76.22	15.70	7.91

As shown in Table 5, groups four, five, and six (experiments with background area probability range between 0.0 and 0.3) achieved relatively better IoU, Dice, and mBA scores. This is because, compared to the sampling group with a background probability range of 0.0 to 0.4, these three groups encompassed a wider background sampling area, helping to repair some tiny crack details undetected in the background. Simultaneously, the sampling group with a background probability range set between 0.0 and 0.2 misclassified too many pixels from the ambiguous boundary areas as background pixels. This resulted in a lack of sufficient sampling points for accurately repairing the boundary details, leading to a comparatively lower mBA. Additionally, comparing groups 4, 5, and 6, it's observed that the model's inference accuracy is highest when the boundary area's range is set between 0.3 and 0.7, with IoU, Dice, and mBA reaching 88.37%, 94.06%, and 93.25%, respectively. This is because the crack body area, compared to the background and edge areas, is a simpler sample with a higher prediction probability (often over 80%), thus not requiring a wider probability range. Whereas the boundary area, being a transitional zone between the background and crack body, often exhibits indistinct pixel colors and contrasts, leading to significant fluctuations in prediction probability, hence necessitating a broader probability range. Ultimately, the parameter configuration of group four was chosen as the optimal sampling parameter for the inference phase to control subsequent experiment inferences. Essentially, the experiment results also indirectly confirm that the main reason for inadequate crack segmentation accuracy concentrates in the ambiguous boundary area, which typically falls within the probability range of 0.3 to 0.7 on the coarse segmentation probability map. Therefore, this study adopts this probability range for guiding refined rendering point sampling during the inference phase in subsequent experiments.

Furthermore, it should be noted that although specific datasets and models may display distinct characteristics, the optimal thresholds determined in this study—0.3 and 0.7—provide a practical framework for other rendering-based fine-grained boundary segmentation tasks. These thresholds can be adapted based on further empirical analysis, such as ROC curve evaluations, to suit different data distributions or the specific needs of new segmentation models. This adaptability ensures that our findings are applicable across various contexts and enhances the precision of the rendering-based segmentation technique.

4.4.3 Ablation Study for Pruning Operations

To verify the effectiveness of model pruning and provide optimal parameter configurations for deployment on carrier devices, an ablation study was conducted on different pruning sparsities for the Transformer architecture in the encoding structure. Specifically, pruning sparsities of 20%, 40%, 60%, and 80% were set for the model's encoding architecture. The pruning training iterations were set to 500 epochs for each. The accuracy, inference speed, and parameter complexity of each model were compared to select the relatively optimal pruning parameters. The test results are shown in Table 6.

The results indicate that with increasing pruning sparsity, the training and validation accuracy of the Pruning Transformer gradually decreases, along with a reduction in prediction time and model size. When the pruning sparsity is between 20%-40%, the loss in accuracy is relatively slow, with average decreases in IoU, mBA, and Dice within 1.02%, 0.51%, and 1.19% respectively. The decrease in mBA is not very significant, which suggests that pruning the coarse crack feature extraction encoding architecture does not impact the boundary-guided branch's ability to extract refined details of the cracks. This further demonstrates the beneficial contribution of the boundary-guided branch proposed in this study to the network's refined segmentation. However, as the sparsity of the coarse crack feature extraction backbone continues to increase, the drop in segmentation accuracy becomes more pronounced. This is because an overly sparse coarse crack feature extraction backbone will miss a large number of features in the crack body area outside the crack edges, and these lost crack body features cannot be effectively recovered from the boundary-guided branch. Specifically, when the sparsity is at 80%, although the total parameter count of the model is reduced to 27% of its original, the IoU drops nearly 20%, no longer meeting the detection requirements of practical engineering. After comprehensively considering the model's recognition accuracy, inference speed, and parameter complexity, the model with 40% sparsity is finally selected for subsequent practical engineering detection. This model achieves a lightweight deployment with 16.53 million parameters while ensuring a boundary recognition accuracy of 93.45% for 4K resolution crack images at a real-time inference speed of 24.76 FPS.

TABLE 7. Comparison of Segmentation Results on UAV-collected Images Between the Refined Segmentation Method Guided by Probability Heatmap Proposed in This Study and the Original Pointrend Architecture Guided by Different Coarse Segmentation Masks

Meticulous segmentation architecture	Source of the boundary sampling guidance		Coarse segmentation accuracy			Refined segmentation accuracy		
			IoU	mBA	Dice	IoU	mBA	Dice
PointRend	Coarse segmentation guidance	SegNet	76.69	71.18	80.54	84.14	85.32	87.02
		UNet	78.38	74.37	82.12	84.21	85.66	87.17
		PSPNet	79.32	75.09	83.44	84.30	85.71	87.81
		BiseNet	79.45	75.56	83.89	84.32	85.89	87.92
		SANet	80.15	76.78	84.02	84.60	86.23	88.72
RLCSN	Probability heat map guidance	Probability interval ∈ [0.3,0.7]	/			87.13	93.45	91.34

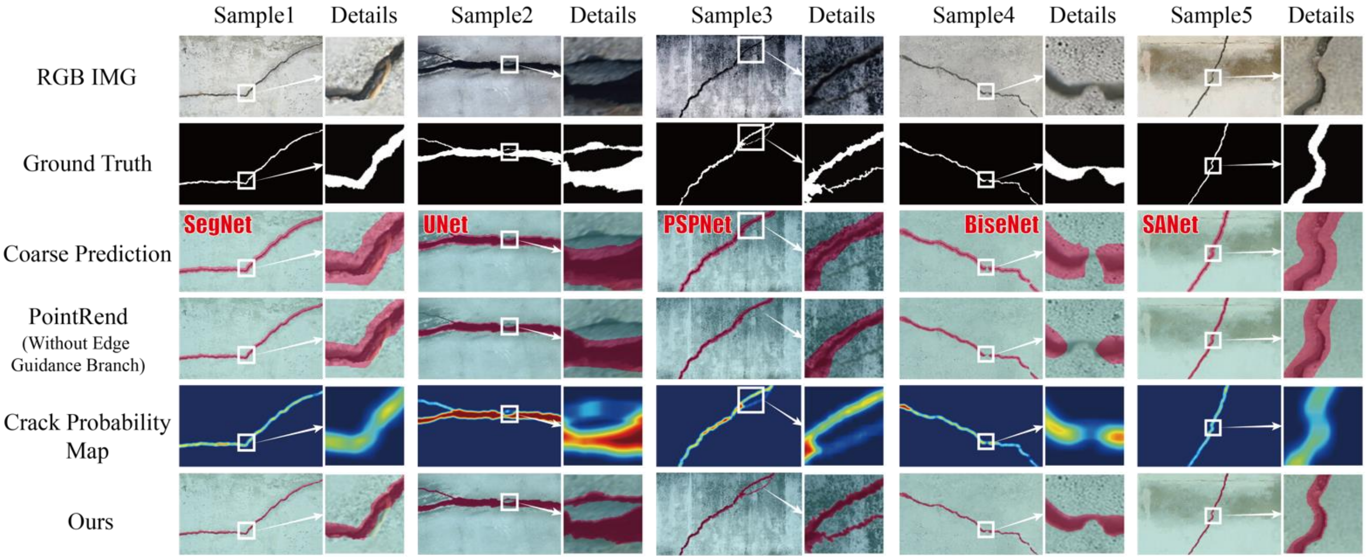

FIGURE 9. Comparison of Refined Segmentation Results on the Self-built HR Crack Image Dataset Using Different Coarse Segmentation Masks Guided by the Original Pointrend Architecture and the Probability Heatmap-guided RLCSN

4.5 Performance Comparison between RLCSN and PointRend

Since RLCSN is an architecture specifically proposed for crack target segmentation based on the original PointRend model, its main improvement lies in using a more accurate refined probability heatmap for guidance during the inference phase. To further illustrate the effectiveness of this improvement, this section compares the segmentation results of the PointRend model guided by different coarse segmentations with those guided by the probability heatmap proposed in this study, using HR crack images collected in the field. Specifically, five mainstream DL segmentation architectures of varying precision, from coarse to fine, including SegNet (Badrinarayanan et al., 2017), UNet (Ronneberger, Fischer and Brox, 2015), PSPNet (Zhao et al., 2017), BiseNet (Yu et al., 2018), and SANet (Fan and Ling, 2017), were selected as the coarse segmentation-generating networks for the original PointRend architecture. In contrast, the method proposed in this study performs boundary sampling point guidance for the point-rendering head based on the probability heatmap proposed from the enhanced crack features extracted by the encoder and boundary-guided branch. It's important to note that all coarse segmentation architectures and refined segmentation networks were trained in the same DL framework under the same configuration with default optimal parameters. Moreover, when predicting with the trained coarse segmentation models, all HR images were proportionally downsized to a long side of 900 pixels to avoid GPU memory overflow due to excessively high original resolutions.

The experimental results are shown in Table 7. Firstly, observing the 2nd to 5th rows (from top to bottom) guided by coarse segmentation models, it's apparent that different coarse segmentation mask generating architectures produce noticeably varied prediction results. From SegNet, the least accurate, to SANet, the most accurate, the gaps in IoU, mBA, and Dice reach 3.46%, 5.60%, and 3.48%, respectively. However, after applying the original PointRend model for refinement, the differences in refined prediction results become less pronounced, with all five sets of experiment results fluctuating within 84.37±0.23% for IoU, 85.78±0.46% for mBA, and 87.87±0.85% for Dice. These results indicate that PointRend's refined segmentation method is indeed independent of specific coarse segmentation masks and robust to different sources of coarse-grained crack features. However, comparing the final experiment results with the best segmentation results guided by coarse segmentation generated by RefineNet in the PointRend group, it's found that the method guided by the probability heatmap further improves the accuracy of segmentation results, with IoU, mBA, and Dice reaching 87.13%, 93.45%, and 91.34%, respectively. Notably, the most significant improvement is observed in mBA, more than double the improvements in IoU and Dice, at 7.22%. This excellent robustness largely benefits from the edge-guided branch introduced in the feature extraction stage and the rendering point sampling strategy guided by the probability heatmap during the inference stage in this study. These two improvements effectively preserve edge areas and tiny crack pixels, allowing them to be finely characterized through dense point-by-point rendering. Especially the latter, which employs probability heatmaps congruent with the input image dimensions to supplant coarse segmentation masks for guiding rendering points, effectively mitigates the loss of tiny crack pixel details that typically occurs during the downsampling process involved in generating coarse segmentation masks. To further demonstrate the validity of the above conclusions, Figure 9 visualizes the test results of five randomly selected HR crack images collected in the field. From the close-up details in Figure 9, it can be more intuitively seen that the method proposed in this study, which guides sampling points using a probability heatmap, significantly outperforms the traditional method guided by coarse segmentation in terms of performance in crack edge detail and tiny crack branch repair.

5 A CASE STUDY

To further evaluate the advantages of the proposed method in processing HR bridge crack images, the UAV was used to collect crack images of the approach bridge of Fuyuan Road Bridge in Changsha City. Fuyuan Road Bridge is a reinforced concrete beam-arch bridge, with the main span consisting of a combination structure of a basket-type steel arch and beams. The bridge is 3575 meters long and was opened to traffic in 2012, serving as one of the most important river-crossing channels in the northern part of Changsha. With the increasing service years and traffic load, large areas of crack damage have appeared on the surfaces of some beams and pier structures. As shown in Figure 10, the DJI M300RTK UAV equipped with a 20-megapixel H20T multi-sensor camera was used for HR crack image collection of the beams and piers. It is important to note that random gusts encountered during the mission can negatively impact UAV flight safety and image collection quality. Therefore, this study implemented three specific controls during the image collection process to minimize negative effects on image quality. Firstly, to ensure UAV flight safety, the minimum distance between the camera and the beam was set at about 3 meters. This distance not only prevents collisions between the UAV and the bridge due to gusts but also ensures that tiny cracks with a width of 0.15 mm or more are fully presented in the collected RGB images as effective pixels. Secondly, a Z15 Aladdin searchlight was added to the lower gimbal for light supplementation to reduce lens defocus blurring due to uneven lighting in the field of view. Lastly, the UAV's flight speed during image collection was controlled at 1 m/s to ensure the camera lens had sufficient time to focus, and images were collected in video recording mode at 4K resolution to avoid missing detections due to insufficient image sampling frequency. Additionally, it is noteworthy that the high-precision inertial measurement unit, flight control system, visual positioning system, and laser rangefinder on the M300RTK, together forming a multi-modal positioning system, maintained a cruising accuracy error within 2 cm in three-dimensional space. This ensured the accurate execution of the collection process as planned, securing the collection of clear crack images at 4K resolution. Ultimately, this study extracted 100 4K resolution (3840 × 2160) crack images from the UAV-collected videos. Following the same annotation guidelines as the CFD dataset, the study used the open-source labeling software Labelme to perform pixel-level annotations on all collected HR crack images, resulting in 100 images with pixel-level labels featuring refined edge details, used for accurate assessment of the detection results.

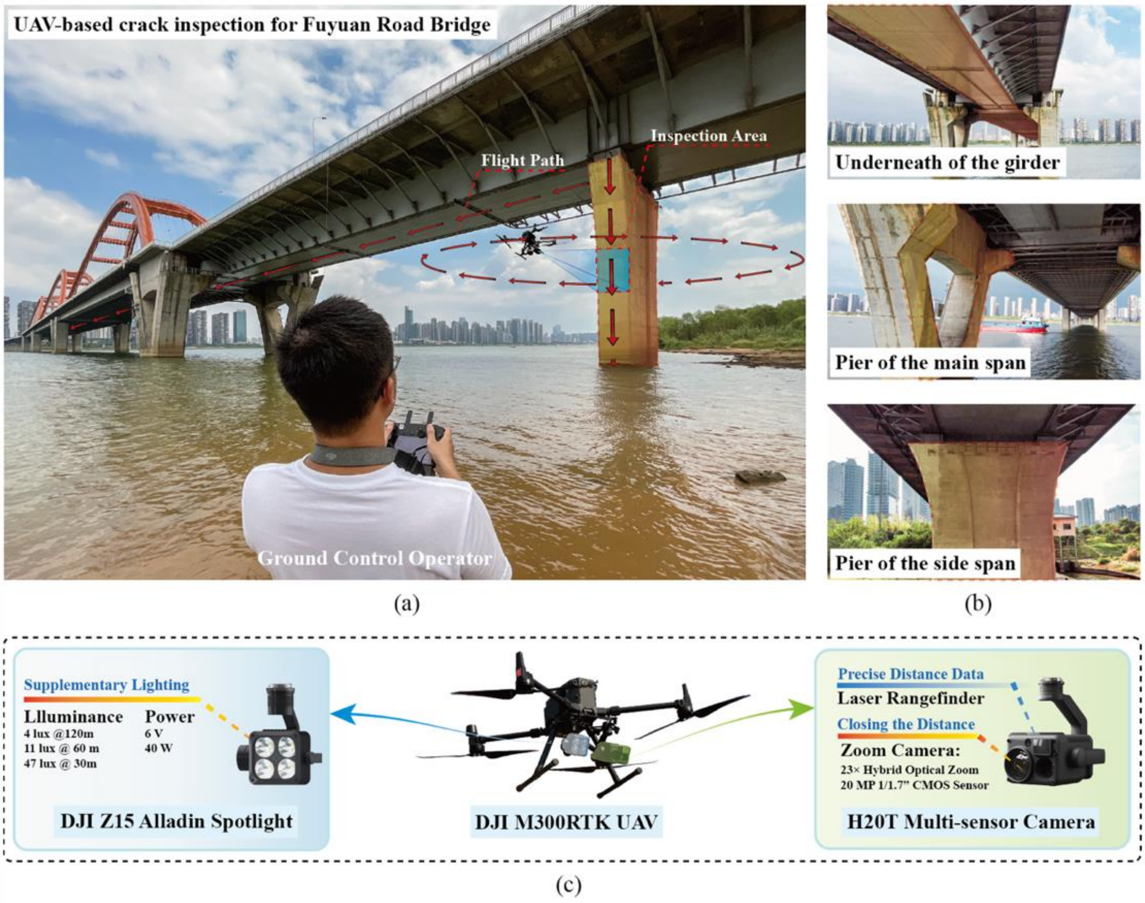

FIGURE 10. Details of Crack Image Collection for Fuyuan Road Bridge: (a) Manually Control the UAV for Crack Image Collection, (b) Inspection Areas, (c) UAV Equipment Information

To effectively validate the advanced nature of the model proposed in this study, the most advanced segmentation methods for HR crack images were used to test UAV-collected images. The performance of all models involved in the comparison was assessed in terms of model complexity, inference speed, and crack recognition accuracy. The HR crack image segmentation technologies for comparison are divided into two categories: The first category includes low-resolution image segmentation frameworks integrated with image preprocessing techniques, and the second category consists of DL segmentation frameworks capable of directly processing HR crack images. Regarding the first category, researchers selected two typical image preprocessing techniques, including sliding windows and proportional scaling, and tested five representative low-resolution image segmentation frameworks from each era, including SegNet, UNet, PSPNet, BiseNet, and SANet. It is important to note that, to ensure fairness in the comparison, the size of the sliding window and the size of the images after proportional scaling were both set to 900 × 900. For the second category, this study chose the most advanced CascadePSP and Segfix network architectures, which perform refined inference for HR crack images from perspectives of cascaded refinement and global progressive refinement, respectively. Regarding the training of all models involved in the test, these models used default pretrained parameter configurations to ensure stable performance. Additionally, to ensure comparability between segmentation architectures developed for natural scene images and those specifically developed for crack scenes in this study, all models were fine-tuned using the crack image data described in Section 4.1. Furthermore, considering that the methods proposed in this study and CascadePSP rely on coarse segmentation masks as prior guiding information for inference. Hence, to ensure the effectiveness of parallel comparison, all coarse segmentation masks were produced using the output of the SANet architecture, labeled No1 in the first category. Table 8 provides a statistical summary of the performance of all models involved in the test. Firstly, by comparing the average performance of the two major categories of methods, it is evident that the first category, which requires image preprocessing before performing inference on low-resolution crack images, is not as effective as the second category that directly processes HR images. The second category, which performs inference directly on HR images, achieved average IoU, mBA, and Dice of 83.27%, 87.79%, and 88.74%, respectively, representing improvements of 3.92%, 11.97%, and 4.94% over the first category. The fundamental reason for this phenomenon is that sliding window operations and proportional scaling, respectively, cause the loss of semantic integrity among global pixels and local tiny crack details in HR images. CascadePSP, Segfix, and the methods proposed in this study, through cascaded, progressive repair, and rendering operations, respectively, rectify these negative impacts, thereby obtaining more accurate refined segmentation masks. While the field tests have confirmed the improvements in safety and efficiency of UAV-based bridge crack detection brought by RLSCN, there are some notable details that may constrain the effective dissemination of this method. Below are the enumerated limitations along with suggested improvements:

- Dependence on High-Quality Probability Heatmaps: The performance of this method heavily relies on the accuracy of probability heatmaps. If the probability heatmaps cannot accurately predict the location and shape of the cracks, the final edge segmentation might be imprecise or could lead to mis-segmentation, resulting in blurred edges or lost crack information. The accuracy of the probability heatmaps largely depends on the feature extraction branch built on the Transformer, which is supervised by refined labels during the initial training phase. Thus, more refined training labels mean more accurate probability heatmaps.
- Sensitivity to Noise and Outliers: Probability heatmaps might be overly sensitive to noise and outliers in the image, which could lead to incorrect crack edge generation during segmentation, especially in cases where the crack boundaries are not clear or the contrast with the background is low. Therefore, ensuring uniform illumination during the crack image collection process is an effective way to avoid such errors.
- Complexity in Parameter Tuning: When using different DL models or facing crack images from different scenes, optimizing segmentation effects might require fine-tuning the threshold parameters defined on the probability heatmaps that determine the crack edge areas, which can increase the difficulty of applying and extending the model in different scenes. However, it should be noted that although specific datasets and models may exhibit different characteristics, the optimal thresholds identified in this study (0.3 and 0.7) provide a reliable benchmark for other rendering-based fine-grained boundary segmentation tasks. These thresholds can be efficiently adjusted based on further empirical analysis (such as ROC curve evaluation) to adapt to different data distributions or specific requirements of new segmentation models.

TABLE 8. Comparison of Segmentation Performance on HR Crack Images Collected by the UAV Using Several Current Mainstream Methods for HR Crack Image Segmentation

Method category	Image preprocessing methods	DL-based segmentation architecture	IoU (%)	mBA (%)	Dice (%)	Para. (M)	Speed (FPS)	GPU (GB)
Coarse segmentation method	sliding window	SegNet	77.34	74.23	82.07	29.80	31.42	6.74
	downsampling		76.69	71.18	80.54		321.76	
	sliding window	UNet	79.65	76.08	84.19	31.12	28.73	9.23
	downsampling		78.38	74.37	82.12		294.16	
	sliding window	PSPNet	80.58	77.34	85.15	46.78	21.37	11.37
	downsampling		79.32	75.09	83.44		218.83	
	sliding window	BiseNet	80.78	78.18	85.98	15.67	36.53	8.43
	downsampling		79.45	75.56	83.89		374.07	
	sliding window	SANet	81.30	79.33	86.57	17.34	29.46	11.78
	downsampling		80.15	76.78	84.02		301.67	
Refined segmentation method	Coarse mask guidance	CascadePSP	81.88	85.23	87.60	47.56	3.78	13.03
	Coarse mask guidance	Segfix	82.49	86.37	88.12	64.70	2.56	18.23
	Probability heat map guidance	RLCSN	85.49	91.76	90.50	16.53	8.76	7.87

Upon further observation of the method proposed in this study and the two HR crack image segmentation methods involved in the comparison, it is evident that the method proposed in this study has significant advantages in both recognition accuracy and inference efficiency. Compared to the previously highest-performing Segfix model, RLCSN achieves IoU, mBA, and Dice scores of 85.49%, 91.76%, and 90.50%, respectively, on 4K resolution crack images, with only a quarter of the model parameters and three times the inference speed. Moreover, among the three accuracy evaluation metrics, the most notable improvement of mBA in RLCSN, exceeding CascadePSP and Segfix by an average of over 5.96%. This once again confirms the effectiveness of the boundary point sampling-based training and inference strategy proposed in this study. It demonstrates that by rationally allocating computational resources from simple sample points concentrated in the background and crack interior areas to difficult boundary points, the model effectively improves the recognition accuracy of ambiguous boundary areas on coarse segmentation masks without increasing dependency on computational resources. To further substantiate these conclusions, Figure 11 randomly selects and visually presents the prediction results of the second category models with relatively better recognition performance. A comparison of the prediction results and close-up details clearly shows that the method proposed in this study outperforms the other methods in terms of crack recognition completeness and the refinement of boundary areas, further illustrating the effectiveness of the approach proposed in this study.

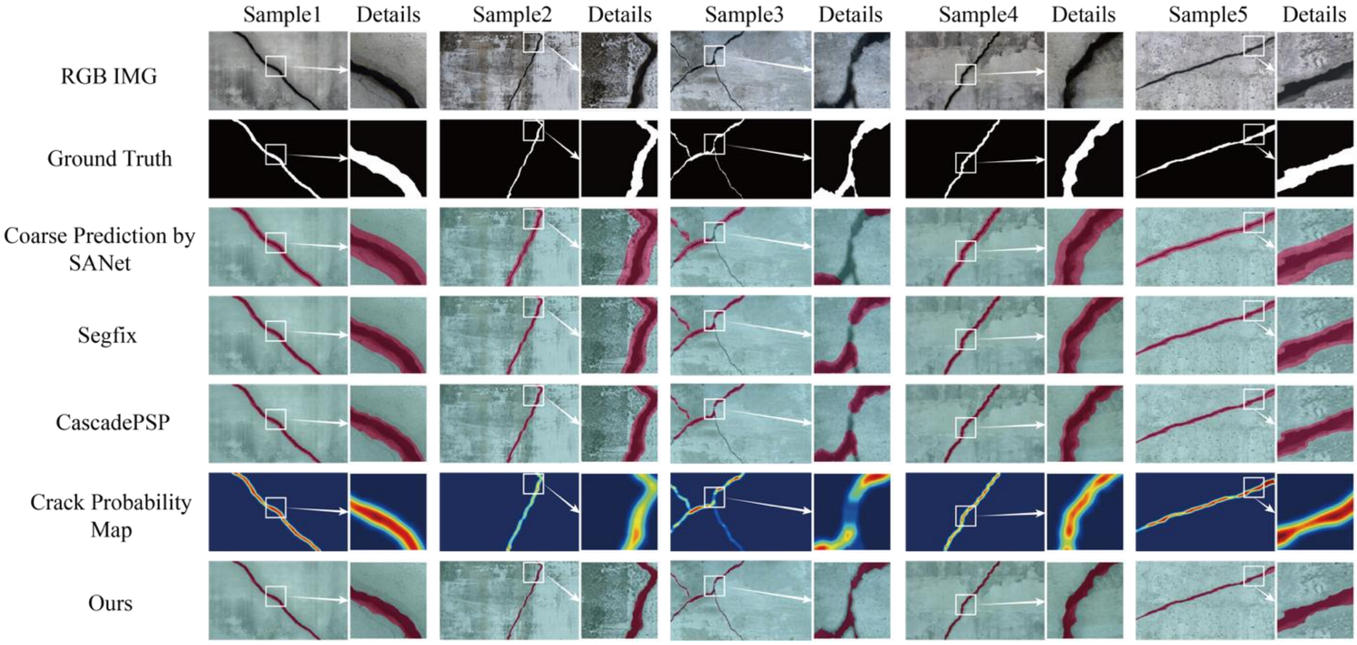

FIGURE 11. Visualization of the Segmentation Performance on Five Randomly Selected UAV-collected Images by the Current Two Most Advanced Coarse Segmentation-guided Refined Segmentation Methods and the Probability Heatmap-guided RLCSN Proposed in this Study

Furthermore, the field test has also demonstrated that the method proposed in this study not only excels in analyzing HR images for crack detection but also exhibits outstanding ability to detect tiny, subtle, and distant cracks, which are often difficult to identify using low-resolution methods. This capability is particularly important as it addresses a critical gap in current crack detection methodologies, where smaller or more distant cracks might go undetected, potentially impacting structural health assessments.

6 CONCLUSION

This study introduces the RLCSN, inspired by refined rendering graphic representation architectures in computer graphics. Addressing three key issues faced by high-precision rendering heads in HR crack image segmentation, the authors have made three targeted improvements, allowing the advantages of the rendering method in accuracy and computational resource friendliness to be fully realized in HR crack image segmentation. Firstly, a deep semantic feature extraction architecture combining Transformer with a super-resolution boundary-guided branch was designed, effectively reducing background noise interference and preserving crack edge details. Secondly, two types of refined rendering point sampling methods were customized for the training and inference phases, enabling the model to concentrate computational power on ambiguous crack edges and tiny crack areas. Thirdly, an efficient sparse training method was developed for the initial RLCSN build, incorporating an L1 norm of weights in the loss function and executing pruning at the weight level to achieve model lightweighting. Through these customizations, refined rendering methods originally used in computer graphics for scene rendering can be effectively applied to the refined segmentation of crack images. Especially, the rendering representation method's high precision and GPU resource-friendly characteristics in edge refinement are fully exploited. The main conclusions of this paper are as follows:

1. RLCSN, as the first architecture to employ point rendering methods for refined segmentation of HR crack images, can generate finely detailed boundary prediction masks for 4K resolution crack images at a speed of 8.76 FPS on a commercial GPU with only 8GB of memory. The method achieved an IoU of 85.49%, an mBA of 91.76%, and a Dice coefficient of 90.50%.
2. This study proposes a new viable paradigm for constructing refined segmentation network architectures for HR crack images in complex backgrounds. This paradigm replaces the decoder part of traditional encoder-decoder architecture segmentation models with a point-rendering head and introduces a detail-restoring boundary-guided branch and boundary point sampling strategy. This innovation allows the model to perform refined segmentation directly on HR crack images without additional computational resource requirements.
3. In the training phase, introducing a reasonable boundary dilation coefficient to expand the sampling range in the boundary sampling process of the rendering head can eliminate the biased guidance caused by the discrepancy between the real boundary and the labeled boundary due to subjectivity in manual annotations, significantly enhancing the model's robustness. This ensures the model does not rely solely on finely annotated data for training.

4. The probability heatmap-guided boundary rendering point sampling strategy proposed for the inference phase concentrates limited computational power from simple samples scattered in the background and crack body areas onto difficult boundary sample areas. This significantly improves the model's recognition accuracy of ambiguous coarse segmentation boundaries without additional computational resource consumption.

Deploying RLCSN on the UAV can significantly enhance the safety and efficiency of UAV-based bridge crack detection. Because RLCSN can directly perform high-precision inference on HR crack images, it alleviates issues like loss of crack detail information caused by image preprocessing in traditional low-resolution crack image inference processes, which is significant for promoting safe and efficient bridge crack detection using UAVs. In the future, the authors will explore the development of multimodal fusion technology capable of processing data such as ultrasonic, laser point clouds, and infrared images, thereby further enhancing the model's ability to accurately segment tiny cracks in complex backgrounds. Additionally, the authors intend to investigate advanced and complex supervised machine learning and pixel-level feature fusion algorithms, including Neural Dynamic Classification algorithm (Rafiei and Adeli, 2017), Dynamic Ensemble Learning Algorithm (Alam et al., 2020), Finite Element Machine for fast learning (Pereira et al., 2020), and self-supervised learning (Rafiei et al., 2022). These techniques aim to significantly improve the precision and robustness of rendering algorithms used for fine-grained crack segmentation in practical engineering tasks.

ACKNOWLEDGMENT

The authors would like to thank the editor, the seven anonymous reviewers, and Panfei Wu for their constructive comments and valuable suggestions, which were very beneficial to the improvement of this paper. This work was supported by the National Natural Science Foundation of China (Grant No. 52278177), the Hunan Provincial Innovation Foundation for Postgraduate (Grant No. QL20210106), and the China Scholarship Council (Grant No. 202206130068).

REFERENCES

Abdallah, A. M., Atadero, R. A. & Ozbek, M. E. (2022), A State-of-the-Art Review of Bridge Inspection Planning: Current Situation and Future Needs, *Journal of Bridge Engineering*, **27**(2), 03121001.

Alam, K. M. R., Siddique, N. & Adeli, H. (2020), A Dynamic Ensemble Learning Algorithm for Neural Networks, *Neural Computing and Applications*, **32**(12), 8675-8690.

Alipour, M., Harris, D. K. & Miller, G. R. (2019), Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks, *Journal of Computing in Civil Engineering*, **33**(6), 04019040.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M. & Farhan, L. (2021), Review of Deep Learning: Concepts, Cnn Architectures, Challenges, Applications, Future Directions, *Journal of Big Data*, **8**, 1-74.

Badrinarayanan, V., Kendall, A. & Cipolla, R. (2017), Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**(12), 2481-2495.

Bang, S., Park, S., Kim, H. & Kim, H. (2019), Encoder‐Decoder Network for Pixel‐Level Road Crack Detection in Black‐Box Images, *Computer‐Aided Civil and Infrastructure Engineering*, **34**(8), 713-727.

Barron, J. T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R. & Srinivasan, P. P. (2021), Mip-Nerf: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855-5864.

Bertasius, G., Shi, J. & Torresani, L. (2016), Semantic Segmentation with Boundary Neural Fields, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3602-3610.

Bertasius, G., Torresani, L., Yu, S. X. & Shi, J. (2017), Convolutional Random Walk Networks for Semantic Image Segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 858-866.

Borse, S., Wang, Y., Zhang, Y. & Porikli, F. (2021), Inverseform: A Loss Function for Structured Boundary-Aware Segmentation, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5901-5911.

Chambon, S. & Moliard, J.-M. (2011), Automatic Road Pavement Assessment with Image Processing: Review and Comparison, *International Journal of Geophysics*, **2011**.

Chen, D., Spencer, J., Mirebeau, J.-M., Chen, K., Shu, M. & Cohen, L. D. (2021), A Generalized Asymmetric Dual-Front Model for Active Contours and Image Segmentation, *IEEE Transactions on Image Processing*, **30**, 5056-5071.

Chen, J., Liu, Y. & Hou, J.-a. (2023), A Lightweight Deep Learning Network Based on Knowledge Distillation for Applications of Efficient Crack Segmentation on Embedded Devices, *Structural Health Monitoring*, **22**(5), 3027-3046.

Chen, Z., Yang, L., Lai, J.-H. & Xie, X. (2023), Cunerf: Cube-Based Neural Radiance Field for Zero-Shot Medical Image Arbitrary-Scale Super Resolution, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21185-21195.

Cheng, H. K., Chung, J., Tai, Y.-W. & Tang, C.-K. (2020), Cascadepsp: Toward Class-Agnostic and Very High-Resolution Segmentation Via Global and Local Refinement, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8890-8899.

Chu, H., Long, L., Guo, J., Yuan, H. & Deng, L. (2023), Implicit Function‐Based Continuous Representation for Meticulous Segmentation of Cracks from High‐Resolution Images, *Computer‐Aided Civil and Infrastructure Engineering*.

Chun, P. j., Izumi, S. & Yamane, T. (2021), Automatic Detection Method of Cracks from Concrete Surface Imagery Using Two‐Step Light Gradient Boosting Machine, *Computer‐Aided Civil and Infrastructure Engineering*, **36**(1), 61-72.

Cole, F., Genova, K., Sud, A., Vlasic, D. & Zhang, Z. (2021), Differentiable Surface Rendering Via Non-Differentiable Sampling, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6088-6097.

Deng, J., Singh, A., Zhou, Y., Lu, Y. & Lee, V. C.-S. (2022), Review on Computer Vision-Based Crack Detection and Quantification Methodologies for Civil Structures, *Construction and Building Materials*, **356**, 129238.

Ding, H., Jiang, X., Liu, A. Q., Thalmann, N. M. & Wang, G. (2019), Boundary-Aware Feature Propagation for Scene Segmentation, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6819-6829.

Ding, H., Jiang, X., Shuai, B., Liu, A. Q. & Wang, G. (2020), Semantic Segmentation with Context Encoding and Multi-Path Decoding, *IEEE Transactions on Image Processing*, **29**, 3520-3533.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D. & Houlsby, N. (2020), An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.

Ellenberg, A., Kontsos, A., Moon, F. & Bartoli, I. (2016), Bridge Related Damage Quantification Using Unmanned Aerial Vehicle Imagery, *Structural Control and Health Monitoring*, **23**(9), 1168-1179.

Fan, H. & Ling, H. (2017), Sanet: Structure-Aware Network for Visual Tracking, *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 42-49.

Guo, F., Liu, J., Lv, C. & Yu, H. (2023), A Novel Transformer-Based Network with Attention Mechanism for Automatic Pavement Crack Detection, *Construction and Building Materials*, **391**, 131852.

Hassanpour, A., Moradikia, M., Adeli, H., Khayami, S. R. & Shamsinejadbabaki, P. (2019), A Novel End‐to‐End Deep Learning Scheme for Classifying Multi‐Class Motor Imagery Electroencephalography Signals, *Expert Systems*, **36**(6), e12494.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep Residual Learning for Image Recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778.

Hu, D., Zhang, Z., Hou, T., Liu, T., Fu, H. & Gong, M. (2023), Multiscale Representation for Real-Time Anti-Aliasing Neural Rendering, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17772-17783.

Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. (2017), Densely Connected Convolutional Networks, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708.

Jeong, E., Seo, J. & Wacker, J. (2020), Literature Review and Technical Survey on Bridge Inspection Using Unmanned Aerial Vehicles, *Journal of Performance of Constructed Facilities*, **34**(6), 04020113.

Ke, T.-W., Hwang, J.-J., Liu, Z. & Yu, S. X. (2018), Adaptive Affinity Fields for Semantic Segmentation, *Proceedings of the European conference on computer vision (ECCV)*, pp. 587-602.

Kim, J., Shim, S., Cha, Y. & Cho, G.-C. (2021), Lightweight Pixel-Wise Segmentation for Efficient Concrete Crack Detection Using Hierarchical Convolutional Neural Network, *Smart Materials and Structures*, **30**(4), 045023.

Kirillov, A., Wu, Y., He, K. & Girshick, R. (2020), Pointrend: Image Segmentation as Rendering, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9799-9808.

Krähenbühl, P. & Koltun, V. (2011), Efficient Inference in Fully Connected Crfs with Gaussian Edge Potentials, *Advances in neural information processing systems*, **24**.

Lan, Z.-X. & Dong, X.-M. (2022), Minicrack: A Simple but Efficient Convolutional Neural Network for Pixel-Level Narrow Crack Detection, *Computers in Industry*, **141**, 103698.

Lee, H. J., Kim, J. U., Lee, S., Kim, H. G. & Ro, Y. M. (2020), Structure Boundary Preserving Segmentation for Medical Image with Ambiguous Boundary, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4817-4826.

Li, G., Liu, T., Fang, Z., Shen, Q. & Ali, J. (2022), Automatic Bridge Crack Detection Using Boundary Refinement Based on Real ‐ Time Segmentation Network, *Structural Control and Health Monitoring*, **29**(9), e2991.

Li, H., Wang, G., Lu, J. & Kiritsis, D. (2022), Cognitive Twin Construction for System of Systems Operation Based on Semantic Integration and High-Level Architecture, *Integrated Computer-Aided Engineering*, **29**(3), 277-295.

Li, X., Li, X., Zhang, L., Cheng, G., Shi, J., Lin, Z., Tan, S. & Tong, Y. (2020), Improving Semantic Segmentation Via Decoupled Body and Edge Supervision, *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, Springer, pp. 435-452.

Liang, J., Homayounfar, N., Ma, W.-C., Xiong, Y., Hu, R. & Urtasun, R. (2020), Polytransform: Deep Polygon Transformer for Instance Segmentation, *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9131-9140.

Liu, G., Ding, W., Shu, J., Strauss, A. & Duan, Y. (2023), Two-Stream Boundary-Aware Neural Network for Concrete Crack Segmentation and Quantification, *Structural Control and Health Monitoring*, **2023**.

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, *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012-10022.

Long, J., Shelhamer, E. & Darrell, T. (2015), Fully Convolutional Networks for Semantic Segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431-3440.

Martins, G. B., Papa, J. P. & Adeli, H. (2020), Deep Learning Techniques for Recommender Systems Based on Collaborative Filtering, *Expert Systems*, **37**(6), e12647.

Mei, Q., Gül, M. J. C. & Materials, B. (2020), A Cost Effective Solution for Pavement Crack Inspection Using Cameras and Deep Neural Networks, **256**, 119397.

Mohan, A. & Poobal, S. (2018), Crack Detection Using Image Processing: A Critical Review and Analysis, *Alexandria Engineering Journal*, **57**(2), 787-798.

Nhat-Duc, H., Nguyen, Q.-L. & Tran, V.-D. (2018), Automatic Recognition of Asphalt Pavement Cracks Using Metaheuristic Optimized Edge Detection Algorithms and Convolution Neural Network, *Automation in Construction*, **94**, 203-213.

Ni, F., Zhang, J. & Chen, Z. (2019), Zernike-Moment Measurement of Thin-Crack Width in Images Enabled by Dual-Scale Deep Learning, *Computer-Aided Civil and Infrastructure Engineering*, **34**(5), 367-384.

Ni, F., Zhang, J., Chen, Z. J. S. C. & Monitoring, H. (2019), Pixel ‐ Level Crack Delineation in Images with Convolutional Feature Fusion, **26**(1), e2286.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P. & Adeli, H. (2020), Fema: A Finite Element Machine for Fast Learning, *Neural Computing and Applications*, **32**, 6393-6404.

Quan, J., Ge, B. & Wang, M. (2023), Crackvit: A Unified Cnn-Transformer Model for Pixel-Level Crack Extraction, *Neural Computing and Applications*, 1-17.

Rafiei, M. H. & Adeli, H. (2017), A New Neural Dynamic Classification Algorithm, *IEEE transactions on neural networks and learning systems*, **28**(12), 3074-3083.

Rafiei, M. H., Gauthier, L. V., Adeli, H. & Takabi, D. (2022), Self-Supervised Learning for Electroencephalography, *IEEE transactions on neural networks and learning systems*.

Rafiei, M. H., Khushefati, W. H., Demirboga, R. & Adeli, H. (2017), Supervised Deep Restricted Boltzmann Machine for Estimation of Concrete, *ACI Materials Journal*, **114**(2), 237.

Reyad, M., Sarhan, A. M. & Arafa, M. (2023), A Modified Adam Algorithm for Deep Neural Network Optimization, *Neural Computing and Applications*, **35**(23), 17095-17112.

Ronneberger, O., Fischer, P. & Brox, T. (2015), U-Net: Convolutional Networks for Biomedical Image Segmentation, *International Conference on Medical image computing and computer-assisted intervention*, Springer, pp. 234-241.

Sacks, R., Kedar, A., Borrmann, A., Ma, L., Brilakis, I., Hüthwohl, P., Daum, S., Kattel, U., Yosef, R. & Liebich, T. (2018), Seebridge as Next Generation Bridge Inspection: Overview, Information Delivery Manual and Model View Definition, *Automation in Construction*, **90**, 134-145.

She, H.-C., Huang, L.-Y. & Duann, J.-R. (2023), A Shared Hippocampal Network in Retrieving Science-Related Semantic Memories, *International journal of neural systems*, 2350034-2350034.

Shen, T., Zhang, Y., Qi, L., Kuen, J., Xie, X., Wu, J., Lin, Z. & Jia, J. (2022), High Quality Segmentation for Ultra High-Resolution Images, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1310-1319.

Sun, Y. & Li, Y. (2022), Dice: Leveraging Sparsification for out-of-Distribution Detection, *European Conference on Computer Vision*, Springer, pp. 691-708.

Takikawa, T., Acuna, D., Jampani, V. & Fidler, S. (2019), Gated-Scnn: Gated Shape Cnns for Semantic Segmentation, *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5229-5238.

Tian, Y., Chen, C., Sagoe-Crentsil, K., Zhang, J. & Duan, W. (2022), Intelligent Robotic Systems for Structural Health Monitoring: Applications and Future Trends, *Automation in Construction*, **139**, 104273.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), Attention Is All You Need, *Advances in neural information processing systems*, **30**.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M. & Wang, X. (2020), Deep High-Resolution Representation Learning for Visual Recognition, *IEEE transactions on pattern analysis and machine intelligence*, **43**(10), 3349-3364.

Wang, W. & Su, C. (2021), Deep Learning-Based Real-Time Crack Segmentation for Pavement Images, *KSCE Journal of Civil Engineering*, **25**(12), 4495-4506.

Wang, Y., Zhao, X., Li, Y. & Huang, K. (2018), Deep Crisp Boundaries: From Boundaries to Higher-Level Tasks, *IEEE Transactions on Image Processing*, **28**(3), 1285-1298.

Xiang, C., Guo, J., Cao, R. & Deng, L. (2023), A Crack-Segmentation Algorithm Fusing Transformers and Convolutional Neural Networks for Complex Detection Scenarios, *Automation in Construction*, **152**, 104894.

Xie, X., Cai, J., Wang, H., Wang, Q., Xu, J., Zhou, Y. & Zhou, B. (2022), Sparse ‐ Sensing and Superpixel ‐ Based Segmentation Model for Concrete Cracks, *Computer ‐ Aided Civil and Infrastructure Engineering*, **37**(13), 1769-1784.

Xu, D., Ouyang, W., Wang, X. & Sebe, N. (2018), Pad-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 675-684.

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X. & Ling, H. (2019), Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection, pp. arXiv:1901.06340.

Yeum, C. M. & Dyke, S. J. (2015), Vision-Based Automated Crack Detection for Bridge Inspection, *Computer-Aided Civil and Infrastructure Engineering*, **30**(10), 759-770.

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G. & Sang, N. (2018), Bisenet: Bilateral Segmentation Network for Real-Time Semantic Segmentation, *Proceedings of the European conference on computer vision (ECCV)*, pp. 325-341.

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G. & Sang, N. (2018), Learning a Discriminative Feature Network for Semantic Segmentation, *Proceedings* *of the IEEE conference on computer vision and pattern recognition*, pp. 1857-1866.

Yuan, Y., Xie, J., Chen, X. & Wang, J. (2020), Segfix: Model-Agnostic Boundary Refinement for Segmentation, *European Conference on Computer Vision*, Springer, pp. 489-506.

Zhang, X., Zhou, X., Lin, M. & Sun, J. (2018), Shufflenet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848-6856.

Zhang, Z. & Sabuncu, M. (2018), Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels, *Advances in neural information processing systems*, **31**.

Zhao, H., Shi, J., Qi, X., Wang, X. & Jia, J. (2017), Pyramid Scene Parsing Network, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881-2890.

Zhou, Z., Zhang, J. & Gong, C. (2023), Hybrid Semantic Segmentation for Tunnel Lining Cracks Based on Swin Transformer and Convolutional Neural Network, *Computer ‐Aided Civil and Infrastructure Engineering*.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q. & Wang, S. (2019), Deepcrack: Learning Hierarchical Convolutional Features for Crack Detection, *IEEE Transactions on Image Processing*, **28**(3), 1498-1512.