Received: 6 June 2025 | Accepted: 24 September 2025

DOI: 10.1111/mice.70088

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Refined segmentation of high-resolution bridge crack images via probability map-guided point rendering technique

Honghu Chu[1,2,3] | **Weiwei Chen**[2] | **Lu Deng**[3]

[1]College of Civil Engineering and Architecture, Zhejiang University of Water Resources and Electric Power, Hangzhou, China

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

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

Correspondence
Lu Deng, College of Civil Engineering, Hunan University, Changsha 410082, China.
Email: denglu@hnu.edu.cn

Weiwei Chen, Bartlett School of Sustainable Construction, University College London, London WC1E 7HB, UK.
Email: weiwei.chen@ucl.ac.uk

Funding information
Horizon Europe Project, D-HYDROFLEX, Grant/Award Number: 101122357; Horizon Europe Project, INHERIT, Grant/Award Number: 101123326; National Natural Science Foundation of China, Grant/Award Number: 52278177; National Key Research and Development Program of China, Grant/Award Number: 2023YFC3806800; China Scholarship Council, Grant/Award Number: 202206130068

Abstract

High-resolution (HR) imaging technology is increasingly employed to capture crack images in civil infrastructure, which is vital for ensuring the safety of the bridge inspection process conducted via unmanned aerial vehicles (UAVs). Such applications require the development of advanced algorithms for the segmentation of HR images. Traditional deep learning-based segmentation methods for inferencing HR images consume considerable GPU resources, which prompts the authors to draw inspiration from the cost-effective rendering technique in computer graphics and try to apply this advanced method to the refined segmentation of HR crack images. However, the original rendering method, designed to guide rendering points by the coarse segmentation masks, often inadequately directs rendering points towards the crucial boundary areas of tiny cracks, leading to unclear or missing boundary predictions. To address this, an innovative rendering technique was proposed, utilizing probability maps to precisely direct rendering points towards crack boundaries and tiny-crack branches during inference. This method enhances the accuracy of crack boundary segmentation and reduces the miss rate of tiny crack branches from HR images, all while conserving computational resources. Through model parameter experiments and ablation studies, the optimal model was obtained, and the effectiveness of the improved components was demonstrated. Furthermore, the field test has confirmed that, equipped with the proposed point rendering technique, the UAV is permitted to effectively perform crack inspection within a 3-m distance from the main beam. Compared to traditional low-resolution semantic segmentation methods, the UAV bridge inspection time is significantly reduced by 50% while maintaining the same accuracy.

1 | INTRODUCTION

Crack formation is a predominant contributor to various bridge pathologies, potentially diminishing the structural integrity and risking catastrophic failures (Rafiei & Adeli, 2018; Yamaguchi & Mizutani, 2024). Consequently, it is imperative for traffic management entities to promptly and precisely detect such impairments,

safeguarding bridge functionality throughout its lifespan (Siriborvornratanakul, 2023). Deep learning (DL)-based semantic segmentation techniques emerge as the leading approach for swift and automated crack detection, occupying a central role in contemporary bridge maintenance research (Rafiei & Adeli, 2016; Rafiei et al., 2017).

In recent years, advancements in imaging technology and rising detection standards have catalyzed the adoption of high-resolution (HR) imaging devices for capturing engineering structure surface cracks. HR images offer the advantage of encapsulating more detailed crack information, enhancing structural safety assessments. Specifically, for unmanned aerial vehicle (UAV)-based bridge crack inspection, the use of HR imaging equipment ensures the captured crack images are clear. Such clarity allows for a significant increase in the operational distance between the UAV and the bridge, which is essential for maintaining the UAV's safety during the whole inspection process (Ma & Hartmann, 2023; Wu et al., 2023). However, conventional DL-based crack segmentation frameworks frequently necessitate intricate convolutional processes to interpolate new pixels in the decoding phase (Liu et al., 2022a) employing deconvolution or transposed convolution techniques. These processes involve handling an expanded data volume correlating with image size, accumulating extensive computational intermediates in graphics processing unit (GPU) memory (Shen et al., 2022). To ensure the smooth execution of crack segmentation algorithms on standard commercial GPUs with HR inputs, it becomes imperative to implement compromise preprocessing measures due to the limitation of GPU memory capacity (Ji et al., 2023; Shen et al., 2022). Unfortunately, these compromises lead to irreversible degradation in segmentation accuracy, especially at the edges and in the finer details. Such degradation significantly hinders the ability of bridge maintenance personnel to accurately assess structural safety and formulate appropriate maintenance strategies.

In fact, to address the aforementioned challenges, the authors, as pioneers in the field of fine-grained recognition of HR crack images, have recently conducted a series of targeted studies, aiming to gradually improve the model's fine-grained segmentation performance and lightweight deployment capability when applied to HR bridge crack images captured by UAVs, collectively establishing a progressive framework for addressing the challenges in this field. The first contribution, Tiny-Crack-Net (Chu et al., 2022), introduced a multi-scale feature fusion network with dual attention mechanisms to enhance the segmentation of tiny cracks at conventional image resolutions, addressing the challenge of fine-grained recognition of sparse patterns. Building on this foundation, the second work (Chu & Chun, 2024) proposed a multiscale cascaded operation logic architecture that gradually extends the fine-grained recognition capability of Tiny-Crack-Net to high-resolution crack images, bridging the gap between tiny crack detection and large-scale scene deployment from both global and local perspectives. Subsequently, the authors improved the cascaded operation logic architecture by designing implicit functions to continuously model the irreversible loss of crack details caused by discrete progressive sampling (Chu et al., 2024a), further enhancing the fine-grained recognition performance. To meet the growing demand for real-time deployment, the authors published a fourth related study earlier this year (Chu et al., 2025), introducing a lightweight network based on boundary-guided branches called RLCSN. This network significantly reduces the computational load by unevenly guiding boundary power, paving the way for practical applications on UAV platforms. However, although RLCSN represents the forefront of high-resolution crack segmentation, its reliance on task-specific customization and separate training for the super-resolution edge-guided branches with uneven computational power significantly reduces the model's transferability in practical engineering applications. Therefore, the authors continue to explore the development of precise, efficient, and universally applicable methods for fine-grained segmentation of high-resolution crack images.

Diverging from conventional DL-based crack segmentation frameworks, PointRend, introduced by Kirillov et al. (2020), innovatively employs lightweight multilayer perceptrons (MLPs) that share weights for pointwise rendering (Tolstikhin et al., 2021), supplanting the traditional decoding schema. This MLP architecture uniquely computes each pixel independently, allowing the network to process pixels individually or in small batches rather than enlarging the entire feature map simultaneously as in standard upsampling. This approach markedly lessens the segmentation network's reliance on GPU memory, ensuring that memory requirements do not escalate with the inference image's dimensions, thus effectively preventing the irreversible loss of crack edge details caused by compromise preprocessing methods such as cropping or downsampling (Lu et al., 2022).

Contrastingly, the MLP-based pointwise rendering, as an alternative to traditional approaches of mitigating GPU memory constraints through downsizing HR images or dissecting them into patches for patchwise inference, facilitates computations at the native resolution. This strategy circumvents the detail diminution associated with resizing and the irregularities or stitching artifacts

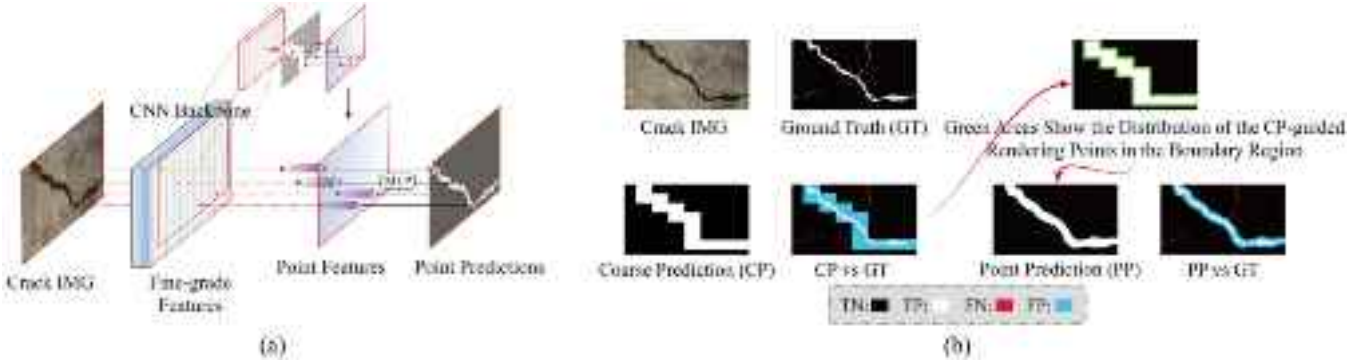

FIGURE 1 The segmentation process of a randomly selected high-resolution (HR) crack image using the original PointRend architecture. (a) provides a schematic representation of the PointRend architecture; (b) displays the coarse segmentation guiding the rendering points and the final prediction outcome influenced by this guidance.

emerging in the reassembly phase, thereby bolstering the uniformity and integrity of prediction outcomes. In essence, PointRend emerges as the premier method for overcoming GPU memory limitations in HR image segmentation at present.

However, the rendering technique-based PointRend, originally tailored for standard objects in natural scenes, faces limitations when applied to crack segmentation due to cracks' elongated topologies and random distribution, which markedly diverge from those of standard objects. Specifically, in PointRend's conventional use during the decoding phase, the method relies on coarse segmentation masks to direct rendering points for enhanced prediction. This coarse segmentation masks-guided approach works well for typically sized objects, as their edge details can be reasonably captured from these coarse masks. However, this guidance strategy is not the case for elongated and randomly distributed objects like cracks, where significant detail, including edges and tiny branches, is lost in the coarse masks. Consequently, rendering points struggle to concentrate on these ambiguous or missing details, leading to vague crack boundary predictions and overlooked tiny branches. Illustrations in Figure 1 demonstrate the ineffectiveness of coarse masks in guiding rendering points towards intricate crack details, with rendering points often misdirected to imprecisely defined coarse edges instead of the actual fine branches. This misalignment not only omits crucial crack details but also impedes the achievement of precise edge predictions.

This study presents an advanced rendering technique aimed at maintaining the GPU memory efficiency inherent in point-rendering methods for segmenting HR images, while enhancing precision in identifying crack boundaries and minuscule branches. Utilizing a probability map-based strategy for directing rendering points, this method recalibrates the delineation between simple and complex samples on the probability map. Consequently, the focus transitions from less intricate areas, such as backgrounds and primary crack bodies, to more challenging zones like crack peripheries and slender branches. This approach enables the proposed network to achieve meticulous segmentation of HR crack images with standard commercial GPUs. The contributions are summarized as follows:

1. For the first time, this study integrates point-rendering technology into HR crack image segmentation, addressing the significant challenges of high GPU computational resource dependence and the lack of fine-grained segmentation detail inherent in traditional methods.
2. The authors have developed two distinct rendering point guidance strategies tailored for both the training and inference phases of the proposed model. This approach not only simplifies the model training process but also markedly improves the accuracy of segmentation in critical areas, including crack edges and minor branches.
3. This study proposes a novel paradigm for constructing networks for fine-grained segmentation of HR crack images. It advocates for the replacement of traditional prediction heads with those driven by the probability map and incorporates the multiscale transformer architecture for enhanced crack detail extraction.

Experiments were conducted on both an open-source HR crack image dataset and the HR crack images collected on-site with the unmanned aerial vehicle (UAV), demonstrating that the proposed model achieves state-of-the-art performance in both quantitative and qualitative outcomes.

2 | LITERATURE REVIEW

2.1 | DL-based crack identification

DL is increasingly employed to tackle infrastructure challenges, notably in automated crack detection (Dick et al., 2019; Konstantakopoulos et al., 2019). Utilizing image classification for categorizing and assessing structural cracks, subsequent developments in object detection algorithms have enhanced structural crack identification, underscoring DL's pivotal role in infrastructure diagnostics (Flah et al., 2021). To accurately identify the specific morphological features of cracks (i.e. direction, edges, and corners), DL-based semantic segmentation algorithms have been introduced into the field of crack image recognition. Initially, FCN and SegNet architectures were used as baselines for constructing crack semantic segmentation architectures (De Nardin et al., 2023; Diaz-Frances et al., 2024). However, these two architectures, due to performing multiple discrete downsampling operations during the feature extraction process, lead to the loss of a large amount of fine crack details in the segmentation results (Zhang et al.,2019a). Subsequently, the U-shaped encoder–decoder architecture improved this issue to a certain extent by introducing skip connections and gradually became the mainstream architecture for crack segmentation (Ronneberger et al., 2015). A series of U-shaped encoder–decoder architectures and their variants have been proposed to perform high-precision recognition of crack images in different scenarios (Huyan et al., 2020; Mei et al., 2020). Chow et al. (2020) leverage the convolutional autoencoder for crack detection in concrete structures, illustrating the capacity of DL to facilitate the visual inspection of civil infrastructure without the need for labeled data, thereby streamlining the identification process of defects. Similarly, Chen et al. (2021a) employ Hessian structure propagation to differentiate pavement cracks from complex textures and noise, showcasing the method's precision and recall in handling various crack types and conditions. However, such U-shaped autoencoder architectures, being predominantly built on CNN feature extraction backbones, are limited by CNNs' inability to model explicit long-range relationships (Khan et al., 2022; Zeng et al., 2024), making it challenging to accurately segment slender, fine cracks. For this reason, some researchers have replaced CNNs with Transformer architectures equipped with self-attention mechanisms for crack feature extraction (Guo et al., 2023; Shamsabadi et al., 2022). Shang et al. (2023) introduce the defect-aware transformer network (DAT-Net) for surface crack inspection, utilizing transformer to model long-range dependencies, thus outperforming traditional convolution-based methods. By further applying the advantages of transformer architecture in global information modeling, Ma et al. (2024) proposed a feature-level domain disentanglement and randomization framework designed to enhance the generalization capability of DL models in rail surface crack segmentation tasks, showcasing the framework's robustness in unseen scenarios. However, the aforementioned methods treat crack boundary and crack main body pixels equally in the prediction head, resulting in insufficient allocation of computational resources to areas of difficult samples (i.e. crack boundary pixels), thereby leading to ambiguous prediction results at the edges of the predicted masks.

2.2 | Conventional fine-grained segmentation

To improve the quality of image segmentation results, fine-grained segmentation techniques were proposed and explored. Initially, traditional methods included the integration of conditional random fields (Alam et al., 2020; Zheng et al., 2015) and graph models (Dias & Medeiros, 2018) with deep neural networks. These methods primarily rely on low-level color boundaries, often failing to effectively incorporate high-level semantic information, thus failing to repair contour areas with small color contrast differences from the background. For these hard-to-recognize contour areas, some researchers designed custom refinement modules (Peng et al., 2017; Zhang et al.,2019b), but the need for manually setting thresholds before making predictions limited the widespread application of such methods. This limitation directly led subsequent research towards the development of universal plug-in modules. Representative studies include: RefineNet proposed by Lin et al. (2017), which performs multi-path boundary refinement during the upsampling process within the network by connecting feature units of different levels; and HRNet proposed by Wang et al. (2020), employs multi-path connections rather than physical cascading mechanisms. This approach effectively captures detailed information in images by leveraging HR representation learning, thereby enhancing the network's ability to fully comprehend and interpret the structure of HR images. However, the multi-path redundant design adopted by the aforementioned methods necessitates reliance on substantial GPU computational resources during inference, making it difficult to perform refined inference for HR crack images. Moreover, the large number of parameters also poses a challenge to the effective training of the model. Recently, Chu et al. (2024b) presented cascade-FcaHRNet, a multiscale framework designed for segmenting HR cracks in bridge structures. The cascade-FcaHRNet showcases the capability of HR representation learning techniques to discern intricate

details by integrating a frequency-channel attention mechanism, thus facilitating precise analysis of 4K resolution crack images. However, the segmented approach to cascading operations within cascade-FcaHRNet disrupts the continuity among global pixels, potentially compromising accuracy. Furthermore, the layered structure of cascade operations elevates the computational resource demands during the analysis phase.

2.3 | Rendering-based fine-grained segmentation

In fact, the issue of ambiguous boundary areas faced by crack image segmentation is similar to the problem of jagged edges encountered in computer graphics when images or graphics are pixelated on display devices (Barron et al., 2021; Wu et al., 2021). Computer graphics primarily focuses on how to generate and render images, including techniques such as image synthesis, animation, virtual reality, and scene rendering, which are widely used in the gaming and film special effects industries, representing to some extent the state-of-the-art in computer vision community (Kellnhofer et al., 2021; Tewari et al., 2022). Inspired by rendering techniques in computer graphics, Kirillov et al. (2020) proposed PointRend, which successfully applies rendering technology to perform refined segmentation for regular objects in natural scenes. Due to the use of MLP with shared weights to continuously perform pixel predictions in a point-wise manner, PointRend leverages the advantages of rendering technology in detailed representation without additionally increasing the network's dependence on GPU memory, making it feasible for use on commercial GPUs for refined inference on HR images. However, as described in the introduction, the rendering points during inference by PointRend are guided by coarse segmentation masks, which cannot meet the requirements for slender targets like cracks that require detailed detection. To retain the advantages of point-wise rendering methods in terms of GPU memory usage while ensuring the accuracy of inference results on crack edges and their tiny branches during HR crack image segmentation, this study designed a refined rendering strategy. The customized strategy employs guidance based on the probability map to direct the precise inference process of rendering points.

3 | METHODOLOGY

The probability map-guided point-rendering crack segmentation network proposed in this study follows the encoder-decoder architecture design pattern, consisting of three main parts: a crack fine-grained feature encoding backbone, a rendering point guidance branch, and a point-rendering based fine-grained prediction head. The crack fine-grained feature encoding backbone is built from a lightweight encoder and a series of transformer blocks; the rendering point guidance branch shares a lightweight encoder with the crack fine-grained feature encoding backbone and generates probability maps for guiding rendering points through a traditional decoder; the point-rendering based fine-grained prediction head is built based on the MLP that can perform pointby-point refinement predictions. It should be noted that the rendering point guidance branch and the point-rendering based fine-grained prediction head together constitute the network's decoder part. Figure 2 visually presents the algorithmic details and computational logic of the proposed framework for HR crack image fine-grained segmentation.

3.1 | Crack fine-grained feature encoding backbone

This research introduces a novel feature extraction encoder to capture detailed crack information in deep semantic feature maps while enhancing inference efficiency. It integrates a lightweight encoding framework with an enhanced pyramid vision transformer (PVT) (Wang et al., 2021). Initially, images undergo processing through the MobileNetV3 architecture (Howard et al., 2019), employing depthwise separable convolutions for efficient feature filtering and fusion, thus maintaining crucial image details with reduced model complexity.

Subsequent processing involves advancing these feature maps through a refined PVT module. This module, leveraging a hierarchical transformer architecture, allows for scalable feature map processing, facilitating the extraction and integration of multi-scale features. It employs a self-attention mechanism at each layer, capturing global contextual information to improve image detail interpretation. As processing progresses, the PVT module enhances the resolution of feature maps, culminating in HR, semantically dense outputs. Figure 2b illustrates the PVT's computational mechanics, featuring patch embedding and transformer encoding layers. Notably, to manage the computational demands of HR image predictions and prevent GPU memory overload, the conventional multi-head attention (MHA) layer is replaced with a spatial reduction attention (SRA) layer. This adjustment allows for pre-attention spatial reduction of key (K) and value (V), significantly boosting computational efficiency while maintaining the input structure with query (Q), key (K), and value (V). The implementation of SRA is

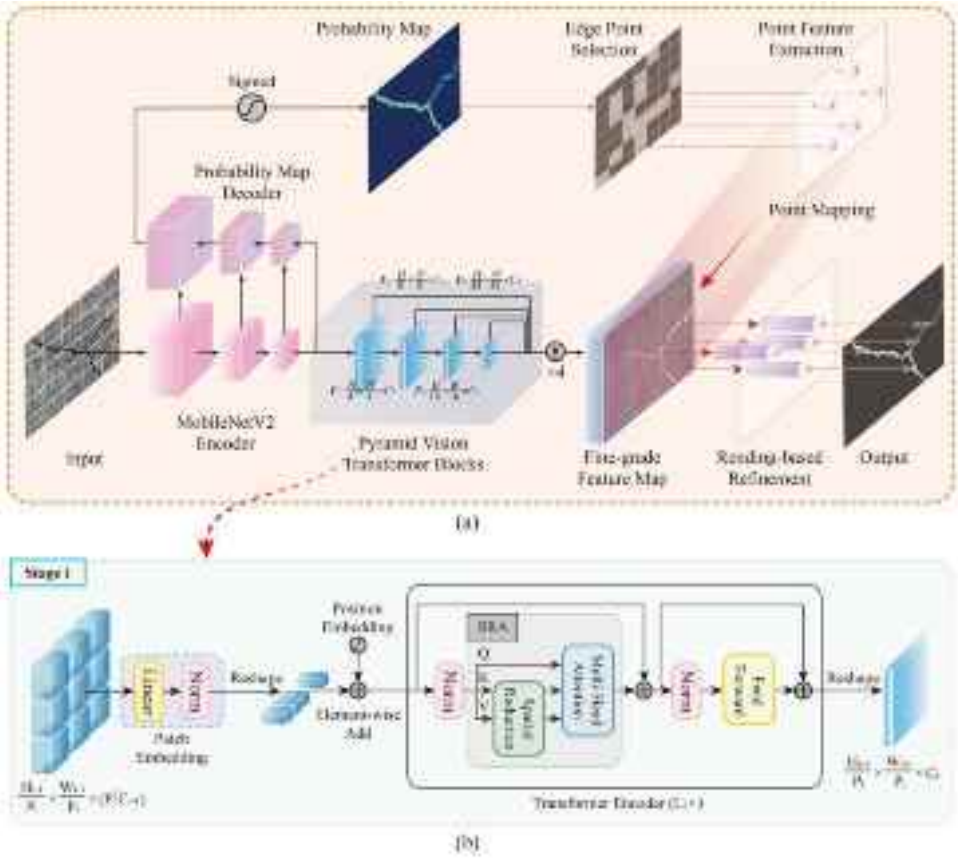

FIGURE 2 Schematic of the probability map-guided point-rendering architecture. (a) the comprehensive architecture; (b) the transformer block integrated with the Spatial Relationship Attention (SRA) module.

detailed through specific equations, underscoring its role in optimizing processing for HR image predictions.

$$\text{SRA}\ (Q,K,V) = \text{Concat}\left(\text{head}_0,\cdots,\text{head}_{N_i}\right)W^O \quad (1)$$

$$\text{head}_j = \text{Attention}\left(QW_j^Q, SR(K)W_j^K, SR(V)W_j^V\right) \quad (2)$$

where Concat$(\cdot)$ is the concatenation operation consistent with the literature (Shen et al., 2022). $W_j^Q \epsilon \mathbb{R}^{C_i \times d_{\text{head}}}$, $W_j^K \epsilon \mathbb{R}^{C_i \times d_{\text{head}}}$, $W_j^V \epsilon \mathbb{R}^{C_i \times d_{\text{head}}}$, and $W^O \epsilon \mathbb{R}^{C_i \times C_i}$ are linear projection parameters. N_i is the number of heads in the attention layer at the ith stage. Hence, the dimension of each head (i.e. d_{head}) is equal to $\frac{C_i}{N_i}$. SR$(\cdot)$ represents the spatial reduction operation that reduces the dimensions for K and V, which can be represented by Equation (3):

$$SR\ (x) = \text{Norm}\left(\text{Reshape}\left(x, R_i\right)W^S\right) \quad (3)$$

where $x \epsilon \mathbb{R}^{(H_i W_i) \times C_i}$ represents the input sequence, R_i denotes the reduction ratio of the attention layer at the ith stage. Reshape(x, R_i) is responsible for adjusting the input sequence x into a sequence of size $\frac{H_i W_i}{R_i^2} \times (R_i^2 C_i)$. $W^S \epsilon \mathbb{R}^{(R_i^2 C_i) \times C_i}$ represents the linear projection operation that reduces the dimension of the input sequence to C_i. Norm$(\cdot)$ denotes layer normalization. Similar to the original transformer, the attention operation of this study, Attention$(\cdot)$, can be represented by Equation (4):

$$\text{Attention}\ (Q,K,V) = \text{Softmax}\left(\frac{QK^{\intercal}}{\sqrt{d_{\text{head}}}}\right)V \quad (4)$$

Through these operations, the computational demand for self-attention operations within the pyramid transformer is reduced to $\frac{1}{R_i^2}$ of the traditional multi-head attention (MHA), thus allowing for the processing of HR images with lengthy sequential features under relatively limited computational resources.

Specifically, during feature sampling, a HR crack input image $X \epsilon R^{H \times W \times C}$ is first orderly cropped into $N = \frac{H}{L} \times \frac{W}{L}$ local image patch blocks (where L is set to 8 in this study). Subsequently, the orderly arranged local image patch blocks are input into an encoding architecture based on MobileNetV3 for flattening. The flattened features are passed to a linear embedding layer with output dimension D_k, resulting in the original embedding sequence $e \epsilon R^{N \times D_k}$. To ensure the transformer can effectively utilize

spatial prior information, a learnable position embedding of the same dimension is added to $e \epsilon R^{N \times D_k}$, producing a sequence with position embeddings $z^0 \epsilon R^{N \times D_k}$. z^0 is then input into a transformer segmentation architecture containing L layers of SRA to obtain a feature map F_1 with dimensions $\frac{H}{4} \times \frac{W}{4} \times C_1$. Similarly, using the feature map from the previous stage as input, the feature maps F_2, F_3, and F_4 for the subsequent three stages can be obtained. Compared to the original input image, the respective strides of the three feature maps are 8, 16, and 32. Finally, the deep semantic features from four scales, containing global long-distance dependencies of slender cracks, are concatenated together to enhance the network's representation of slender and small-sized cracks in the image.

In summary, the pyramid visual transformer enables the deep feature extraction of HR images by cropping them into low-resolution local image patches with positional associations and simplifies the computational load of the self-attention mechanism through spatial dimension reduction. This significantly reduces the model's reliance on high GPU memory during HR image inference. Moreover, owing to the distinct advantages of the pyramid structure and the Transformer's inherent self-attention mechanism in representing multi-scale features and modeling long-distance dependencies of global features respectively, the encoding architecture's ability to represent slender and minute cracks is greatly enhanced.

3.2 | Point rendering-based decoder

To fully decode the fine-grained crack information in the deep semantic feature maps, the authors improved upon the original PointRend by proposing a decoder composed of a point-rendering based fine-grained prediction head and a rendering point guidance branch. For the fine-grained prediction head, the same architecture as PointRend was utilized—that is, a point-by-point rendering structure built on the MLP. This is because, compared to traditional CNNs, using the MLP in rendering technology offers computational efficiency advantages due to weight sharing and point-wise prediction. Regarding the rendering point guidance branch, considering that the rendering point guidance strategy in the PointRend architecture was designed for traditional large-scale natural scene objects, which is not suitable for objects with slender topological structures like tiny cracks, the guidance strategy used in the branch was specifically customized.

Specifically, the authors designed a probability map-based rendering point guidance strategy for the model inference, aimed at directing computational resources efficiently towards precise delineation of crack boundaries and tiny crack branches. As illustrated in Figure 2a, the probability map is mainly obtained through forward propagation involving a lightweight encoder and two convolutional blocks, without the need for thresholding or other post-processing steps to determine the final class assignments. Therefore, compared to the coarse segmentation masks required for rendering point guidance in the original PointRend architecture, the acquisition process of the probability map is more direct and computationally efficient. More importantly, the probability map can reflect the probability of pixels belonging to each category, rather than simply categorizing them like in coarse segmentation masks, which means this probability information provides a continuous confidence measurement for each pixel instead of categorizing pixels in a binary manner, thus preserving more uncertainty and subtle differences about image areas, which is crucial for understanding the nuances and complex scenes within images. Due to the fact that the crack segmentation task is essentially a binary classification task, in which the pixels to be classified include two simple samples, namely background and crack body, as well as a difficult sample type located at boundaries and tiny crack branches, this study divides the regions on the probability map into three parts. Namely, the region with a probability close to 0 is considered relatively certain background pixels, the region with a probability close to 1 is considered relatively certain crack areas, and the intermediate region where the probability fluctuates around 0.5 is difficult for the model to determine accurately. These intermediate regions often consist of pixels with intensity or contrast levels between cracks and the background and are primarily concentrated around crack boundaries and tiny crack branches. Based on the aforementioned division principles, only areas on the probability map where probabilities fluctuate around 0.5 are subjected to refined rendering point sampling during inference. As for areas on the probability map with probabilities close to 0 and 1, which are considered easily identifiable pixels, they are directly mapped as background pixels and crack pixels on the prediction mask without redundant computations, considering the low complexity for recognition and to reduce the computational cost. It's noteworthy again that the reason for not adopting the coarse segmentation guidance from the original PointRend is because the multiple downsampling required in the acquisition phase of coarse segmentation leads to a significant loss of tiny cracks and crack edge details in crack images, whereas the probability map undergoes only one downsampling on the original image, retaining crack details as much as possible while also consuming less computational resources. To visually demonstrate the refined rendering point sampling method during the inference phase, Figure 3 visualizes a randomly selected probability map example. It's evident that on

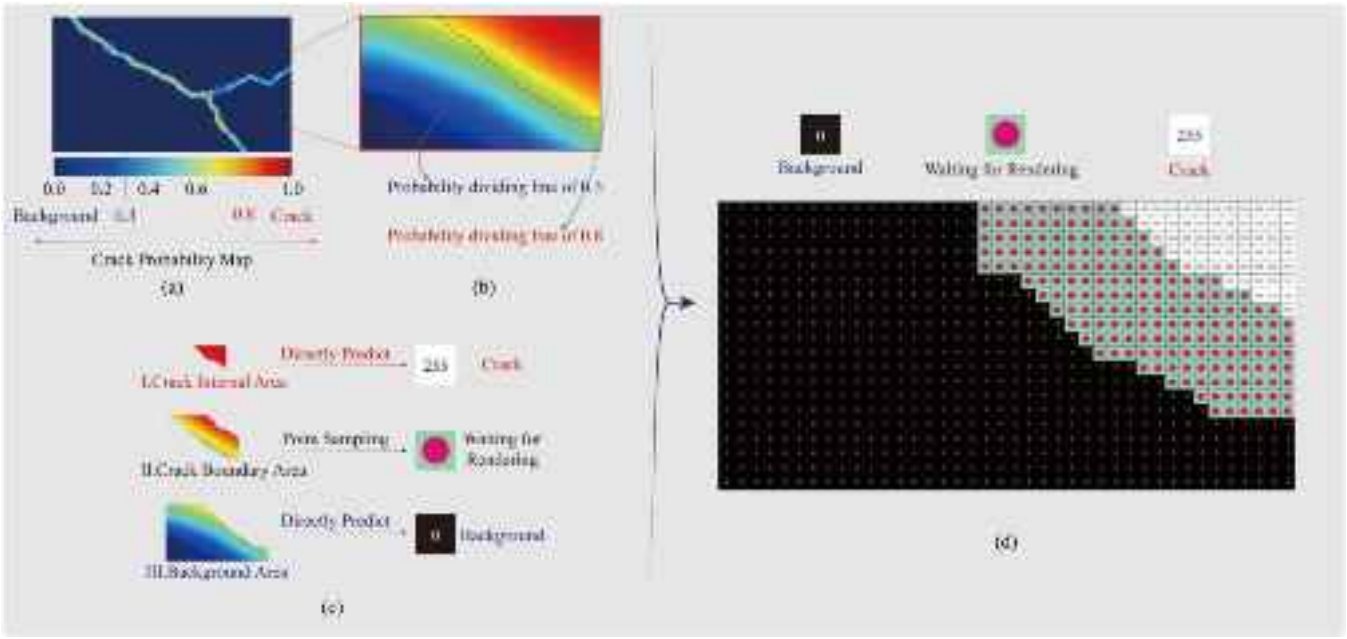

FIGURE 3 Visualization of the probability map-based rendering points sampling performed on the deep semantic feature map. (a) the probability map of a randomly selected crack image; (b) the local details of the probability map and the visualization of the probability intervals established in this study; (c) the operations faced by pixels in different probability intervals on the deep semantic feature map; (d) the prediction results of each area of the crack image divided by probability interval.

the probability map, probabilities in the background and crack main body areas concentrate around 0 and 1 respectively, while in the boundary areas, due to issues such as annotation errors and insignificant color differences, pixel probabilities on the probability map fluctuate around 0.5. This study sets the probability interval for these hard-to-identify pixels between 0.3–0.8, and in the subsequent refined rendering phase, only hard-to-identify samples with probabilities between 0.3–0.8 undergo refined inference. The parameter settings for the rendering point guidance during the training and inference phases will be detailed in Section 4.2.

4 | EXPERIMENTS

4.1 | Datasets

Adequate crack image data is a prerequisite for training and evaluating the proposed method. There are already numerous publicly available datasets for evaluating the performance of crack segmentation models on different civil infrastructure, including Stone331 (Zou et al., 2019), Bochum Crack DataSet (Çelik & König, 2022), Deepcrack537 (Liu et al., 2022b), CRKWH100 (Quan et al., 2023), and more. Due to the difficulties in collecting and finely annotating crack images, each crack image dataset tends to be smaller in scale and image size compared to natural scene image segmentation datasets, with the number of images in a single dataset often not exceeding 500, and the size of crack images not exceeding 600 × 800. Insufficient training samples can easily lead to model overfitting on data of a single distribution type, while relatively low-resolution training data can reduce training difficulty and decrease the model's dependence on computational resources. To ensure the model's robustness to diverse crack images in real-world scenarios, the authors used a mix of crack images from the Aigle-RN, CrackTree260, and Crack500 datasets (Ai et al., 2023) for model training and preliminary evaluation. To ensure that crack images from different sources could be smoothly input into the model for training, all images were resized to a uniform 256 × 256 pixels. Ultimately, a total of 800 resized crack images from the three open-source datasets were divided into three groups (500 for training, 150 for validation, 150 for testing) for model training and preliminary performance evaluation.

Given that higher-resolution images necessitate more substantial downsampling to fit within the constraints of limited GPU memory resources during inference, they are more susceptible to ambiguous predictions in boundary areas compared to their low-resolution counterparts. This requires HR images to undergo more meticulous refinement and repair processes. Consequently, the authors curated a HR crack image dataset to thoroughly evaluate model performance, as illustrated in Figure 4. Specifically,

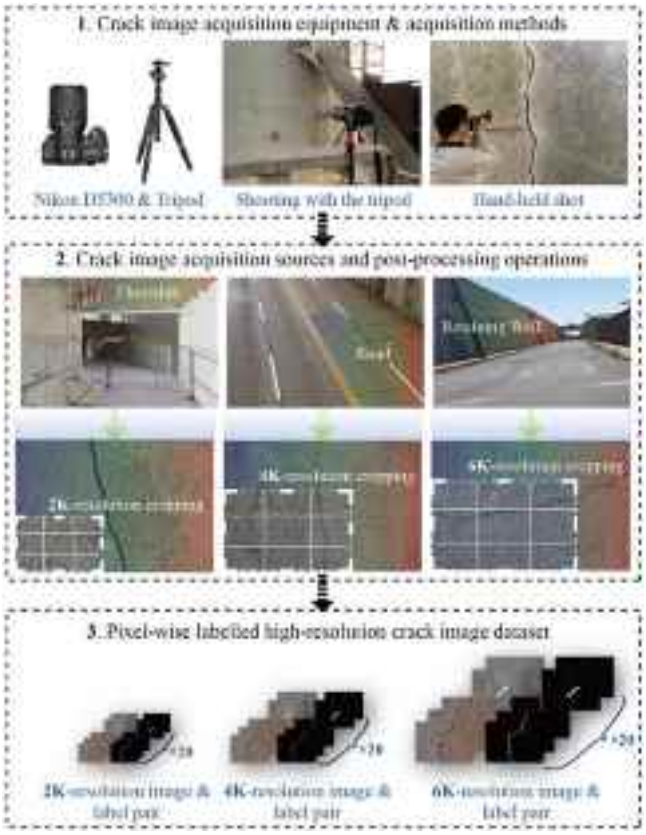

FIGURE 4 The establishment of the HR crack data set with refined annotation.

the dataset was enriched with images from three distinct structures—floor slabs, roads, and walls—located within the urban area of Changsha. These high-definition crack images were captured using both a tripod and a hand-held Nikon D5300 camera. A total of 300 high-definition crack images were collected in 6K RAW format. To mitigate the adverse effects on model performance stemming from imbalanced ratios of positive to negative samples, the authors selectively cropped the crack regions from the original high-definition images, resulting in 20 images each at resolutions of 2K, 4K, and 6K. Subsequently, all the cropped images were meticulously annotated.

It is important to note that while the network was trained entirely on low-resolution crack images, its design—featuring a multiscale pyramid structure—enables it to robustly adapt to different image resolutions during inference. This allows the model to directly process and accurately segment high-resolution crack images without requiring additional fine-tuning or retraining. The construction of the HR crack image dataset in this study serves primarily for evaluation purposes, aiming to validate the model's effectiveness under real-world high-resolution inspection conditions. Thus, although the dataset includes pixel-level annotations, they are not used for network training. This design choice not only circumvents the high manual cost of HR data annotation for training but also demonstrates the model's strong generalizability across image scales. The experimental results in Section 4 further support the effectiveness of this strategy.

4.2 | Evaluation method

To quantitatively assess the experimental outcomes, two widely adopted metrics, intersection over union (IoU) and the dice similarity coefficient (Dice), were utilized. Additionally, to emphasize the efficacy of the proposed method in delineating boundaries, the mean boundary accuracy (mBA) metric, introduced in CascadePSP (Cheng et al., 2020), was employed. The mBA metric fundamentally calculates the IoU between the ground truth (GT) and the predicted mask specifically within boundary regions, with Figure 5 providing a visual representation of its calculation procedure.

4.3 | Implementations

4.3.1 | Hardware equipment

All the segmentation models used in this study were trained on the desktop workstation installed on the Ubuntu 18.04 system, using the DL framework Pytorch version 1.8.0 with an Intel i7-8700k processor, 32GB of RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB of VRAM.

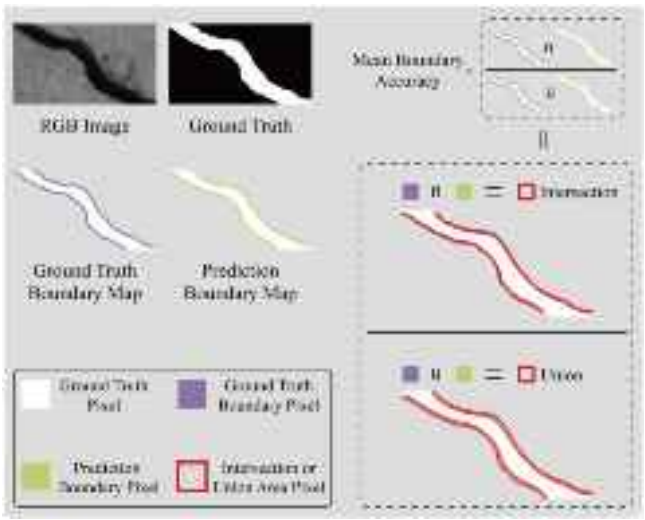

FIGURE 5 A visual demonstration for calculating the mean boundary accuracy.

4.3.2 | Hyperparameters in the learning process

To ascertain the global optimum of the loss function during training, the authors employed the Adam optimizer, integrating momentum and RMSprop benefits, with momentum set at 0.9 and weight decay at 1×10^{-4}. Training was conducted for a maximum of 800 epochs on low-resolution, open-source crack image datasets. The batch size was established at 8, starting with a learning rate of 0.001, which was reduced by 0.0001 after every 10 epochs. Following initial training, the authors applied same hyperparameter setup to further refine the model over an additional 200 epochs using field-collected concrete crack images, culminating in the development of the final crack segmentation model for future application.

4.4 | Model training

4.4.1 | Model training strategy

To obtain the well-trained model, it is necessary to use accurate probability maps and crack labels to effectively supervise the guiding positions of the rendering points and the prediction results of the rendering head, respectively. However, considering that probability maps cannot be directly obtained from open-source datasets like crack labels, but are generated by the rendering point guidance branch constructed by the encoder–decoder architecture, it is nearly impossible to generate the corresponding probability maps for the input crack images using an untrained rendering point guidance branch at the initial stage of model training. This poses a challenge to the training of the model, seemingly making it an impossible task. To address the aforementioned issue, the authors have developed a training method that combines refined labels with edge detection operators for the preliminary guidance of rendering points.

In fact, what the authors need to declare is that the essential role of the probability map is to concentrate the rendering points, which are originally uniformly distributed on the deep semantic feature map, from simple sample areas to hard sample areas such as crack edges and tiny crack branches, by querying the probability intervals of these hard samples. In simple terms, the indirect purpose of the probability map is to accurately locate these hard sample areas. The areas of crack edges and tiny crack branches, which are considered hard samples, can in fact be directly obtained based on edge detection algorithms on the corresponding crack labels. Since every piece of training data comes with a corresponding refined label, and edge detection algorithms require almost no computational resources, this method of rendering point supervision based on labels and edge detection algorithms not only provides accurate location guidance for the model's rendering points but also significantly reduces the model's training cost compared to obtaining the probability map through other means. This approach, which utilizes directly obtainable refined labels for guidance, avoids the extra computational load and the accumulated errors during the computation of uncertain points introduced in the original PointRend, thereby achieving more efficient and accurate guidance of boundary details.

Specifically, edge detection algorithms are used to extract the edges of refined labels, and some rendering 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 to negative samples during training, and considering the need for efficient training, the total number of rendering points per image is set to $N = \frac{H\times W}{20}$, with all sampling points randomly distributed in the crack main body, crack edges, and background areas at a ratio of 0.3:0.4:0.3, respectively. Figure 6 visually demonstrates the final result of rendering point guidance using labels and edge detection algorithms during the training process. When the number of available pixels in any designated region (such as crack body, crack edges, or background) is smaller than the required number of rendering points for that region, the rendering points are initially assigned to fill all available pixels in that region. Any remaining rendering points are then randomly distributed among the available pixels in that area. This process effectively leads to multiple samplings of the same pixel, thereby enhancing the network's learning in underrepresented regions and improving prediction accuracy for those areas. This approach ensures that the model can make accurate predictions even when pixel density in certain regions is low.

It must be emphasized again that the areas most likely to produce erroneous predictions are primarily located at the boundaries and their adjacent areas, rather than the single pixel width crack boundary contours represented in Figure 6. This is due to the unavoidable errors between the actual boundaries and the labeled boundaries caused by the subjectivity of manual annotation. Therefore, if sampling for training rendering points is conducted solely within the boundary region of a unit pixel width as shown in Figure 6, it cannot effectively prevent the incorrect guidance to the model's refinement performance caused by the aforementioned subjective errors. Expanding the sampling to the boundary and its adjacent areas to enlarge the region

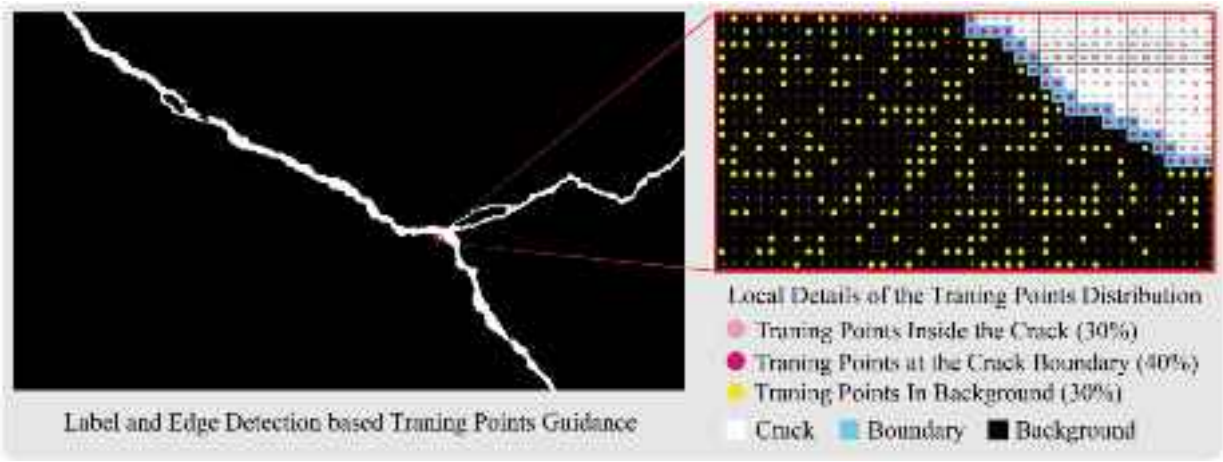

FIGURE 6 Visualization demonstration of providing rendering point location guidance for model training based on the refined label and edge detection algorithm.

for rendering-point guidance to cover areas where subjective errors exist is an effective means to eliminate this kind of incorrect guidance.

4.4.2 | Parameter experiment for the model training

To eliminate the degradation of refinement recognition performance at boundaries due to annotation errors and to obtain the optimal training model, an dilation algorithm is employed to perform expanded sampling on the boundary and its adjacent areas. To implement this boundary expansion operation in practice, the authors first use an edge detection algorithm to extract crack boundaries with a width of one unit pixel, and then uniformly expand towards both the crack interior and the background area from this boundary as the center of dilation, according to a predefined dilation coefficient. As shown in Figure 7, considering the size of the crack image and the width of the crack pixels, four sets of boundary expansions were performed, with the expanded boundary widths being 3, 5, 7, and 9 pixels, respectively. Ultimately, the coordinate points within these dilated areas will be mapped to the crack feature map for feature sampling during the model training process.

The performance of the models trained under the guidance of expanded boundaries of varying widths is summarized in Table 1. Observing the test results in Table 1 reveals that the model performs best when the width of the expanded boundary is 5 (i.e. when the dilation coefficient is 2), with IoU, mBA, and dice scores reaching 80.64%, 84.87%, and 89.28%, respectively. This is because the subjective annotation errors present at the crack boundaries in the training data used in this study coincide with this interval. A comprehensive analysis of the experimental results in Table 1 and the details of the probability map in Figure 3 leads to the following conclusion: Under conditions of too low (dilation coefficients of 0, 1) or too high (dilation coefficients of 3, 4) dilation coefficients, although the performance of models trained with boundary dilation is better compared to models trained without any boundary dilation, these two methods of boundary expansion do not significantly improve performance because the sampling area does not fully cover the entire subjective annotation error area, or because areas outside the error interval dissipate computational resources. Specifically, when the dilation coefficient is 0 or 1, the range of the expanded boundary area is insufficient to encompass deviations produced near the manually annotated boundaries; whereas when the dilation coefficient is 5 or 7, too much of the crack main body and background areas without manual annotation errors are included as ambiguous boundary areas for refinement sampling. These unnecessary simple sample areas divert computational power that should belong to the ambiguous boundary areas, thus reducing the model's learning and representation capabilities for ambiguous boundary areas and making the improvement in model recognition accuracy brought by boundary-guided sampling very limited.

In addition to the experimental results, the selection of a dilation coefficient of 2 (resulting in a 5-pixel-wide expansion) is also physically reasonable based on the spatial characteristics of annotation uncertainty at crack boundaries. Crack boundary pixels typically exhibit irregular morphology, gradual grayscale transitions, and low contrast against adjacent background or crack interior regions, which leads to variability in manual annotations. A 5-pixel dilation width allows the sampling region to

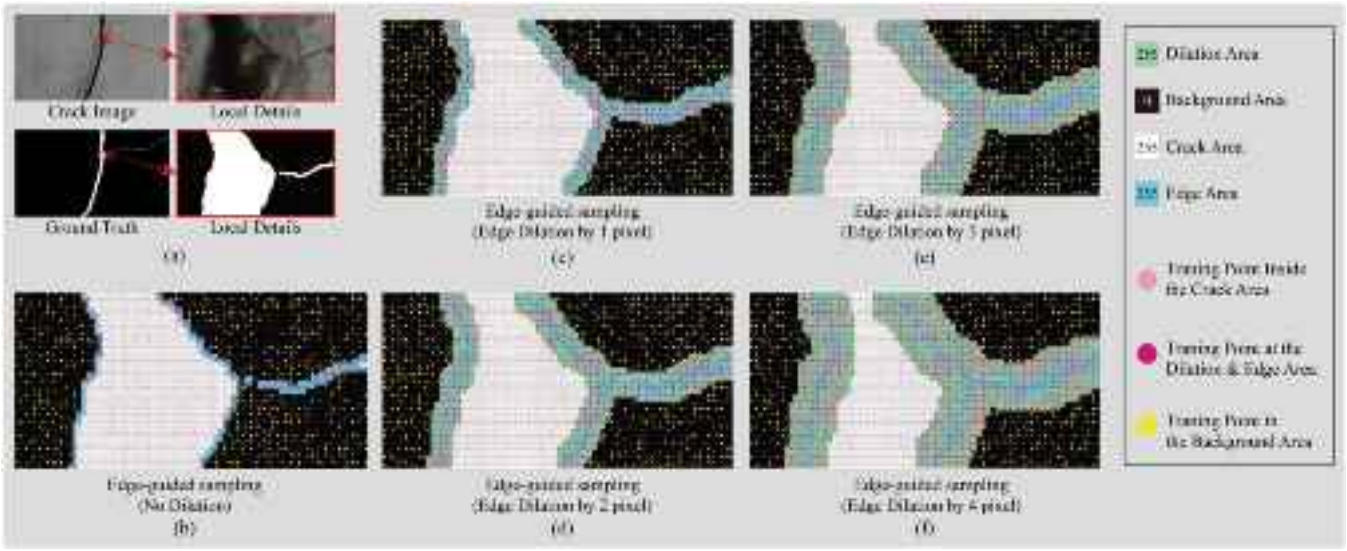

FIGURE 7 Visualization of rendering point sampling guided by boundary areas obtained at different dilation coefficients in a randomly selected crack image, (a) local details of crack image and the corresponding label; (b) distribution of training points guided by boundary areas sampled at a one-pixel width edge; (c–f) distribution of training points after expanding the edges with different dilation coefficients.

TABLE 1 Performance comparison of models trained with rendering point guidance in boundary areas expanded by different dilation coefficients on field-collected HR crack images.

Dilating coefficient	Width of the boundary area after dilation	IoU (%)	mBA (%)	Dice (%)
0	1	75.80	73.51	86.23
1	3	78.87	80.46	88.19
2	5	80.64	84.87	89.28
3	7	78.11	79.86	87.71
4	9	74.64	71.23	85.48

encompass these transitional zones, effectively covering the uncertainty zone caused by annotation imprecision.

4.4.3 | Model fine-tuning

Furthermore, it is important to note that there is a discrepancy between the method of guiding rendering points based on labels and edge detection algorithms during the training phase and the actual operation of guiding rendering points through probability maps required during the inference stage. This can lead to a certain degree of bias between the model on training data and actual test data, as ensuring consistency in the physical steps between model training and inference processes is a prerequisite for guaranteeing model performance (Larochelle et al., 2009). To adhere to the above principle, in the later stages of training, probability maps generated by the lightweight rendering point guidance branch itself are used to fine-tune the guidance for rendering points. Specifically, after conducting preliminary training of the model for 800 epochs using refined labels and corresponding hyperparameters, guidance for rendering points is provided based on the crack probability maps generated by the rendering point guidance branch from the input crack training samples, continuing for another 200 epochs to enhance the model's actual inference performance. It should be noted that the training and testing data used for the first 800 epochs come from low-resolution open-source datasets, while the training data for the last 200 epochs are derived from 250 images patches of 256 × 256 resolution cropped from HR crack images collected in the field as described in Section 4.1.

Equation (5) fully represents the loss function used for model training, which can be seen to consist of two parts: the binary cross-entropy (BCE) Loss for supervising the accuracy of the probability map and the dice loss for supervising the refinement level of the predicted mask. The reason for choosing these two types of loss is as follows: BCE loss is suitable for handling probability outputs and can effectively promote the model's learning of the probability distribution (Zhang & Sabuncu, 2018), optimizing the accuracy of the predictions; dice loss can better address

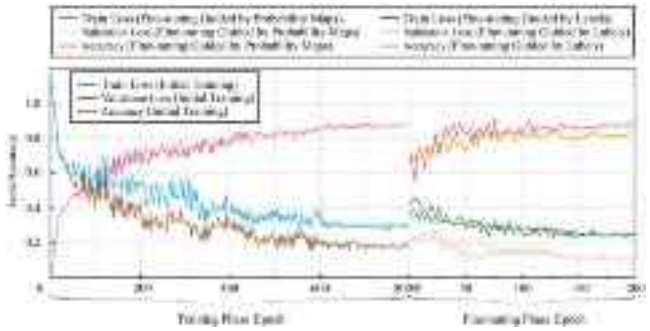

FIGURE 8 Visualization of model performance during the initial training phase and under two fine-tuning strategies.

the issue of class imbalance, and the pixel-level mask of cracks precisely meets the condition of an imbalanced ratio of positive to negative samples (Reib et al., 2023). Considering the decisive role both play in model training, to effectively integrate and balance the advantages of these two supervisory signals, the authors combine BCE loss and dice loss in equal weight proportions to form a composite loss function.

$$L = 0.5 \times L_{\text{BCE}} + 0.5 \times L_{\text{Dice}} \tag{5}$$

Figure 8 provides a visual representation of the changes in the loss and accuracy curves during the model's training and fine-tuning processes. It is evident that after fine-tuning using the method proposed in this study, the fluctuations in the model's loss function are more stable, and the recognition accuracy on the validation set composed of field-collected crack images has also been correspondingly improved. In fact, fine-tuning on the target source dataset to improve model performance is a widely accepted practice. However, it is important to note that the fine-tuning method proposed in this study differs fundamentally from traditional fine-tuning approaches. As described at the beginning of this section, in order to eliminate the performance degradation caused by the inconsistency in the rendering point guidance method between the training and inference stages, the fine-tuning process for 200 epochs utilizes probability heatmaps generated by the model itself to guide the rendering points. To highlight the advantages of this improvement, Figure 8 visualizes the accuracy and loss of the model fine-tuned with crack image labels. It is evident that, in the initial phase of fine-tuning with probability heatmap guidance, the loss fluctuations are higher than those of the label-guided fine-tuning method. However, as the fine-tuning progresses, around 180 epochs, the fluctuations stabilize, and the model's performance surpasses that of the label-based fine-tuning model.

Furthermore, to more intuitively demonstrate the customized training and inference process, the corresponding pseudocode has been added at the Appendix part of the paper.

4.5 | Ablation study

4.5.1 | Ablation study for the encoding backbone

The effectiveness of the proposed encoding backbone structure—SRA enhanced pyramid transformer—was tested on a test set composed of HR crack images collected in the field. Specifically, five typical encoding architectures were selected for performance comparison, including ResNet50 (He et al., 2016), DenseNet (Huang et al., 2017), MobileNetV3 (Howard et al., 2019), vision transformer (ViT) (Dosovitskiy et al., 2020), and pyramid vision transformer (PVT) (Wang et al., 2021). The detection results of models under different configurations on the test set are summarized in Table 2.

By parallely comparing the performance of six groups of models with different crack feature enhancement encoding architectures, it is found that the three models using transformer architecture as the encoding backbone have the highest recognition performance, with the average value of IoU, mBA, and DICE surpassing 78.28%, 80.65%, and 87.87% respectively. Compared to the average improvement of the first three groups of models using CNN backbone for feature enhancement, the increase reached more than 3.35%, 7.17%, and 2.30%, respectively. This is because, compared to the other three types of models built on CNN, Transformers can consider all positions within the crack images simultaneously based on their inherent self-attention mechanism, allowing the model to capture a broader context of crack information. This helps the model understand the relationship between global and local information of crack features, thereby effectively enhancing the model's crack recognition performance. Among all improved performance metrics, the most significant improvement of models using the pyramid transformer architecture is observed in mBA, indicating that the pyramid transformer's ability to capture crack edges and minor crack details in the global image has been significantly strengthened. This is due to the pyramid architecture's fusion of multi-scale features across different levels, allowing features from various scales to complement each other, thereby ensuring that the model can utilize both local detail information and global deep contextual semantic information to enhance the representation of crack edges.

It is also worth noting that the total number of parameters for models incorporating the pyramid transformer

TABLE 2 Performance comparison of models with different encoding backbones on field-collected HR images.

Encoding backbones	IoU (%)	mBA (%)	Dice (%)	Para. (M)
ResNet50	75.38	73.47	85.96	25.6
DenseNetV3	76.18	74.82	86.44	8.02
MobileNet	73.13	72.16	84.30	4.23
ViT	76.39	75.24	86.88	86.37
PVT	77.72	81.85	87.45	24.51
SRA enhanced PVT	80.64	84.87	89.28	18.76

is almost the same as that of models based on CNN for enhancing crack feature representation. Compared with the traditional ViT, the pyramid transformer, by adopting a sliding window approach to process image blocks and introducing compression operations in its self-attention mechanism, is more computationally efficient, especially suitable for large-scale image data, resulting in a more noticeable improvement in model accuracy with the same computational power. Lastly, a parallel comparison of the performance of two groups of models using the PVT architecture shows that, after applying the SRA proposed in this study to reduce the dimensionality of the original multi-head attention mechanism, the model's number of parameters was reduced by nearly 23.46%, while maintaining high recognition accuracy, with IoU, mBA, and DICE reaching 80.64%, 84.87%, and 89.28%, respectively. The comprehensive testing accuracy and visualization results show that the introduction of the Pyramid Transformer in the encoding phase of this study has significant advantages in improving the model's crack recognition accuracy and efficiency.

4.5.2 | Ablation study for different probability interval on the probability map

To maximize the effect of using probability maps to guide rendering points for the enhancement of the model's fine-grained segmentation performance, as described in Section 3.2, it is necessary to designate appropriate probability intervals on the probability map for difficult samples (crack edges and tiny crack branch areas) as well as for simple samples (background and crack main body). Another benefit of performing this operation is that it can minimize the consumption of computational resources during the inference process while maintaining the required inference accuracy because, according to the design in Section 3.2, only the pixels within the difficult sample intervals need to undergo subsequent fine-grained rendering. Simply put, larger probability intervals mean more probability points requiring fine-grained rendering, leading to higher accuracy but also significantly increasing computational redundancy in the inference process; whereas too small probability intervals, although they speed up inference, result in many tiny cracks and boundary details not being effectively rendered in a fine-grained manner, thus severely affecting the final recognition accuracy. Therefore, it is necessary to determine reasonable probability intervals on the probability map for difficult sample areas with uncertain prediction results concentrated around a probability of 0.5. It should be noted that the determination of the probability interval range parameter differs from the boundary dilation coefficient in Table 1. The probability interval range parameter must be adjusted only after the model training is complete. Specifically, two probability values are selected, one as the critical probability value α between the background and boundary areas, and the other as the critical probability value β between the boundary areas and crack pixels.

For the critical probability value α, this study set three groups of different probability parameters, namely 0.2, 0.3, and 0.4; for the critical probability value β, three groups of different probability parameters were also set, namely 0.6, 0.7, and 0.8. The above two types of three different critical probability values defined nine groups of boundary areas with different probability interval ranges. Table 3 provides statistics on the inference results on the test set for models that performed sampling using these nine different probability intervals.

From Table 3, it can be observed that groups 4, 5, and 6 (i.e. the experimental groups with the background probability range between (0.0, 0.3)) achieved relatively superior IoU, dice, and mBA. This is because, compared to the sampling groups where the background probability range is set between (0.0, 0.4), the sampling method under these three groups of parameter settings encompassed a wider background sampling area, thereby more effectively repairing some of the tiny crack details not detected in the background. At the same time, the sampling groups with the background probability range set between (0.0, 0.2) classified too many pixels that belong to the boundary area as background pixels, causing the ambiguous boundary area to be unable to fully achieve accurate boundary detail repair due to insufficient number of sampling points,

TABLE 3 Inference results obtained by rendering points guided by different boundary interval ranges defined on the probability map.

Set No.	Probability range for background area	Probability range for boundary area	Probability range for crack internal area	IoU (%)	mBA (%)	Dice (%)
1	(0.0,0.2)	(0.2,0.6)	(0.6,1.0)	77.37	80.60	87.24
2	(0.0,0.2)	(0.2,0.7)	(0.7,1.0)	78.78	81.91	88.13
3	(0.0,0.2)	(0.2,0.8)	(0.8,1.0)	80.64	84.87	89.28
4	(0.0,0.3)	(0.3,0.6)	(0.6,1.0)	79.01	82.66	88.28
5	(0.0,0.3)	(0.3,0.7)	(0.7,1.0)	82.48	86.80	90.40
6	(0.0,0.3)	(0.3,0.8)	(0.8,1.0)	84.31	94.02	91.49
7	(0.0,0.4)	(0.4,0.6)	(0.6,1.0)	78.29	81.45	87.82
8	(0.0,0.4)	(0.4,0.7)	(0.7,1.0)	81.97	83.96	90.09
9	(0.0,0.4)	(0.4,0.8)	(0.8,1.0)	83.24	87.73	90.86

thus resulting in a relatively lower mBA. Furthermore, comparing groups 4, 5, and 6 reveals that when the range of the crack boundary area is set the largest, that is, when the probability range is between (0.3, 0.8), the accuracy of the model inference is the highest, with IoU, dice, and mBA reaching 84.31%, 94.02%, and 91.49%, respectively. This is because the crack main body area, compared to the background and edge areas, belongs to simple samples with a higher prediction probability (often exceeding 80% confidence), thus not requiring a too broad probability range; whereas the boundary area, as a transitional area between the background and the crack main body area, often exhibits undefined pixel colors and contrasts, leading to significant fluctuations in its prediction probability, hence necessitating a relatively wide probability interval range. Eventually, the sampling parameter configuration set in group 4 is used as the optimal sampling parameter for the inference stage to control the model's subsequent experiments. Indeed, the experimental results also indirectly confirm that the main reasons for the insufficient accuracy in crack segmentation are concentrated in the ambiguous boundary areas, and this area's probability interval on the coarse segmentation probability map is roughly concentrated between (0.3–0.8).

From another perspective, the rationale for setting the probability range of the boundary region between (0.3–0.8) is rooted in the physical characteristics of crack images—particularly the boundary areas, which often exhibit substantial variations in pixel intensity and luminance. These variations stem from factors such as lighting conditions, surface roughness, and crack texture, rendering the boundary regions visually ambiguous. A broader probability interval is therefore necessary to capture these fluctuations and uncertainties, as prediction probabilities in these regions tend to vary more widely than those in the crack body or background, which typically display more consistent and uniform pixel intensities. This probability range facilitates more accurate detection of boundary details, which are essential for precise crack segmentation.

4.6 | Performance comparison with the traditional pointrend architecture

To further illustrate the advantages of the method that uses probability maps for guiding rendering points, this section compares the proposed method with the traditional PointRend architecture (Kirillov et al., 2020), which guides rendering points based on coarse segmentation. Specifically, the authors selected five mainstream DL segmentation architectures with precision from coarse to fine, including FCN-18 (Long et al., 2015), UNet (Ronneberger et al., 2015), DeepLabV3+ (Chen et al., 2018), RefineNet (Lin et al., 2017), and Swin transformer (Liu et al., 2021), to generate the coarse segmentation masks required for guiding rendering points in the PointRend architecture; whereas this study uses probability maps with difficult sample probability intervals set between 0.3–0.8 to guide rendering points. It should be noted that all coarse segmentation architectures and the rendering networks for fine-grained segmentation were trained using the default optimal parameters within the same DL framework under the same configuration. Moreover, when making predictions with the trained coarse segmentation models, all field-collected HR crack images used for testing model performance were scaled down to have a maximum length of 900 pixels to avoid GPU memory overflow issues due to excessively high original resolutions.

The experimental results are shown in Table 4. Initially, observing the accuracy of the masks generated by five coarse segmentation architectures reveals significant differences in the predicted results, ranging from the lowest

TABLE 4 Comparison of inference results on field-collected HR crack images between the segmentation method proposed in this study and the original PointRend architecture.

Rendering based HR image refined segmentation method	Source of the guidance for the rendering points		Coarse segmentation accuracy			Refined segmentation accuracy		
			mIoU	mBA	Dice	mIoU	mBA	Dice
PointRend	Coarse segmentation Guidance	FCN-18	68.96	64.76	81.63	78.15	80.21	87.73
		UNet	70.46	67.54	82.67	78.83	80.56	88.16
		DeepLabV3+	74.11	71.38	85.13	80.16	81.42	88.99
		RefineNet	72.34	70.94	83.95	79.68	81.05	88.69
		Swin transformer	75.08	73.03	85.77	80.97	81.89	89.48
Ours	Probability map guidance	Probability interval for refined rendering $\epsilon[0.3, 0.8]$	/			84.31	94.02	91.49

accuracy of FCN-18 to the highest accuracy of Swin Transformer, with gaps of 6.12%, 8.27%, and 4.14% in IoU, mBA, and Dice, respectively. However, after refined with the original PointRend model, the differences in the refined prediction results become less noticeable, with all five groups of experimental results fluctuating within intervals of 79.56%±1.41%, 81.05%±0.84%, and 88.61%±0.88% for IoU, mBA, and dice, respectively. These results indicate that the PointRend method for fine-grained segmentation indeed does not depend on specific coarse segmentation masks and exhibits good robustness to coarse-grained crack features from different sources. Comparing the final experimental results with the best segmentation results guided by coarse segmentation generated by Swin Transformer within the PointRend group shows that guiding with probability maps further improves the accuracy of the segmentation results. Notably, the improvement in mBA is the most significant among the three metrics, more than double the improvements in mIoU and Dice, reaching 12.13%. This exceptional performance in recognizing boundary details largely benefits from the guidance of rendering points by probability maps with complete detail information and the reasonable setting of probability map intervals, allowing the fine-grained rendering points originally uniformly distributed across the whole image to be effectively concentrated in ambiguous crack edges and tiny crack branches and other difficult sample intervals during the inference stage. To further illustrate the validity of the above inference, Figure 7 visually displays the test results of five randomly selected HR crack images collected from the field. From Figure 9, it can be seen that the segmentation masks obtained by guiding fine-grained rendering points based on probability maps in this study achieve better recognition effects on crack edges and tiny cracks than any prediction masks obtained by guidance based on coarse segmentation, further verifying the validity of the conclusions drawn from the quantitative results.

5 | ON-SITE CASE STUDY

To further assess the advantages of the proposed method in processing HR images of bridge cracks, the UAV was used to collect field crack images of the FuYuan Road Bridge approach in Changsha city. FuYuan Road Bridge is a beam arch bridge constructed primarily from concrete and reinforced steel materials. The bridge spans a total length of 608 meters and was opened to traffic in 2012, serving as one of the most crucial river-crossing channels connecting the northern part of Changsha from east to west. With the increase in the bridge's service life and the growing volume of traffic it supports, noticeable crack damage has appeared on some beam surfaces. As shown in Figure 10, using a DJI M300RTK drone equipped with a top-mounted three-axis stabilized gimbal and H20T multi-sensor camera, 4K resolution images of cracks were collected from the underside of the bridge beams, the main span piers, and the side span piers.

It should be noted that field experiments with the UAV need to consider the negative impact of harsh weather over the river (mainly including random gusts and uneven lighting) on the quality of the collected images, as the shaking of the UAV caused by random gusts may lead to blurring of the collected crack images. To minimize this unavoidable negative impact during field experiments, the authors controlled the UAV's collection process from the following three aspects. First, to ensure the UAV could fly smoothly along the designated flight path, DJI's SmartMap was used for route planning, maintaining the distance between the UAV and the beam at about 3.2 m during the actual image collection process. This not only prevented the UAV from colliding with the beam due to gusts but also ensured that tiny cracks wider than 0.15 mm were fully presented in the collected RGB images as effective pixels. Second, a Z15 Aladdin searchlight was added to the UAV's underslung gimbal to provide supplemental lighting for areas with uneven lighting such as the underside of beams,

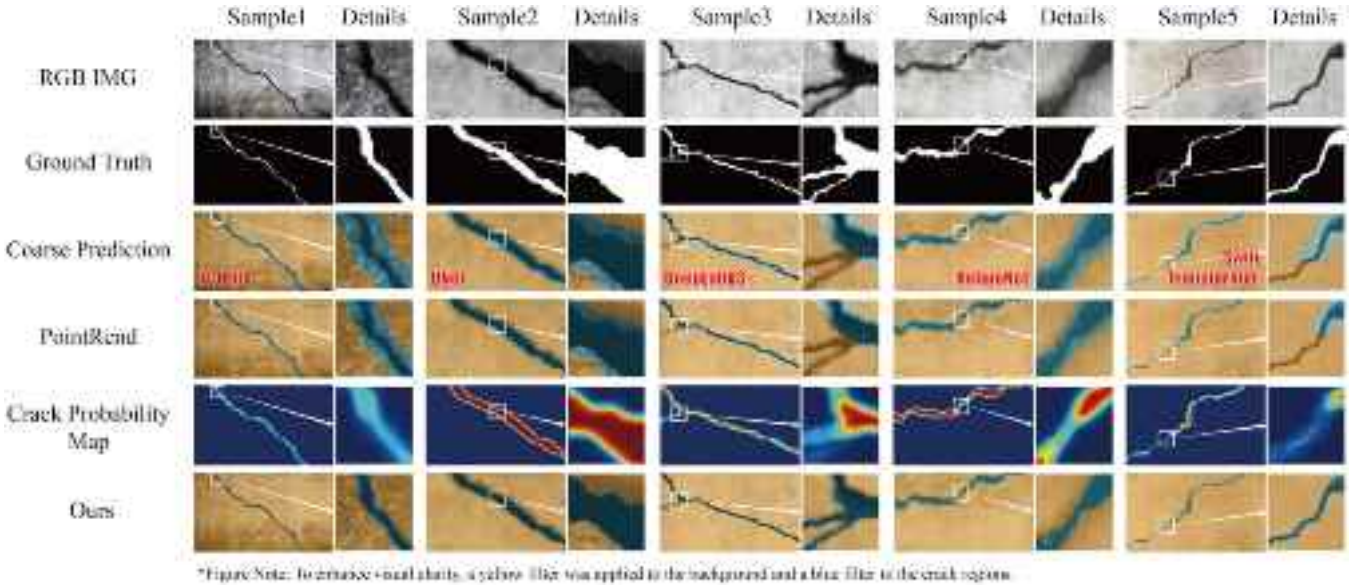

FIGURE 9 Visualization of fine-grained segmentation results of the PointRend architecture guided by different coarse segmentation masks and the proposed method guided by the probability map.

FIGURE 10 Details of the UAV-based crack inspection process conducted on the FuYuan Road Bridge and specific information about the UAV equipment.

significantly improving the issue of lens blur caused by uneven illumination. Lastly, the flight speed of the UAV during the image collection process was controlled at 1 m/s while the camera's shutter speed was set to 1/1000 s to ensure the camera lens had enough 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 should be noted that the M300RTK's built-in high-precision inertial measurement unit, flight control system, visual positioning system, and laser rangefinder form a multimodal positioning system, keeping the accuracy error of the UAV 's cruising in

three-dimensional space within 2 cm. This further ensured that the collection process was accurately executed according to the plan above, thereby ensuring the collection of clear crack images at 4K resolution.

Ultimately, the authors extracted 100 4K resolution (3840 × 2160) crack images from the UAV-collected videos. Additionally, following the same annotation guidelines as the CrackTree260 dataset, the authors utilized the open-source labeling software Labelme to perform pixel-level annotations on all collected HR crack images, ultimately obtaining 100 pixel-level labels with refined edge details for accurate assessment of the detection results.

To effectively validate the advancements of the method proposed in this study, the authors selected two typical types of HR crack image segmentation algorithms for parallel comparative testing on UAV-collected crack data, evaluating model performance primarily in terms of model complexity, inference speed, and crack detection accuracy. Specifically, the first type includes low-resolution image segmentation architectures combined with image preprocessing techniques; the second type encompasses segmentation architectures capable of HR image inference without the need for image preprocessing. For the first type, the authors selected two typical image preprocessing techniques and tested five typical low-resolution image segmentation architectures under each image preprocessing technique, including FCN-18 (Long et al., 2015), UNet (Ronneberger et al., 2015), DeepLabV3+ (Chen et al., 2018), RefineNet (Lin et al., 2017), Swin transformer (Liu et al., 2021), and TransUnet (Chen et al., 2021b); for the second type, the authors chose the most advanced CascadePSP (Cheng et al., 2020), which is based on cascaded progressive refinement segmentation, Segfix (Yuan et al., 2020), which is based on global-local refinement processing, and RLCSN (Chu et al., 2025), which conducts HR crack segmentation also guided by the probability map method. It is important to emphasize that all the above models were fine-tuned on the same crack image dataset described in Section 4.1, using the default pretrained parameter configuration to make these models, originally developed for natural scenes, more comparable to the architecture proposed in this study specifically for crack image segmentation. Additionally, it should be noted that the method proposed in this study and the CascadePSP architecture require coarse segmentation masks containing prior guidance information for HR crack image segmentation. Here, to ensure the effectiveness of parallel comparison while possibly enhancing the performance of the models, the source of the coarse segmentation masks comes from the Swin transformer architecture, which performed best among the first type of methods.

Table 5 provides a statistical analysis of the performance of all models tested. Initially, by comparing the average performance of the two major types of methods, it is evident that the performance of models that require image preprocessing before inferring on low-resolution crack images (the first type) is inferior to that of the second type. Specifically, the average value of IoU, mBA, and dice of the second type of methods reached 80.78%, 87.62%, and 89.41%, respectively, achieving improvements of 12.53%, 23.79%, and 8.31% over the first type. This is because the sliding window operation in the first type of segmentation methods can cause a loss of global semantic information integrity in HR images, while the proportional scaling operation can lead to the loss of tiny crack details in HR images. Furthermore, the downsampling stage of low-resolution image segmentation algorithms further results in a loss of detail information on the already compromised crack feature maps, ultimately leading to suboptimal network recognition performance. On the other hand, the second type of methods, based on coarse segmentation masks, mitigate these negative effects through cascading, step-by-step repair, and rendering operations, thereby obtaining more accurate fine-grained segmentation masks.

Further comparison of the four HR crack image segmentation architectures within the second type of methods reveals that the method proposed in this study achieves certain improvements in IoU, mBA, and dice, reaching 83.67%, 93.46%, and 91.12%, respectively, compared to the other three methods. Notably, the proposed method can achieve 12.74 FPS inference on 4K resolution images without significantly increasing model complexity, which is almost four times faster than the other two coarse mask-guided HR segmentation methods. Additionally, among the three accuracy evaluation metrics, the advantage of the proposed method in boundary prediction accuracy is particularly significant, with its mBA exceeding that of CascadePSP and Segfix by 9.59% and 10.45%, respectively. This primarily benefits from the probability map-based rendering point guidance strategy proposed in this study for ambiguous crack boundaries and tiny crack branch areas, which reallocates computational resources excessively assigned to simple samples to these difficult sample areas, thereby correcting potential random errors from original undersampling or single sampling with concentrated computational power. Additionally, the introduction of the pyramid visual Transformer architecture has enhanced the ability of the method proposed in this study to recognize targets across multiple scales. All accuracy metrics now exceed those of the state-of-the-art RLCSN model in the same category. Furthermore, by directly utilizing multi-scale techniques to improve edge detail representation, this method eliminates the need for complex super-resolution reconstruction in RLCSN's edge-guiding branch, leading to a nearly 50% increase in model inference speed. Moreover, the authors visualize some predictive results of models with relatively good recognition performance among the second type of methods, as shown in

TABLE 5 Comparison of segmentation performance of several current mainstream HR segmentation methods on HR crack images collected by the UAV.

Method category	Preprocessing or predependencies	Specific architecture for segmentation	IoU (%)	mBA (%)	Dice (%)	Para. (M)	Speed (FPS)	GPU (GB)
Traditional low-resolution coarse segmentation architectures	Sliding window cropping	FCN-18	67.43	63.70	80.55	11.72	31.53	3.27
	Downsampling		59.54	51.58	74.64	11.72	11.13	3.27
	Sliding window cropping	UNet	69.80	66.21	82.21	17.14	21.78	5.26
	Downsampling		61.90	53.77	76.47	17.14	7.66	5.26
	Sliding window cropping	DeepLabV3+	72.95	70.30	84.36	14.10	15.36	5.54
	Downsampling		65.06	58.36	78.83	14.10	5.42	5.54
	Sliding window cropping	RefineNet	71.37	68.73	83.30	17.52	18.72	5.78
	Downsampling		63.48	56.11	77.66	17.52	6.58	5.78
	Sliding window cropping	Swin Transformer	75.04	74.55	85.72	47.38	11.74	5.53
	Downsampling		67.06	62.36	80.28	47.38	4.13	5.53
	Sliding window cropping	TransUnet	76.51	76.14	86.34	68.71	8.21	10.86
	Downsampling		68.80	64.12	82.79	68.71	2.88	10.86
High-resolution refined segmentation architectures	Coarse mask	CascadePSP	79.88	83.87	88.81	45.56	3.86	13.06
	Coarse mask	Segfix	79.21	83.01	88.40	64.70	2.54	18.23
	Probability map	RLCSN	80.35	90.13	89.31	16.53	8.76	7.87
	Probability map	Ours	83.67	93.46	91.12	31.16	12.74	8.47

Figure 11a. From the detailed zoom-in images in Figure 11a, the advantages of the proposed method in repairing crack edge details and tiny crack branches are more intuitively evident.

In terms of deployment, while the proposed method has shown good performance on a desktop workstation, its applicability to edge devices such as the NVIDIA Jetson series is an important consideration. Edge devices like the Jetson Nano (4GB memory) and Jetson Xavier NX (8GB memory) offer powerful yet compact solutions for real-time inference. However, the model's 31.16M parameters and 8.47 GB GPU memory requirement on desktop systems suggest that deployment on edge devices may require model optimizations. Potential strategies, including pruning and quantization, will be explored to enhance performance without compromising detection accuracy. It is important to note that inference speed on edge devices may be slower than on workstations, with potential implications for real-time crack detection, especially in dynamic and field-based scenarios.

To comprehensively evaluate the performance of the proposed method, the authors also analyzed failure cases observed during real-world bridge inspections. As shown in Figure 11b, two common misidentifications were found in the field: 1) rough, water-corroded surfaces of concrete after erosion were misclassified as cracks, and 2) dark moss near moist cracks was mistaken for cracks. These non-crack areas were incorrectly assigned probabilities greater than 0.8 in the heatmap. This misclassification occurred due to the model's reliance on texture and shape features that resembled crack patterns. To mitigate such issues, future work will explore the introduction of attention mechanisms prior to convolutional layers to focus the model more precisely on crack regions, reducing false positives in non-crack areas. Moreover, expanding the training dataset with more challenging non-crack features and incorporating a wider range of environmental conditions will improve the model's ability to differentiate between cracks and non-crack features, ultimately reducing errors and enhancing its robustness for real-world applications.

6 | CONCLUSION

This study proposes a probability map guided rendering architecture for fine-grained segmentation of HR crack images. Firstly, a pyramid transformer block embedded with SRA is introduced as an efficient encoding backbone to capture fine-grained deep features of cracks. Secondly, a rendering point guidance method based on probability maps is proposed to replace coarse segmentation, providing more efficient and accurate guidance for rendering points and effectively concentrating computational resources on ambiguous crack edges and tiny crack branches. Additionally, an efficient training strategy is devised to perfectly address the contradiction between the absence of probability maps in training samples and the supervision of accurate rendering point locations. All the

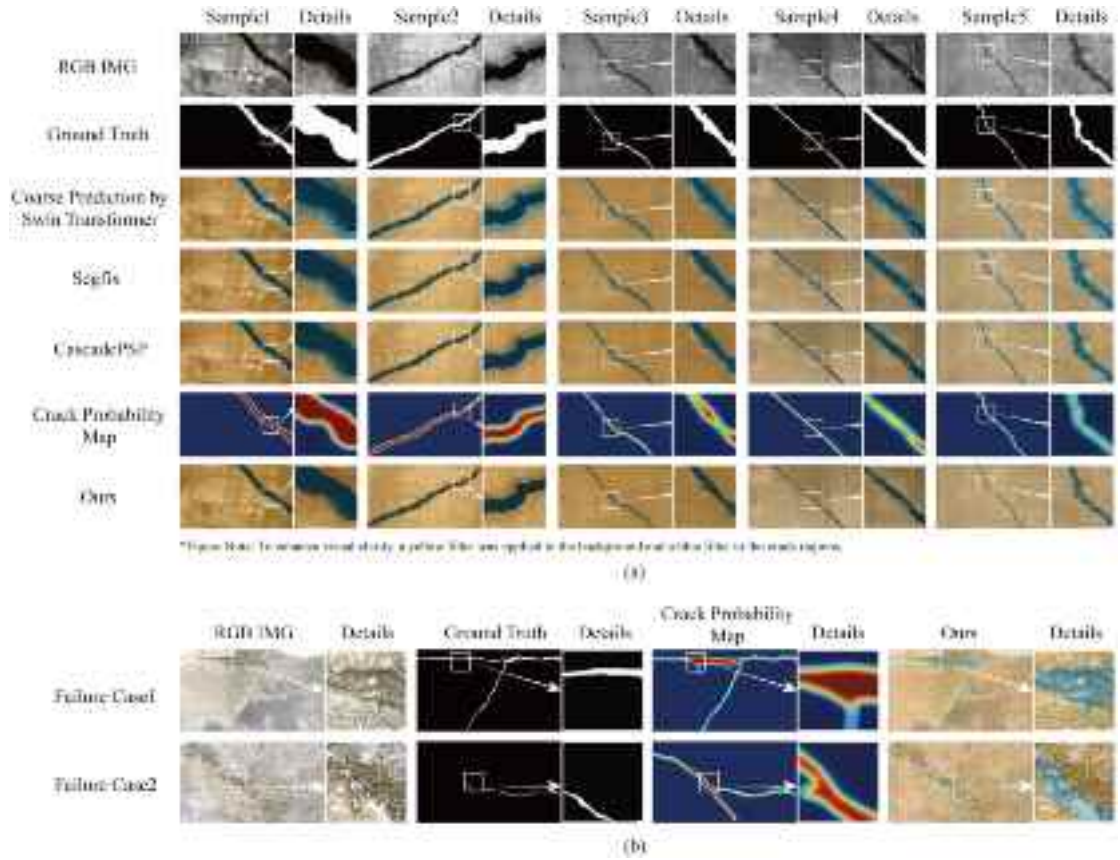

FIGURE 11 (a) Visualization of the segmentation performance on five randomly selected UAV-collected field images by the current two most advanced methods of fine-grained segmentation guided by coarse segmentation, as well as the probability map guided rendering method proposed in this study. (b) Visualization of two typical failure cases. Failure Case 1: Rough, water-corroded surface. Failure Case 2: Moss-like vegetation adhering to the crack.

designs enable the GPU memory-friendly advantages of rendering methods in computer graphics and a high degree of refinement to be fully leveraged in HR crack image segmentation tasks. The ablation studies and the UAV-based field test confirm the superior performance of the proposed method compared to the most advanced methods. Based on the experimental results, the following conclusions can be drawn:

1. This study introduces point rendering technique into the field of HR crack image segmentation for the first time, providing predictive masks with refined boundaries for 4K resolution crack images at 12.74 FPS on commercial GPUs with 24 GB of memory, achieving an IoU of 83.67%, mBA of 93.46%, and dice of 91.12%.
2. A novel paradigm for constructing HR crack image segmentation networks is proposed, where traditional decoding structures' prediction heads are replaced with point rendering heads guided by probability maps. The integration of multi-scale transformer architecture into the encoding structure endows the network with the ability to finely extract crack feature maps. This scheme outperforms the most advanced cascaded refinement segmentation methods in both accuracy and efficiency.
3. During the training phase, introducing a reasonable boundary dilation coefficient for boundary sampling in the rendering head expands the sampling range, eliminating biased guidance caused by discrepancies between real and labeled boundaries due to subjective annotation, significantly enhancing model robustness, and making the model training not wholly dependent on finely annotated data.
4. The custom probability map-based guidance method for boundary rendering point sampling during the inference phase, by concentrating limited computational resources from simple samples dispersed across background and crack main body to hard boundary sample areas, significantly improves model recognition accuracy of ambiguous coarse segmentation boundaries without additional computational resource consumption.

For practical engineering, the application of this method allows HR crack images to be directly used for

inference on some low-cost edge devices, avoiding the loss of crack details caused by image scaling in traditional methods. Especially for UAV-based bridge inspections, the direct application of HR crack images for model inference enables UAVs to conduct safer and more efficient bridge crack detection from a farther distance and with a broader field of view. In future research, the authors will delve into the underlying mechanisms of rendering technology for fine-grained crack segmentation, aiming to more targetedly carry out network pruning and model quantification to further reduce the model's storage needs and dependence on computational resources, enabling its more effective application across various HR UAV platforms for more efficient and cost-effective bridge crack inspection. In addition to the 2D segmentation of crack images, the authors will also explore the integration of 3D spatial data captured by UAVs, including the height and orientation of the camera, to precisely locate cracks on the structure. This will provide maintenance personnel with targeted, actionable insights, facilitating more efficient and localized repair efforts.

From a broader perspective, this method can be adopted for hydropower projects and historic buildings, similarly to detect defects on dams and heritage structures. A notable limitation for promoting the current method is the need for manual adjustment of the probability interval range for fine-grained rendering segmentation when applied to different materials. This step, essential for accurate crack detection across various structural materials, may limit the method's generalizability, as the optimal range can vary depending on material-specific characteristics. To overcome this limitation, future work will focus on developing more adaptive and automated approaches to determine the appropriate probability ranges (Pereira et al., 2020; Rafiei & Adeli, 2017).

Beyond fine-grained damage segmentation, the core rendering-guided sampling mechanism proposed in this study demonstrates strong potential for generalization to other vision tasks requiring boundary refinement, such as multi-class semantic segmentation in remote sensing, medical imaging, or industrial surface defect inspection. With sufficient domain-specific annotated data, the method can be transferred or fine-tuned to accommodate complex boundary structures in diverse application scenarios.

ACKNOWLEDGMENTS

The authors would like to thank the editor, the 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 Horizon Europe Project, D-HYDROFLEX (Grant Number 101122357), the Horizon Europe Project, INHERIT (Grant Number 101123326), the National Natural Science Foundation of China (Grant Number 52278177), the National Key Research and Development Program of China (Grant Number 2023YFC3806800), and the China Scholarship Council (Grant Number 202206130068).

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts.

ORCID

Honghu Chu https://orcid.org/0000-0002-8846-6563
Weiwei Chen https://orcid.org/0000-0003-3359-0556

REFERENCES

Ai, D., Jiang, G., Lam, S.-K., He, P., & Li, C. (2023). Computer vision framework for crack detection of civil infrastructure—a review. *Engineering Applications of Artificial Intelligence*, *117*, 105478. https://doi.org/10.1016/j.engappai.2022.105478

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. https://doi.org/10.1007/s00521-019-04359-7

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). IEEE.

Çelik, F., & König, M. (2022). A sigmoid-optimized encoder–decoder network for crack segmentation with copy-edit-paste transfer learning. *Computer-Aided Civil and Infrastructure Engineering*, *37*(14), 1875–1890.

Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A. L., & Zhou, Y. (2021b). TransUNet: Transformers make strong encoders for medical image segmentation. *arXiv:2102.04306*.

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

Chen, Q., Huang, Y., Sun, H., & Huang, W. (2021a). Pavement crack detection using Hessian structure propagation. *Advanced Engineering Informatics*, *49*, 101303. https://doi.org/10.1016/j.aei.2021.101303

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). IEEE.

Chow, J. K., Su, Z., Wu, J., Tan, P. S., Mao, X., & Wang, Y.-H. (2020). Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Advanced Engineering Informatics*, *45*, 101105. https://doi.org/10.1016/j.aei.2020.101105

Chu, H., Chen, W., & Deng, L. (2024b). Cascade operation-enhanced high-resolution representation learning for meticulous segmentation of bridge cracks. *Advanced Engineering Informatics*, *61*, 102508. https://doi.org/10.1016/j.aei.2024.102508

Chu, H., & Chun, P. J. (2024). Fine-grained crack segmentation for high-resolution images via a multiscale cascaded network. *Computer-Aided Civil and Infrastructure Engineering*, *39*(4), 575–594. https://doi.org/10.1111/mice.13111

Chu, H., Long, L., Guo, J., Yuan, H., & Deng, L. (2024a). Implicit function-based continuous representation for meticulous segmentation of cracks from high-resolution images. *Computer-Aided Civil and Infrastructure Engineering*, *39*(4), 539–558. https://doi.org/10.1111/mice.13052

Chu, H., Wang, W., & Deng, L. (2022). Tiny-crack-net: a multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks. *Computer-Aided Civil and Infrastructure Engineering*, *37*(14), 1914–1931. https://doi.org/10.1111/mice.12881

Chu, H., Yu, D., Chen, W., Ma, J., & Deng, L. (2025). A rendering-based lightweight network for segmentation of high-resolution crack images. *Computer-Aided Civil and Infrastructure Engineering*, *40*(3), 323–347. https://doi.org/10.1111/mice.13290

De Nardin, A., Zottin, S., Piciarelli, C., Colombi, E., & Foresti, G. L. (2023). Few-shot pixel-precise document layout segmentation via dynamic instance generation and local thresholding. *International Journal of Neural Systems*, *33*(10), 2350052. https://doi.org/10.1142/S0129065723500521

Dias, P. A., & Medeiros, H. (2018). Semantic segmentation refinement by Monte Carlo region growing of high confidence detections. Asian Conference on Computer Vision (pp. 131–146). Springer.

Diaz-Frances, J. A., Fernandez-Rodriguez, J. D., Thurnhofer-Hemsi, K., & López-Rubio, E. (2024). Semi-supervised semantic image segmentation by deep diffusion models and generative adversarial networks. *International Journal of Neural Systems*, *34*(11), 2450057. https://doi.org/10.1142/S0129065724500576

Dick, K., Russell, L., Souley Dosso, Y., Kwamena, F., & Green, J. R. (2019). Deep learning for critical infrastructure resilience. *Journal of Infrastructure Systems*, *25*(2), 05019003. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000477

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., & Houlsby, N. (2020). An image is worth 16×16 words: transformers for image recognition at scale. *arXiv:2010.11929*.

Flah, M., Nunez, I., Chaabene, W. B., & Nehdi, M. L. (2021). Machine learning algorithms in civil structural health monitoring: A systematic review. *Archives of Computational Methods in Engineering*, *28*(4), 2621–2643. https://doi.org/10.1007/s11831-020-09471-9

Guo, F., Qian, Y., Liu, J., & Yu, H. (2023). Pavement crack detection based on transformer network. *Automation in Construction*, *145*, 104646. https://doi.org/10.1016/j.autcon.2022.104646

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). IEEE.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., & Vasudevan, V. (2019). Searching for Mobilenetv3. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1314–1324). IEEE.

Huang, G., Liu, Z., Maaten, L. V. D., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2261–2269). IEEE. https://doi.org/10.1109/CVPR.2017.243

Huyan, J., Li, W., Tighe, S., Xu, Z., & Zhai, J. (2020). Cracku-Net: A novel deep convolutional neural network for pixelwise pavement crack detection. *Structural Control and Health Monitoring*, *27*(8), e2551. https://doi.org/10.1002/stc.2551

Ji, D., Zhao, F., Lu, H., Tao, M., & Ye, J. (2023). Ultra-high resolution segmentation with ultra-rich context: a novel benchmark. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 23621–23630). IEEE.

Kellnhofer, P., Jebe, L. C., Jones, A., Spicer, R., Pulli, K., & Wetzstein, G. (2021). Neural lumigraph rendering. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4287–4297). IEEE.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, *54*(10s), 1–41. https://doi.org/10.1145/3505244

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). IEEE.

Konstantakopoulos, I. C., Barkan, A. R., He, S., Veeravalli, T., Liu, H., & Spanos, C. (2019). A deep learning and gamification approach to improving human-building interaction and energy efficiency in smart infrastructure. *Applied Energy*, *237*, 810–821. https://doi.org/10.1016/j.apenergy.2018.12.065

Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, *10*(1), 1–40.

Lin, G., Milan, A., Shen, C., & Reid, I. (2017). RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1925–1934). IEEE.

Liu, C., Zhu, C., Xia, X., Zhao, J., & Long, H. (2022a). FFEDN: Feature fusion encoder decoder network for crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 15546–15557. https://doi.org/10.1109/TITS.2022.3141827

Liu, J., Fang, W., Love, P. E., Hartmann, T., Luo, H., & Wang, L. (2022b). Detection and location of unsafe behaviour in digital images: A visual grounding approach. *Advanced Engineering Informatics*, *53*, 101688. https://doi.org/10.1016/j.aei.2022.101688

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). IEEE.

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). IEEE.

Lu, Q., Lin, J., Luo, L., Zhang, Y., & Zhu, W. (2022). A supervised approach for automated surface defect detection in ceramic tile quality control. *Advanced Engineering Informatics*, *53*, 101692. https://doi.org/10.1016/j.aei.2022.101692

Ma, L., & Hartmann, T. (2023). A proposed ontology to support the hardware design of building inspection robot systems. *Advanced Engineering Informatics*, *55*, 101851. https://doi.org/10.1016/j.aei.2022.101851

Ma, S., Song, K., Niu, M., Tian, H., Wang, Y., & Yan, Y. (2024). Feature-based domain disentanglement and randomization: A generalized framework for rail surface defect segmentation in unseen scenarios. *Advanced Engineering Informatics*, *59*, 102274. https://doi.org/10.1016/j.aei.2023.102274

Mei, Q., Gül, M., & Azim, M. R. (2020). Densely connected deep neural network considering connectivity of pixels for automatic crack detection. *Automation in Construction*, *110*, 103018. https://doi.org/10.1016/j.autcon.2019.103018

Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2017). Large kernel matters–improve semantic segmentation by global convolutional

network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4353–4361). IEEE.

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. https://doi.org/10.1007/s00521-019-04146-4

Quan, J., Ge, B., & Wang, M. (2023). Crackvit: A unified CNN-transformer model for pixel-level crack extraction. *Neural Computing and Applications*, *35*, 10957–10973. https://doi.org/10.1007/s00521-023-08277-7

Rafiei, M., & Adeli, H. (2018). Novel machine learning model for construction cost estimation taking into account economic variables and indices. *Journal of Construction Engineering and Management*, *144*(12), 04018106. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001570

Rafiei, M. H., & Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, *142*(2), 04015066. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001047

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE transactions on neural networks and learning systems*, *28*(12), 3074–3083. https://doi.org/10.1109/TNNLS.2017.2682102

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. https://doi.org/10.14359/51689560

Reib, S., Seibold, C., Freytag, A., Rodner, E., & Stiefelhagen, R. (2023). Decoupled semantic prototypes enable learning from diverse annotation types for semi-weakly segmentation in expert-driven domains. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 15495–15506). IEEE.

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 (pp. 234–241). Springer.

Shamsabadi, E. A., Xu, C., & Dias-da-Costa, D. (2022). Robust crack detection in masonry structures with transformers. *Measurement*, *200*, 111590. https://doi.org/10.1016/j.measurement.2022.111590

Shang, H., Sun, C., Liu, J., Chen, X., & Yan, R. (2023). Defect-aware transformer network for intelligent visual surface defect detection. *Advanced Engineering Informatics*, *55*, 101882. https://doi.org/10.1016/j.aei.2023.101882

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). IEEE.

Siriborvornratanakul, T. (2023). Pixel-level thin crack detection on road surface using convolutional neural network for severely imbalanced data. *Computer-Aided Civil and Infrastructure Engineering*, *38*(16), 2300–2316. https://doi.org/10.1111/mice.13010

Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Yifan, W., Lassner, C., Sitzmann, V., Martin-Brualla, R., & Lombardi, S. (2022). Advances in neural rendering. *Computer Graphics Forum* (pp. 703–735). Wiley.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., & Uszkoreit, J. (2021). MLP-mixer: An all-MLP architecture for vision. *Advances in Neural Information Processing Systems*, *34*, 24261–24272.

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. https://doi.org/10.1109/TPAMI.2020.2983686

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 568–578). IEEE.

Wu, G., Liu, Y., Fang, L., & Chai, T. (2021). Revisiting light field rendering with deep anti-aliasing neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5430–5444.

Wu, Y., Meng, F., Qin, Y., Qian, Y., Xu, F., & Jia, L. (2023). UAV imagery based potential safety hazard evaluation for high-speed railroad using real-time instance segmentation. *Advanced Engineering Informatics*, *55*, 101819. https://doi.org/10.1016/j.aei.2022.101819

Yamaguchi, T., & Mizutani, T. (2024). Quantitative road crack evaluation by a U-Net architecture using smartphone images and lidar data. *Computer-Aided Civil and Infrastructure Engineering*, *39*(7), 963–982. https://doi.org/10.1111/mice.13071

Yuan, Y., Xie, J., Chen, X., & Wang, J. (2020). SegFix: Model-agnostic boundary refinement for segmentation. *European Conference on Computer Vision* (pp. 489–506). Springer.

Zeng, C., Hartmann, T., & Ma, L. (2024). ConSE: An ontology for visual representation and semantic enrichment of digital images in construction sites. *Advanced Engineering Informatics*, *60*, 102446. https://doi.org/10.1016/j.aei.2024.102446

Zhang, C., Lin, G., Liu, F., Yao, R., & Shen, C. (2019b). CANet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5217–5226). IEEE.

Zhang, X., Rajan, D., & Story, B. (2019a). Concrete crack detection using context-aware deep semantic segmentation network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(11), 951–971. https://doi.org/10.1111/mice.12477

Zhang, Z., & Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, *31*, 8792–8802.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1529–1537) . IEEE.

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. https://doi.org/10.1109/TIP.2018.2878966

How to cite this article: Chu, H., Chen, W., & Deng, L. (2025). Refined segmentation of high-resolution bridge crack images via probability map-guided point rendering technique. *Computer-Aided Civil and Infrastructure Engineering*, *40*, 4946–4969. https://doi.org/10.1111/mice.70088

APPENDIX

Pseudocode for the Model Training and Inference

Configuration:
cnn_encoder: MobileNetV3 or ResNet based backbone
cnn_decoder: upsampling decoder such as in Unet
transformer_encoder: PVT encoder enhanced with spatial reduction attention layers
point_rendering_decoder: pointwise rendering decoder with MLP

Operation:
sample_from_label_map: sampling in the background, expanded edges, and crack main body at a ratio of 0.3:0.4:0.3.
sample_from_probability_map: sampling from the probability map according to the corresponding probability interval.

Parameter:
Epoch: total number of training rounds, set to 1000
Epoch _1: number of training rounds in the first stage, which sampling from label map, set to 800
EDGE_THICKNESS: the edge width sampled from labelmap, set to 5

Training:

```
for epoch in range(Epoch):
    feature_map = cnn_encoder(input_img)
    logit_map = cnn_decoder(feature_map)
    probability_map = sigmoid(logit_map)
    probability_map_loss = cross_entropy_loss(probability_map, label_map)
    fine_grained_feature_map = transformer_encoder(feature_map)
    if epoch < Epoch_1:
        guidance_point_coordinates = sample_from_label_map(label_map, EDGE_THICKNESS)
    else:
        guidance_point_coordinates = sample_from_probability_map(probability_map, α, β)
    logit_points = point_rendering_decoder(fine_grained_feature_map[sample_point_coordinates])
    probability_points = sigmoid(logit_points)
    mask_loss = dice_loss(probability_points, label_map[sample_point_coordinates])
    loss = 0.5*probability_map_loss+0.5*mask _loss
    loss.backward()
```

Inference:

```
feature_map = cnn_encoder(input_img)
logit_map = cnn_decoder(feature_map)
probability_map = sigmoid(logit_map)
fine_grained_feature_map = transformer_encoder(feature_map)
guidance_point_coordinates = sample_from_probability_map(probability_map, α, β)
logit_points = point_rendering_decoder (fine_grained_feature_map[guidance_point_coordinates])
probability_points = sigmoid(logit_points)
probability_map[guidance_point_coordinates] = probability_points
return probability_map
```

Notes:
[] represents feature point sampling form the corresponding coordinates