Kalpa Publications in Computing
Volume 22, 2025, Pages 854–867
Proceedings of The Sixth International Conference on Civil and Building Engineering Informatics

Probability Map Guided Point Rendering Technique for Refined Segmentation of High-Resolution Crack Images

Honghu Chu[1], Weiwei Chen[2], and Lu Deng[3*]
[1] Ph.D., College of Civil Engineering, Hunan University, Changsha, China.
[2] Ph.D., Lect., Bartlett School of Sustainable Construction, University College London, London, UK.
[3] Ph.D., Prof., College of Civil Engineering, Hunan University, Changsha, China.
chuhonghu@hnu.edu.cn, weiwei.chen@ucl.ac.uk, denglu@hnu.edu.cn

Abstract

High-resolution (HR) imaging devices are now widely used for capturing crack images from civil structures, necessitating the development of algorithms for HR image segmentation. However, the traditional refined segmentation of HR images requires substantial GPU resources, which leads to the adoption of the cost-effective point rendering technique for inference. Considering that traditional rendering techniques require the use of coarse masks to guide the rendering points for processing prediction, these coarse masks typically fail to effectively focus the rendering points on the boundary regions of the slender cracks, resulting in ambiguous predictions at crack boundaries. In contrast, we introduce a novel rendering point sampling paradigm that enables the network to focus rendering points on crack boundary regions, guided by the probability maps during the inference phase. This approach significantly improves the segmentation accuracy of crack boundary regions from HR images without increasing computational resource dependence. Experiments on an open-source HR crack image dataset consistently show our method's superiority over state-of-the-art approaches, with final results of 84.24%, 93.78%, and 91.45% on IoU, mBA, and Dice, respectively.

1 Introduction

Bridge cracks are a leading cause of many bridge diseases, which can reduce the structural load-bearing capacity of bridges and may trigger structural disasters (Sony et al., 2021; Deng et al., 2022). Therefore, for traffic management departments, it is crucial to identify crack damage timely and accurate to ensure the normal operation of bridges during their service period (Xie et al., 2022). Semantic segmentation methods based on deep learning represent the most promising solution for the

I. Hkust, J. Cheng and Y. Yantao (eds.), ICCBEI 2025 (Kalpa Publications in Computing, vol. 22), pp. 854–867

automatic and rapid identification of these cracks and have become a key topic in the field of bridge maintenance research (Chu et al., 2022).

In recent years, with the development of imaging technology and the increased requirements for detection, high-resolution (HR) imaging devices have begun to be gradually promoted and applied to collect surface crack images of engineering structures. The advantage of collecting HR images is that they contain richer crack detail information, which can be better used for the assessment of structural safety performance. However, traditional deep learning architectures often need to perform complex convolutional operations to generate new pixels during the decoding stage, based on deconvolution or transposed convolution. These convolutional operations require processing a larger number of data points as the size of the image increases, and the intermediate computational results produced need to be cumulatively stored in GPU memory. This makes the use of current traditional deep learning segmentation methods for accurate segmentation of HR crack images a computationally demanding operation, which is difficult to implement on conventional commercial GPUs, thus significantly limiting the advantages of HR imaging devices in terms of detailed characterisation from being fully utilised in practical tasks.

Unlike traditional encoder-decoder architectures, PointRend, proposed by Kirillov et al. (Kirillov et al., 2020), replaces the conventional decoding architecture with lightweight MLPs (Multilayer Perceptrons) that share weights and perform predictions through point-wise rendering. The MLP designed for point-wise rendering calculates each pixel of the image independently, meaning the network only needs to process a single or a small batch of pixel data at any given time. Therefore, this method is different from traditional upsampling, which requires expanding the entire feature map to a higher resolution all at once, thereby occupying a significant amount of GPU memory. In simple terms, this lightweight MLP performs predictions through point-wise rendering, significantly reducing the segmentation network's dependency on GPU memory while ensuring that the GPU memory does not increase with the size of the inference image.

Meanwhile, compared to traditional methods of dealing with GPU memory constraints by resizing HR images to smaller sizes or cropping HR images into patches for patch-wise inference (Cheng et al., 2020; Wang et al., 2020), the MLP-based point-wise rendering can perform calculations at the original resolution, thus avoiding the potential detail loss caused by proportional resizing and the inconsistencies or stitching artefacts that may occur during the stitching process, thereby better ensuring the coherence and consistency of the prediction results. Overall, the PointRend is considered the optimal solution for addressing GPU memory limitations in high-resolution image segmentation currently available.

However, this point-rendering method was originally designed for conventional targets in natural scenes, but cracks, as a type of target with elongated topological structures and random distribution, differ greatly from conventional targets. Therefore, directly applying the point-rendering method for segmenting cracks poses challenges. Specifically, the original point-rendering technique, in the decoding phase, uses coarse segmentation masks to guide the rendering points for refined prediction, which is feasible for targets of conventional sizes because the edge details of such targets can be relatively accurately represented in the coarse segmentation masks. However, for targets like cracks that have elongated topological structures and random distribution, a large amount of detail features, including the edges of cracks and tiny crack branches, are already lost in the coarse segmentation masks. This deficiency results in rendering points being unable to effectively focus on these lost detail feature areas, thus producing ambiguous predictions for crack boundaries and missing predictions for tiny crack branches. Figure 1 provides a visual demonstration of the process and results of using coarse segmentation masks to guide rendering points for segmenting crack images. Due to the inaccuracy of the coarse segmentation, the refined rendering points are unable to be effectively guided to the tiny crack branches (the red marked areas in Figure 1(b)) and are instead more often directed to the inaccurately defined coarse seg-mentation crack edges (the green marked areas in Figure 1(b)).

This results in the final predictions obtained through rendering techniques not only missing the tiny crack details but also failing to achieve refined predictions at the crack edges.

It should be noted that, in the field of crack inspection for civil engineering, the precise segmentation of crack edges and the complete identification of tiny crack branches are of indispensable importance for assessing structural integrity, implementing early damage recognition, formulating maintenance strategies, and conducting long-term health monitoring to ensure structural safety.

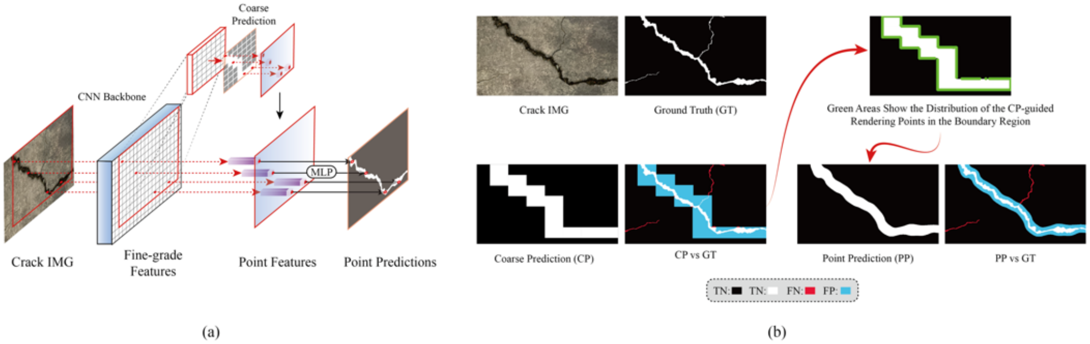

Figure 1. The process of segmenting a randomly selected HR crack image using the original PointRend architecture. (a) Schematic diagram of the PointRend architecture; (b) Coarse segmentation used to guide the rendering points and the final point prediction of the rendering points under its guidance.

To maintain the GPU memory efficiency benefits of the point-rendering method for segmenting HR images while ensuring precise inference at crack boundaries and tiny crack branches, this study proposes a refined rendering approach. This approach utilizes a probability map during inference to guide rendering point sampling. By redefining the intervals of simple and hard samples on the probability map, rendering points can be concentrated from simple samples, such as the background and main body of cracks, towards hard-sample areas like crack edges and tiny crack branches. Consequently, the network achieves refined segmentation for HR crack images on standard commercial GPUs. Our contributions are summarized as:

– For the first time, we incorporate point-rendering technology into HR crack image segmentation, tackling two significant challenges inherent to conventional segmentation methods: the substantial reliance on GPU computational resources and the inadequacy in achieving fine-grained segmentation detail.

– We customized a set of rendering point guidance strategies for the inference phase, which not only reduces the difficulty of training the rendering model but also significantly increases the model's segmentation accuracy for detailed areas such as crack edges and minor crack branches.

– 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 multi-scale Transformer architecture for enhanced crack detail extraction.

We conducted experiments on an HR crack image dataset collected in the field and demonstrated that our model achieves state-of-the-art performance in both quantitative and qualitative results, with enhanced fine-grained segmentation capabilities and lower computational resource dependency.

2 Related Work

2.1 Deep Learning-based Crack Inspection

To enhance the detection of cracks' morphological characteristics (direction, edges, and corners), semantic segmentation algorithms rooted in deep learning have been applied to crack image analysis. Initially, FCN and SegNet served as foundational frameworks (Bang et al., 2019; Liu et al., 2019; Zhang et al., 2019), albeit their repeated downsampling during feature extraction obscured significant fine crack details in the output (Zhang et al., 2019). The introduction of skip connections within U-shaped encoder-decoder architectures addressed these limitations partially, setting a new standard for crack detection (Ronneberger et al., 2015). Despite their advancements, these architectures, largely reliant on CNNs for feature extraction (Huyan et al., 2020; Mei et al., 2020), struggle with modelling long-distance interactions, complicating the segmentation of slender cracks (Khan et al., 2022). Recent shifts towards Transformer architectures (Qu et al., 2022; Shamsabadi et al., 2022; Guo et al., 2023), with their self-attention capabilities for extracting crack features, aim to overcome CNNs' constraints. Yet, these approaches do not adequately prioritize computational resources for critical areas like crack edges, leading to indistinct outcomes on mask boundaries.

2.2 Refined Segmentation

To enhance segmentation accuracy, refinement methodologies were introduced, starting with the amalgamation of Conditional Random Fields (Zheng et al., 2015; Lin et al., 2016; Chen et al., 2017) and Graph Models with deep learning frameworks (Dias & Medeiros, 2018). These initial attempts often relied on basic colour boundaries and did not leverage advanced semantic insights, inadequately addressing contours with minimal contrast to the background. Addressing this, some scholars proposed specialized refinement modules (Peng et al., 2017; Zhang et al., 2019), although their applicability was hampered by the prerequisite of predetermined thresholds. This prompted a shift towards the creation of versatile plug-in modules for broader utility. Notable advancements include RefineNet by Lin et al. (Lin et al., 2017), which refines boundaries through multi-path connections that integrate features across layers during network upsampling, and HRNet by Sun et al. (Wang et al., 2020), employing multi-path connections to enhance detail capture via high-resolution representation learning, thereby offering comprehensive image structural insights. Yet, these approaches' reliance on extensive computational re-sources and their substantial parameter counts challenge their application to high-resolution crack image segmentation and model training efficiency.

2.3 Rendering-based Technology for Refined Segmentation

The dilemma of indistinct boundary regions in crack image analysis parallels the jagged edge phenomenon in computer graphics when images are pixelized (Barron et al., 2021; Wu et al., 2021). This domain, crucial for generating and rendering images for applications like gaming and film effects, showcases the pinnacle of computer vision technology through image synthesis, animation, and virtual reality (Kellnhofer et al., 2021; Wang et al., 2021; Tewari et al., 2022). Drawing inspiration from these rendering techniques, Kirillov et al. (Kirillov et al., 2020) introduced PointRend, a novel approach utilizing rendering for enhanced segmentation in natural scenes. PointRend utilizes MLPs with shared weights for point-wise pixel prediction, capitalizing on rendering's precision without burdening GPU memory, thus enabling its application on standard GPUs for high-resolution imagery. Nevertheless, PointRend's dependency on coarse masks for rendering point guidance falls short of accurately detecting slender crack features. Addressing this, our study introduces a refined rendering strategy that maintains PointRend's memory efficiency while improving accuracy on crack peripheries and fine branches through a probability map-guided rendering point approach.

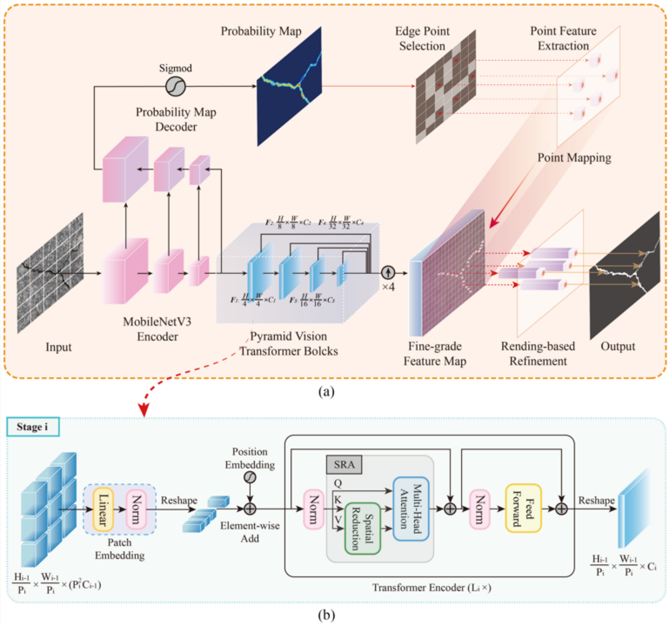
Probability Map
Edge Point Selection
Point Feature Extraction
Sigmod
Probability Map Decoder
Point Mapping
$F_2: \frac{H}{8}\times\frac{W}{8}\times C_2$
$F_4: \frac{H}{32}\times\frac{W}{32}\times C_4$
$F_1: \frac{H}{4}\times\frac{W}{4}\times C_1$
$F_3: \frac{H}{16}\times\frac{W}{16}\times C_3$
×4
Input
MobileNetV3 Encoder
Pyramid Vision Transformer Bolcks
Fine-grade Feature Map
Rending-based Refinement
Output
(a)
Stage i
Linear
Norm
Patch Embedding
Reshape
Position Embedding
Element-wise Add
SRA
Q
K
V
Norm
Spatial Reduction
Multi-Head Attention
Norm
Feed Forward
Reshape
$\frac{H_{i-1}}{P_i}\times\frac{W_{i-1}}{P_i}\times(P_i^2C_{i-1})$
$\frac{H_{i-1}}{P_i}\times\frac{W_{i-1}}{P_i}\times C_i$
Transformer Encoder (L_i ×)
(b)

Figure 2. Schematic diagram of the proposed network. (a) The overall architecture; (b) Transformer block with the SRA embedded.

3 Method

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 point-by-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

To ensure that the deep semantic feature maps contain rich crack detail information while improving the model's inference efficiency, this study employs a custom feature extraction encoder that combines a lightweight encoding architecture with an improved pyramid visual Transformer. An image is first input into the front end based on a lightweight encoding architecture, specifically adopting the MobileNetV3 architecture [31]. At this stage, the image undergoes a series of depthwise separable convolution operations that filter each channel of the input image using depth convolutions, followed by pointwise convolutions that fuse the features of these channels. This effectively reduces the model's parameter count and computational complexity while retaining key image features. The output of the lightweight encoding architecture is a set of feature maps that are relatively smaller in spatial dimensions but still retain important visual information of the image.

Subsequently, these feature maps are fed into an improved Pyramid Visual Transformer (PVT) module (Wang et al., 2021). The original PVT module adopts a hierarchical Transformer architecture capable of processing feature maps at different scales, achieving the extraction and fusion of multi-scale features. Within each layer, feature maps are processed through a self-attention mechanism, allowing the model to capture global contextual information and enhance its understanding of image details. Through progressive processing, the PVT module gradually increases the resolution of feature maps while integrating richer contextual information, ultimately producing a set of high-resolution, semantically rich feature maps. Figure 2(b) visualizes the computational logic at each level of the PVT used in this study, which consists of a Patch Embedding layer and a Transformer encoding layer at the L_i level. It is important to note that since this study involves making predictions on HR images, to avoid GPU memory overflow due to excessive computation, the original Transformer encoding layer's multi-head attention layer (MHA) is replaced with a Spatial Reduction Attention layer (SRA). Similar to MHA, SRA still takes Query (Q), Key (K), and Value (V) as inputs. The difference lies in that SRA reduces the spatial dimensions of K and V through spatial reduction operations before the attention operation, thereby significantly enhancing computational efficiency.

In summary, by adopting a feature extraction encoder that combines a lightweight encoder with the visual Transformer, the advantages of both architectures in computational efficiency and feature expression capability are effectively integrated, providing an efficient solution for the extraction of fine-grained crack features.

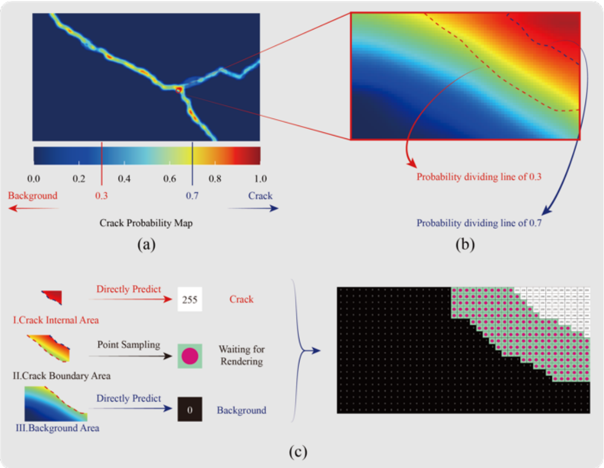

Figure 3. Visual demonstration of the position for the guided rendering point used in the inference phase.

3.2 Point Rendering-based Decoder

To fully decode the fine-grained crack information from the deep semantic feature maps captured by the encoder, the authors improved upon the original PointRend and proposed a decoder that composed of a rendering point guidance branch and a point-rendering based fine-grained prediction head. For the fine-grained prediction head, the authors adopted the same architecture as PointRend because the use of MLPs in rendering technology offers computational efficiency advantages due to weight sharing and point-by-point prediction compared to traditional CNNs. Regarding the branch that guides the rendering points, considering that the rendering point guidance strategy in the

PointRend architecture was designed for traditional large-size natural scene targets and is not suitable for tiny crack targets with elongated topological structures, this study conducted specialized customization.

We specifically designed a rendering point guidance strategy based on the probability map for the inference phase to ensure that the model can effectively concentrate computational resources on difficult-to-predict tiny cracks and crack boundary areas. Specifically, we aimed to achieve efficient boundary rendering point guidance through the refined probability map. As shown in Figure 2(a), the probability map is primarily obtained through forward propagation calculations involving a lightweight encoder followed by two convolution blocks, without the need for thresholding or other post-processing steps to determine final category assignments. Therefore, this process is more direct and efficient computationally compared to generating coarse segmentation masks. More importantly, the probability map can reflect the probability of pixels belonging to each category, rather than a simple categorical affiliation. This probabilistic information provides a continuous measure of confidence for each pixel, rather than categorically assigning pixels in a binary manner, thereby preserving more uncertainty and subtle differences about the image area, which is crucial for understanding the nuances and complex scenes within the image. Since crack segmentation is inherently a binary classification task, based on the characteristics of binary classification tasks, this study divided areas on the probability map into three parts: areas with probabilities close to 0 are considered definite background pixels, areas with probabilities close to 1 are considered definite crack regions, and areas with probabilities fluctuating around 0.5 are considered indeterminate areas by the model. These indeterminate areas often consist of pixels with intensity or contrast levels between cracks and backgrounds, primarily concentrated around tiny crack branches and crack boundary areas. Based on the above division principles, only areas on the probability map with probabilities fluctuating around 0.5 undergo refined rendering point sampling during the inference process. Regarding areas on the probability map with probabilities close to 0 and 1, which are considered easily recognizable pixels, they will be directly mapped as background and crack pixels on the prediction mask without further computation, considering the difficulty of recognition and computational efficiency. To visually demonstrate the rendering point guidance method during the inference phase, Figure 3 visualizes the probability map for a randomly selected crack sample. It can be clearly seen on the probabilistic heatmap that the probabilities of the background and main crack areas are concentrated around 0 and 1, respectively. In boundary areas and tiny crack regions, due to issues such as manual annotation errors and insignificant color differences, the probabilities of pixels on the heatmap fluctuate around 0.5. For this reason, this study sets the probability interval for these hard-to-identify boundary and tiny crack pixels between 0.3-0.8. In the subsequent refinement rendering stage, only pixels with probabilities between 0.3-0.8 undergo refined inference. The parameter settings involved in the rendering point guidance strategy during the training and inference phases will be detailed in Section 3.2.

4 Experiments

4.1 Datasets

Gathering substantial crack image datasets is vital for the development and assessment of our segmentation method. Despite the existence of various datasets like Stone331, Bochum CrackDataSet, Deepcrack537, and CRKWH100 (Quan et al., 2023) for civil infrastructure analysis, the intricacies of acquiring and annotating crack images limit dataset sizes and resolution, typically capping at 500 images with dimensions under 600 × 800. Such constraints risk model overfitting and lessen computational demand due to lower resolution. To bolster model adaptability across varying realworld cracks, this study amalgamated images from AigleRN, CrackTree260, and Crack500 (Ai et al., 2023), adjusting all to a uniform resolution of 256 × 256 for cohesive model training and evaluation. This resulted in 800 adjusted images distributed across training, validation, and testing phases with a ratio of 0.6:0.2:0.2 to rigorously assess model efficacy.

Given that higher-resolution images demand more extensive downsampling to fit within GPU memory constraints during inference, they are inherently more susceptible to indistinct boundary delineations than their lower-resolution counterparts, necessitating enhanced refinement. To address this, the study amassed a HR crack image dataset for thorough model assessment. In Changsha's urban context, diverse structural elements such as walls, roads, and foundations were selected for HR image acquisition, employing both tripod-mounted and handheld Nikon D5300 cameras for crisp imagery. A total of 300 6K RAW crack images were compiled. To mitigate the skew from disproportionate sample sizes, crack regions in these images were meticulously cropped, yielding 60 images across 2K, 4K, and 6K resolutions. Each cropped image was subsequently subjected to detailed annotation for precise model evaluation.

4.2 Evaluation Index

Two commonly used metrics, namely Intersection over Union (IoU) and the Dice similarity coefficient (Dice), were selected to quantitatively evaluate the experimental results. Furthermore, to highlight the performance of the proposed method in boundary areas, the Mean Boundary Accuracy (mBA) introduced in CascadePSP (Cheng et al., 2020) was also used as a metric. The core concept of mBA involves calculating the IoU between the Ground Truth (GT) and the predicted mask within the boundary area.

4.3 Implementation Details

Hardware equipment: All the crack segmentation networks mentioned in this document were completed training under the Ubuntu 18.04 system, using the Pytorch 1.8.0 version as the deep learning framework, with an Intel i7-8700k processor, 32GB of RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB of VRAM.

Hyperparameters: To ensure the global optimum of the loss function is found during the training process, Adam, which combines the advantages of momentum and RMSprop, was chosen as the optimizer for the model, with momentum set at 0.9 and weight decay at 1×10^{-4}. The maximum number of training epochs on the low-resolution open-source crack image dataset was set to 800. The batch size was set at 8, with an initial learning rate of 0.001, decaying by 0.0001 every 10 training epochs. After completing the initial training, the same hyperparameter configuration was used to fine-tune the model's performance for an additional 200 epochs using onsite collected concrete crack images, to obtain the final model for subsequent crack segmentation tasks

4.4 Ablation Study

Ablation Study for the SRA Enhanced PVT: The effectiveness of the improved Pyramid Transformer introduced at the end of the encoder was tested on the test set. Specifically, four typical encoders including ResNet50, DenseNet, MobileNet, Vision Transformer (ViT), and the original Pyramid Vision Transformer (PVT) were selected for performance comparison. The segmentation results of the rendering models with different encoders are summarized in Table 1.

By conducting a parallel comparison of the performance of five groups of models utilizing different crack feature enhancement encoders, it can be observed that the networks adopting the Transformer architectures generally outperform those built with CNN architectures, with average improvements in IoU, mBA, and DICE reaching 2.81%, 3.99%, and 1.81%, respectively. This is because, compared to the other three types of models built on the CNN, Transformers can consider all positions within crack images simultaneously through their inherent self-attention mechanism, enabling the model to capture a broader context of crack information. This assists the model in understanding the connections between global and local information of crack features, thereby effectively enhancing the model's crack recognition performance. Further comparison among the three groups of models using the Transformer architecture reveals that the two models adopting PVT outperform the ordinary VIT in recognition performance. Notably, the most significant improvement is observed in mBA, indicating that the Pyramid Transformer has significantly strengthened its ability to capture crack edges and tiny crack details in the global image. This is attributed to the Pyramid Transformer's fusion of multi-scale features across different levels, allowing features from various scales to complement each other. Thus, ensuring the model can utilize both local detail information and global deep contextual semantic information to enhance the representation of crack edges. Lastly, comparing the two groups that employed the PVT architecture, it is evident that after implementing the SRA pro-posed by this study to reduce the dimensions of the original multi-head attention mechanism, the model's number of parameters decreased by nearly 24%, while maintaining high recognition accuracy.

Feature Extraction Backbone	IoU(%)	mBA(%)	Dice(%)	Parameter (M)
ResNet50	75.48	73.17	86.03	25.6
DenseNet	76.21	73.82	86.50	8.02
MobileNet	73.06	72.39	84.44	4.23
ViT	77.47	74.76	87.30	86.37
PVT	77.81	78.25	87.52	24.51
SRA Enhanced PVT	77.91	78.33	87.58	18.76

Table 1. Performance comparison of models with different crack feature enhancement backbones introduced during the encoding phase on the test dataset

Ablation Study for the Probability Map-based Rendering Point Guidance: To minimize the computational resource consumption during the inference process while maintaining the required inference accuracy, it is necessary to determine a reasonable probability range for areas with uncertain prediction outcomes where the probabilities are concentrated around 0.5. This is because a larger probability range means that more sampling points need to be refined, which, while increasing accuracy, also significantly raises the computational redundancy of the inference process; conversely, a smaller probability range can enhance inference speed but may result in many tiny cracks and boundary details not being effectively refined, thus severely affecting the refinement level of prediction outcomes. Therefore, it is essential to establish suitable threshold probability values to reasonably control the boundary range. Specifically, two probability values are selected: the critical probability value α between the background and crack boundary areas, and the critical probability value β between the crack boundary and crack internal areas.

For the critical probability value α, this study set three different probability parameters: 0.2, 0.3, and 0.4; for the critical probability value β, three different probability parameters were also set: 0.6, 0.7, and 0.8. These two sets of three different critical probability values define nine different boundary area probability ranges. Table 2 presents the statistical inference results on the test set for models using sampling with these nine different probability ranges.

Table 2 shows that sets 4, 5, and 6 (i.e., the experimental groups with the background probability range set between 0.0 and 0.3 achieved relatively superior IoU, Dice, and mBA scores. This is because, compared to the sets with the background probability range set between 0.0 and 0.4, these three parameter settings encompass a wider background sampling area, thereby facilitating the repair of tiny crack details that were not detected in the background. At the same time, the sets with the background probability range set between 0.0 and 0.2 classified too many pixels, which should belong to the boundary area, as background pixels. This resulted in an insufficient number of rendering points guided towards the ambiguous boundary area, unable to fully repair the crack boundary details in that area, and therefore achieved the relatively lower mBA scores. Furthermore, comparing sets 4, 5, and 6 reveals that the highest accuracy in model inference was achieved when the probability range for the crack boundary area was set to its maximum, i.e., when the probability range was between 0.3 and 0.8, with IoU, Dice, and mBA reaching 84.24%, 93.78%, and 91.45%, respectively. This is because the crack main body area, compared to the background and edge areas, is considered a simple sample with a high prediction probability (often exceeding 80% confidence), thus not requiring a too wide probability range. However, as the boundary area serves as a transition zone between the background and the crack main body, where pixel color and contrast often present indistinct conditions, leading to significant fluctuations in prediction probability, a relatively wide probability interval range is needed. Ultimately, the sampling parameter configuration in set 4 was adopted as the optimal inference phase sampling parameter to guide the model in providing sufficient guidance for rendering points during the inference stage. Indeed, the experimental results also indirectly confirm that the main reason for inadequate crack segmentation accuracy concentrates on the ambiguous boundary area, with the probability range on the coarse segmentation probability map roughly concentrating between 0.3 and 0.8.

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)	77.16	80.39	87.11
2	(0.0,0.2)	(0.2,0.7)	(0.7,1.0)	78.59	81.77	88.01
3	(0.0,0.2)	(0.2,0.8)	(0.8,1.0)	80.04	84.56	88.92
4	(0.0,0.3)	(0.3,0.6)	(0.6,1.0)	78.72	82.34	88.10
5	(0.0,0.3)	(0.3,0.7)	(0.7,1.0)	82.35	86.73	90.32
6	(0.0,0.3)	(0.3,0.8)	(0.8,1.0)	84.24	93.78	91.45
7	(0.0,0.4)	(0.4,0.6)	(0.6,1.0)	78.06	81.32	87.68
8	(0.0,0.4)	(0.4,0.7)	(0.7,1.0)	81.80	83.60	89.99
9	(0.0,0.4)	(0.4,0.8)	(0.8,1.0)	83.03	87.24	90.71

Table 2. Performance comparison between the traditional decoder and the proposed rendering-based fine-grained prediction head across models trained with various parameterized feature point guidance strategies

Refinement Segmentation Architecture	Source of the Boundary Sampling Guidance		Coarse Segmentation Accuracy			Refined Segmentation Accuracy		
			IoU	mBA	Dice	IoU	mBA	Dice
Point Rend	Coarse Mask Guidance	FCN-18	68.73	64.36	81.47	77.90	79.79	87.58
		UNet	70.15	67.28	82.46	78.59	80.03	88.01
		DeepLab V3+	73.87	71.12	84.97	79.92	81.03	88.83
		RefineNet	72.10	70.70	83.79	79.45	80.78	88.54
		Swin Transformer	74.85	72.89	85.61	80.72	81.36	89.33
Ours	Probability Map Guidance	Probability interval ∈ [0.3,0.8]		/		84.24	93.78	91.45

Table 3. Performance comparison between the PointRend guided by various coarse masks and the proposed probability map-guided architecture

4.5 Comparison of the Performance of Models that use the Heatmap and Coarse Segmentation for Guiding Rendering Points during the Inference Phase

To elucidate the benefits of the decoding architecture, which utilizes a probability map for rendering point guidance, over the original PointRend architecture that guided by the coarse segmentation mask, a comparative analysis of their performance was conducted based on the test dataset. Specifically, we selected five mainstream deep learning segmentation architectures with varying degrees of segmentation precision, including FCN-18, UNet, DeepLabV3+, RefineNet, and Swin Transformer, as the networks generating coarse segmentation masks required for predictions by the original PointRend architecture. The network proposed in this study, which guides rendering points with the probability map, participated in the performance comparison using optimal parameters obtained from prior ablation experiments. It is noteworthy that all the coarse segmentation architectures and the refinement segmentation networks were trained under the same configurations in the same deep learning framework with default optimal parameters. Additionally, when making predictions using the trained coarse segmentation models, all HR images were scaled down to have their longer side be 900 pixels to prevent GPU memory overflow caused by excessively high original resolutions.

Experimental results, as shown in Table 3., indicate significant differences in the Coarse segmentation accuracy generated by various coarse segmentation architectures, as observed from rows 2 to 5 from the top. The disparities in IoU, mBA, and Dice scores range from 6.12%, 8.53%, to 4.15%, respectively, from the lowest accuracy with FCN-18 to the highest accuracy with Swin Transformer. However, after applying the original PointRend model for refinement, the differences in refined prediction results become less pronounced, with all five sets of experimental results fluctuating within the ranges of 79.31±1.4% for IoU, 88.46±0.79% for mBA, and 88.46±0.88% for Dice. These findings demonstrate that the original PointRend architecture is in-deed independent of specific coarse segmentation masks, showing good robustness to coarse-grained crack features from different sources. Nevertheless, comparing the final experimental results with the best refinement predictions guided by coarse segmentation masks generated by Swin Transformer in the PointRend group reveals that the method using the probability map for rendering point guidance further improves the accuracy of segmentation results. Notably, the most significant improvement is observed in mBA, more than doubling the increases in IoU and Dice, reaching 12.42%. This outstanding robust performance is largely due to the probability map's ability to reflect the probability of pixels belonging to each category, rather than merely showing simple category membership like coarse segmentation masks. This probabilistic information provides a continuous confidence measure for each pixel, rather than categorically assigning pixels in a binary fashion, thereby preserving ample information on tiny cracks and crack boundaries. Such details allow for the discovery and refinement rendering of these pixels during the inference stage through point-by-point reasoning with the MLP. To further substantiate the validity of these conclusions, Figure 4 visualizes the prediction results under all comparative methods for five randomly selected HR crack images from the test set. The visualizations demonstrate that the predictive masks obtained by the probability map-guided rendering point method outperform those obtained by any coarse segmentation-guided method in recognizing crack edges and tiny cracks, thereby further validating the conclusions drawn from the quantitative results.

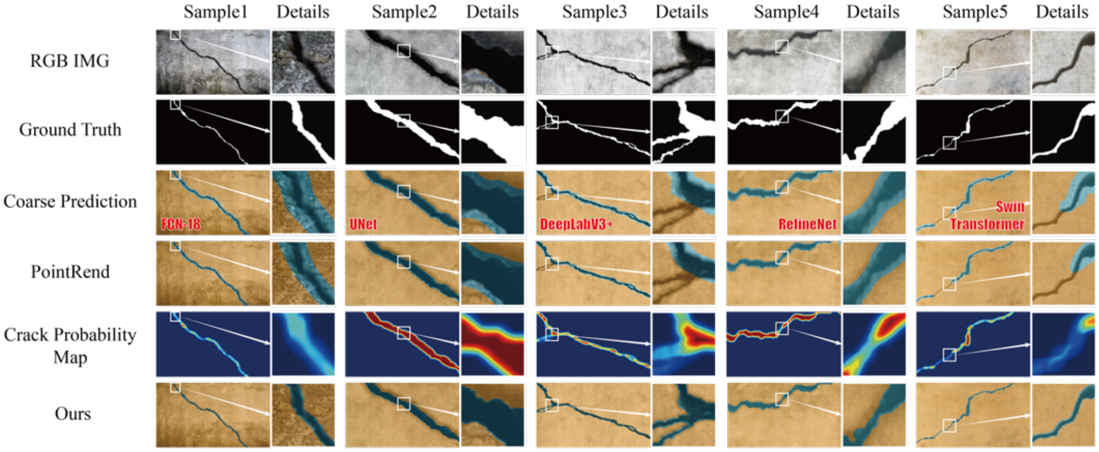

Figure 4. Visualization of fine-grained segmentation results of the PointRend architecture guided by different coarse segmentation masks and the proposed method guided by the probability map.

5 Conclusion

This study pioneers the application of rendering techniques from computer graphics to HR crack image segmentation, enhancing the fine-grained segmentation of crack im-ages through two novel modifications. Initially, a Pyramid Transformer Block with Spatial Relationship Attention (SRA) is implemented within the encoding architecture for effective extraction of deep, detailed crack feature maps. Additionally, a tailored rendering point guidance strategy in the decoder concentrates computational resources on the crack edges and tiny cracks, increasing pixel recognition accuracy for these hard samples. Demonstrated performance on HR crack images confirms this approach as the latest benchmark in the field.

6 Acknowledgments

This work is supported by the Europe Commission project D-HYDROFLEX (No. 101122357), Europe Commission project INHERIT (No. 101123326), the National Natural Science Foundation of China (No. 52278177) and the National Key Research and Development Program of China (No. 2023YFC3806800)

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.

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

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

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. (2017), Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected Crfs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834-848.

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*, Seattle, USA, pp. 8890-8899.

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.

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

Dias, P. A. & Medeiros, H. (2018), Semantic Segmentation Refinement by Monte Carlo Region Growing of High Confidence Detections, *Asian Conference on Computer Vision*, Springer, Perth, Australia, pp. 131-146.

Guo, F., Qian, Y., Liu, J. & Yu, H. (2023), Pavement Crack Detection Based on Transformer Network, *Automation in Construction*, 145, 104646.

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.

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*, virtual online, pp. 4287-4297.

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.

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.

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*, Honolulu, USA, pp. 1925-1934.

Lin, G., Shen, C., Van Den Hengel, A. & Reid, I. (2016), Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 3194-3203.

Liu, W., Huang, Y., Li, Y. & Chen, Q. (2019), Fpcnet: Fast Pavement Crack Detection Network Based on Encoder-Decoder Architecture, pp. arXiv:1907.02248.

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.

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*, Honolulu, USA, pp. 4353-4361.

Qu, Z., Li, Y. & Zhou, Q. (2022), Crackt-Net: A Method of Convolutional Neural Network and Transformer for Crack Segmentation, *Journal of Electronic Imaging*, 31(2), 023040-023040.

Quan, J., Ge, B. & Wang, M. (2023), Crackvit: A Unified Cnn-Transformer Model for Pixel-Level

Crack Extraction, *Neural Computing and Applications*, 1-17.

Ronneberger, O., Fischer, P. & Brox, T. (2015), U-Net: Convolutional Networks for Biomedical Image Segmentation, *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, Munich, Germany, pp. 234-241.

Shamsabadi, E. A., Xu, C. & Dias-da-Costa, D. (2022), Robust Crack Detection in Masonry Structures with Transformers, *Measurement*, 200, 111590.

Sony, S., Dunphy, K., Sadhu, A. & Capretz, M. (2021), A Systematic Review of Convolutional Neural Network-Based Structural Condition Assessment Techniques, *Engineering Structures*, 226, 111347.

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*, Wiley Online Library, pp. 703-735.

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

Wang, Q., Wang, Z., Genova, K., Srinivasan, P. P., Zhou, H., Barron, J. T., Martin-Brualla, R., Snavely, N. & Funkhouser, T. (2021), Ibrnet: Learning Multi-View Image-Based Rendering, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, vitural online, pp. 4690-4699.

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*, vitural online, pp. 568-578.

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.

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

Zhang, C., Lin, G., Liu, F., Yao, R. & Shen, C. (2019), 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*, Long Beach, USA, pp. 5217-5226.

Zhang, J., Lu, C., Wang, J., Wang, L. & Yue, X.-G. (2019), Concrete Cracks Detection Based on Fcn with Dilated Convolution, *Applied Sciences*, 9(13), 2686.

Zhang, X., Rajan, D. & Story, B. (2019), Concrete Crack Detection Using Context‐Aware Deep Semantic Segmentation Network, *Computer‐Aided Civil and Infrastructure Engineering*, 34(11), 951-971.

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*, Santiago, Chile, pp. 1529-1537.