2024 European Conference on Computing in Construction
Chania, Crete, Greece
July 15-17, 2024

FINE-GRAINED SEGMENTATION OF HIGH-RESOLUTION BRIDGE CRACK IMAGES USING RENDERING TECHNOLOGY

Honghu Chu[1,2], Weiwei Chen[1,3*], Lu Deng[2]
[1] Bartlett School of Sustainable Construction, University College London, London, UK
[2] College of Civil Engineering, Hunan University, Changsha, China
[3]Civil Engineering Division, Engineering Department, University of Cambridge, CB3 0FA, UK

Abstract

Drawing on insights from computer graphics, this study introduces the Crack Boundary Point Rendering Network (CBPRN), an innovative high-resolution (HR) image segmentation framework designed to improve UAV-based bridge crack inspections. We developed an edge-guided branch and an uneven sampling strategy, enhancing detail preservation on crack boundary areas effectively. Through comprehensive ablation experiments, the efficacy of the CBPRN was validated, demonstrating its superior performance with remarkable outcomes: a processing speed of 13.45 FPS and mIoU, mBA, and Dice scores of 87.23%, 93.56%, and 89.59%, respectively, for images beyond 2K resolution. The CBPRN establishes a new standard in HR crack image segmentation.

Introduction

The emergence of surface cracks can effectively reflect the recent load-bearing status of bridges and provide a strong reference for traffic management departments to make reasonable maintenance decisions, thus preventing potential catastrophes (Manjunatha et al., 2023, Park et al., 2019). Therefore, accurate and efficient crack detection is crucial for ensuring the bridge's safety during its service life. In recent years, advancements in crack detection technology, driven by digital image processing algorithms, have seen rapid progress. This development has markedly enhanced efficiency and reduced the high costs traditionally associated with manual inspection methods (Çelik and König, 2022). Presently, extensive research has been undertaken by researchers in the field of crack detection utilizing image processing methods (Munawar et al., 2021, Ren et al., 2020). Within this domain, deep learning (DL)-based crack segmentation algorithms have garnered significant interest over classification and object detection algorithms. This preference is due to the segmentation algorithms' superior ability to accurately delineate the contours and shape characteristics of cracks with pixel-level precision.

A considerable amount of research based on DL for crack segmentation has been conducted, with some advanced algorithms reporting an impressive mIoU of over 93% on certain open-source crack datasets (Yang et al., 2023, Ali et al., 2021). However, it is noteworthy that these studies and their corresponding algorithmic improvements have primarily focused on identifying the main body of cracks while neglecting the recognition of fine-grained representations at crack boundaries. This oversight is significant because the quality of mask boundaries plays a crucial role in image segmentation; precise object segmentation directly benefits various downstream applications, such as damage quantification and assessment (Liu et al., 2023, Li et al., 2017). To comprehend the limitations in achieving fine-grained representations at crack boundaries, a critical analysis of the supervisory principles governing these segmentation algorithms is required, with a particular focus on the Mask Intersection-over-Union (Mask IoU) loss function (Cheng et al., 2021). This loss function, a benchmark in model training, guides models in predicting masks at pixel wise. Under the supervision of Mask IoU loss, models strive to maximize the overlap between the predicted mask and the actual mask. However, this loss evaluates all pixels equally, both internal and boundary pixels, making it less sensitive to the boundary quality of coarser cracks. As the crack size increases, the number of internal pixels grows quadratically and can far exceed the linearly increasing number of boundary pixels. This discrepancy leads to ambiguous segmentation effects in crack boundaries, especially at higher image resolutions where the difference between crack boundary and main body pixels is more pronounced. This trend is problematic for the industry's shift towards HR imaging for crack detection, as it can significantly affect segmentation results and impede accurate structural safety assessments. Therefore, a systematic study of the fine-grained recognition of crack edges is necessary to address this challenging issue.

To this end, this study introduces the Crack Boundary Point Rendering Network (CBPRN), which enhances three key components of traditional point-based rendering architecture, enabling the rendering head to fully utilize its advantages in refined segmentation of cracks. The network architecture is illustrated in Figure 1. Initially, an edge-guided branch based on a super-resolution encoding architecture was designed, ensuring that details of crack boundaries and related tiny crack information are adequately preserved in the deep semantic feature maps used as a source for refined rendering. For the second improvement, an uneven sampling strategy focusing on

boundary areas was developed for the rendering-based prediction head, allowing the network to concentrate computational power on challenging areas like crack boundaries. Overall, these two improvements fully leverage the advantages of graphic rendering methods in the fine-grained segmentation of crack images. The CBPRN achieves a good balance between computational resource consumption and practical deployability while providing crack image outputs with precise boundaries, which is significant for accurate structural safety assessments of bridges.

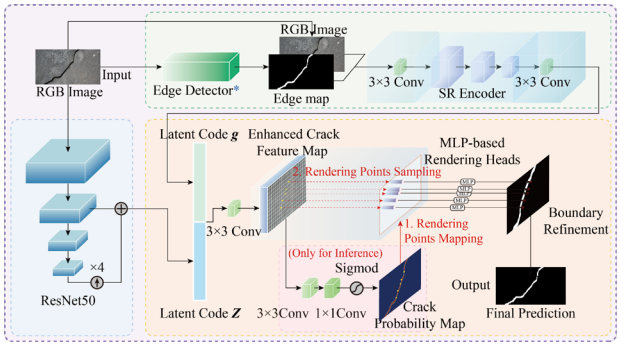

Figure 1: Visualization of boundary region-guided sampling with different dilation coefficients during the training phase

Methodology

The CBPRN proposed in this study consists of three main components: a coarse crack segmentation feature extraction backbone, an edge-guided branch, and a point rendering-based fine-grained prediction head. The coarse crack segmentation feature extraction backbone is built on a ResNet50-based encoding architecture, designed to perform coarse-grained feature extraction from high-resolution crack images. The edge-guided branch comprises a fixed-parameter edge detector and a super-resolution image encoding architecture, which guide the edge details in high-dimensional implicit features for coarse-grained crack features. The fine-grained prediction head based on point rendering primarily aggregates the implicit crack features outputted from the first two components and restores the fine-grained edge details of the cracks through point-by-point refined rendering based on the shared-weight multi-layer perceptron (MLP). Figure 1 visually presents some algorithmic details and computational logic of the proposed CBPRN. The edge-guided branch and the fine-grained prediction head constitute two innovative enhancements to the original PointRend architecture, respectively, and are detailed in the following subsections.

Edge-guided Branch

To ensure that the deep semantic feature maps of cracks sufficiently retain the details of crack boundaries and minute cracks for refined rendering, this study customizes an edge-guided branch in addition to the coarse crack segmentation feature extraction backbone. In fact, previous research has utilized guided image filtering (He et al., 2012) for boundary guidance in natural scene image segmentation, an effective edge-preserving smoothing operator based on guided images. However, edge recognition methods based solely on morphological operations are easily disturbed by environmental noise like cracks and struggle to accurately extract edges of small-sized cracks. To address this, the authors retain the fixed-parameter guided image filtering operator while introducing an encoder designed for super-resolution reconstruction tasks, aiming to eliminate noise interference in the boundary feature map while enhancing the representation of minute crack boundary features. This super-resolution encoding architecture, as shown in Figure 2, is primarily constructed using three residual modules. To improve the extraction performance of tiny crack details, the authors incorporate a transformer module in each residual module (b1, b2, b3), hoping to model long-distance dependencies for crack pixels scattered across the global view through the inherent self-attention mechanism of the transformer.

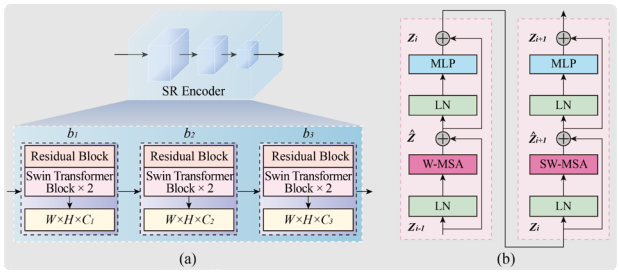

Figure 2: Details of the super-resolution reconstruction encoding architecture and some internal components in the edge-guided branch, (a) implementation details of the super-resolution reconstruction encoding architecture, (b) implementation details of two consecutive Swin Transformer blocks

Fine-Grained Prediction Head Based on Point Rendering

Inspired by refined rendering methods in computer graphics, the authors propose a crack fine-grained feature prediction decoding architecture based on point rendering. It is important to note that the principle of boundary refinement rendering point sampling remains consistent with the original PointRend architecture (Kirillov et al., 2020), ensuring computational efficiency during the training phase while enabling the trained model to perform effective end-to-end inference. Therefore, two different point sampling methods are designed for model training and inference.

For the training phase: due to the availability of precise pixel-level labels for effective supervision of prediction results, selecting boundary points for adequate sampling based on these refined pixel-level labels is a more accurate and computationally efficient method compared to the original approach in PointRend, which involved boundary prediction based on the most uncertain points. Therefore, during the training phase, boundary information is directly extracted from the refined pixel-level labels of crack images to guide the selection of sampling points. Specifically, an edge detection algorithm is used to extract

the edge areas of the crack labels, and some of the sampling points, originally uniformly distributed across the background and the main body of the crack, are concentrated in these extracted edge areas. It is important to note that to prevent a decline in model performance due to an imbalance in the ratio of positive to negative samples during training, and to ensure efficient training, the total number of sampling points per image is set to $N = \frac{H \times W}{20}$. These sampling points are randomly distributed in the crack body, crack boundary, and background areas in a ratio of 0.3:0.4:0.3. Figure 3 visually demonstrates the sampling strategy for the rendering points used in training the model. Ultimately, all the sampling points identified on the labels are mapped onto the corresponding enhanced crack feature maps for model training.

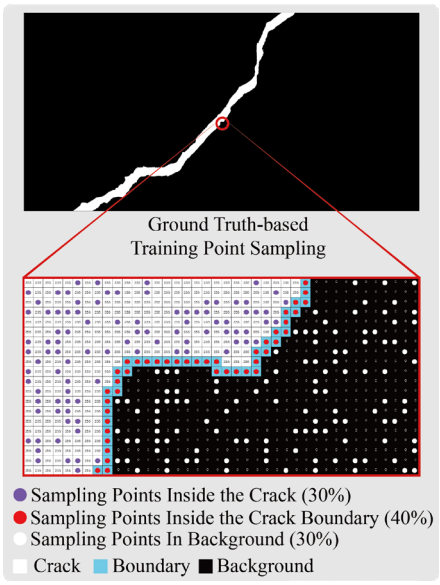

Figure 3: Visualization of the sampling points selection strategy in the training phase for the point rendering-based prediction head

For the inference phase: Refined labels, which are the ultimate prediction results, are not available at the beginning of the inference phase, hence the sampling method used during the training phase cannot be applied. To effectively focus computational resources on accurately predicting minute cracks and crack boundaries, this study proposes a boundary-guided rendering point sampling strategy based on a probabilistic heatmap. Specifically, two convolutional modules are added to the enhanced crack feature map to generate a refined crack pixel probability heatmap. Rendering points are guided by identifying pixels with high uncertainty in predicted values on the probability heatmap. Based on the probability, pixels on the heatmap can be roughly divided into three areas: pixels with probabilities close to 0 and 1 represent background and crack body, which are easily recognized by the network; pixels with probabilities around 0.5 represent areas of minute cracks or boundaries that are difficult for the model to determine with certainty. During the inference phase, for pixel areas on the probability heatmap with probabilities close to 0 and 1, the corresponding background and crack labels are directly used to represent the final prediction values, eliminating the need for further refined rendering. However, for pixel areas with probabilities around 0.5, refined rendering is required to further refine these ambiguous predictions. The refined rendering points are uniformly distributed in these hard-to-identify pixel areas. It is important to note that the probabilistic heatmap is used instead of the coarse segmentation results from the original PointRend architecture because the enhanced crack feature map used to obtain the heatmap is unified in size according to the second block of ResNet. It retains more details of minute cracks with only one downsampling compared to the original coarse segmentation and requires less computational resources than coarse segmentation. To visually represent the refined rendering point sampling method during the inference phase, Figure 4 displays a visualization of a randomly selected probabilistic heatmap example. On the heatmap, probabilities in the background and main body of the crack are concentrated near 0 and 1, respectively, while in the boundary area, due to issues like manual annotation errors and insignificant color differences, the probabilities of pixels fluctuate around 0.5. This study sets the probability range for these hard-to-identify pixels between 0.3 and 0.7, and in the subsequent refined rendering phase, only samples with probabilities between 0.3 and 0.7 undergo refined inference. The parameter settings for sampling points during the training and inference phases will be detailed in subsection 3.3.2.

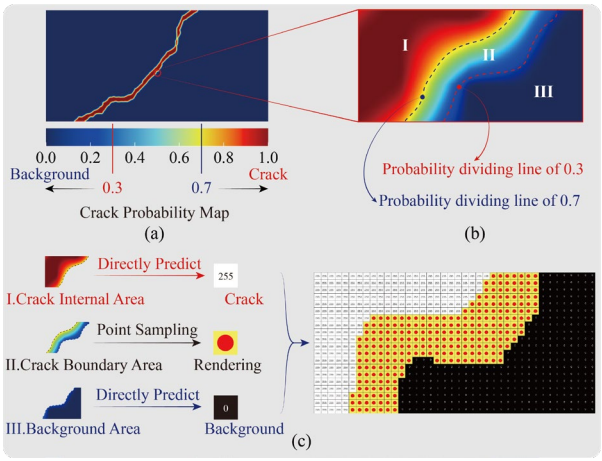

Figure 4: Visualization of the sampling points that require refined rendering during the inference phase

Experimental Setup and Results

Dataset

The CFD (Shi et al., 2016), Crack500 (Yang et al., 2019), and Deepcrack537 (Zhou et al., 2022) were chosen for model training. The crack images in these three datasets almost encompass most crack forms in engineering structures and include crack samples collected under different lighting conditions, which is beneficial for enhancing the model's robustness. Importantly, these

three datasets all have finely annotated pixel-level labels, which are helpful for accurately evaluating the advantages of the algorithm proposed in this study in terms of boundary refinement segmentation. It is noteworthy that before training the model, images from the three datasets were uniformly resized to 256 × 256 pixels to facilitate uniform training input and save computational resources required for training. In total, 1200 resized crack images from the three open-source datasets were used, including 900 training samples, 150 validation samples, and 150 test samples.

Furthermore, a total of 60 crack images were collected in the urban area of Changsha, as shown in Figure 5, to test the segmentation performance of CBPRN on HR crack images.

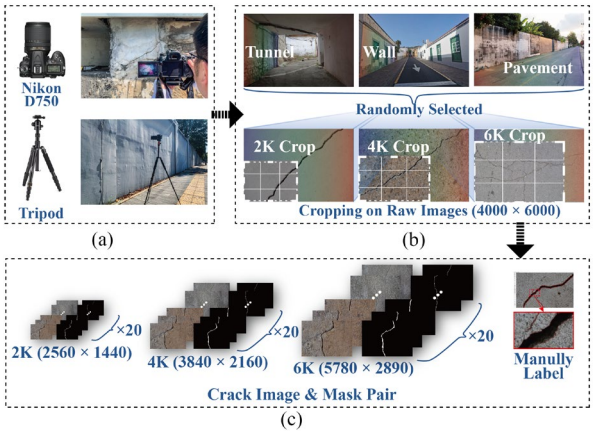

Figure 5: Details of establishing the high-resolution crack image dataset with pixel-level annotations

Implementation Details and Metrics

Training Hyperparameters: The experiments were conducted on a system equipped with an i9-9820X CPU and two NVIDIA RTX 3090 Ti GPUs, running Ubuntu 20.04, and the network model was implemented within the PyTorch framework. The total number of iterations was set to 1000 epochs, with a batch size of 16. The initial learning rate was set at 0.01, using a warming-up strategy for the first 100 epochs followed by a poly learning rate decay strategy with a decay rate of 0.9. Additionally, to ensure the global optimum of the loss function can be obtained during training, Adam optimizer, which combines the advantages of momentum and RMSprop, was used with a momentum of 0.9 and a weight decay of 1×10^{-4}. With these parameters, the preliminary model trained on the low-resolution open-source crack dataset was further fine-tuned using field-collected crack images.

Evaluation Metrics: Two common metrics were chosen for quantitative assessment of the experimental results: mean Intersection over Union (mIoU) and Dice Similarity Coefficient (Dice). Additionally, to highlight the performance of the proposed method in boundary areas, Mean Boundary Accuracy (mBA) was used as an additional evaluation metric. The core concept of mBA involves calculating the IoU between the Ground Truth (GT) and the predicted mask within the boundary area (Cheng et al., 2020).

Ablation Study

Ablation study for the point rendering-based prediction head: To fully illustrate the advantages of performing fine-grained crack mask using the point rendering method, a performance comparison was first made between the proposed decoding architecture and the traditional decoding architecture based on multiple convolutional layers and upsampling operations. The relevant experimental results are listed in Table 1. From the first two rows of Table 1, it can be seen that the point rendering-based decoding architecture has achieved improvements in mIoU, Dice, and mBA compared to the traditional decoding architecture, with the most significant improvement observed in mBA, reaching 86.78%. This is because the MLP in the point rendering-based decoding architecture is position-sensitive, calculating the prediction value for each pixel independently. Therefore, it can flexibly capture details and spatial relationships in crack images. In contrast, the traditional upsampling-based decoding architecture is limited by discrete feature sampling and struggles to capture local crack details.

After demonstrating the superiority of the proposed decoding architecture, parameter performance experiments need to be conducted to obtain the optimal parameters matching the model architecture. Since the training and inference stages use distinct fine-grained rendering point sampling methods, as described in Section 2.2, the following two sets of parameter performance experiments are conducted to obtain relatively optimal point sampling parameters for both the training and inference processes.

Point sampling study in the training phase: It is necessary to emphasize again before conducting parameter experiments that the regions most likely to have erroneous predictions are mainly concentrated at the boundaries and their adjacent areas, because these regions often have colors and contrasts similar to those of nearby cracks in RGB images. However, if the training point sampling is carried out as shown in Figure 3, where only a one-pixel-wide edge area is used for boundary guidance, it may not be possible to avoid the biasing guidance caused by the subjective nature of human annotation, resulting in errors between the true boundary and the labeled boundary. To avoid the negative impact of erroneous guidance on the model during training, it is necessary to expand the guided boundaries. Specifically, in this study, simple morphological operations were used, with the outermost pixels of the crack label as the center of dilation, and dilation operations were performed with the same dilation factor towards the background area and inside the crack. Considering the image size and the average pixel width of cracks in the training dataset, four different edge dilation coefficients were used to expand the boundary region. As shown in Figure 6, the widths of

the expanded boundary regions (yellow outlined areas) after dilation are 1, 3, 5, and 7, respectively. The total number of sampling points on each training image is $N = \frac{H \times W}{20}$, distributed randomly in the background, expanded crack edges, and inside the cracks at proportions of 30%, 40%, and 30%, respectively.

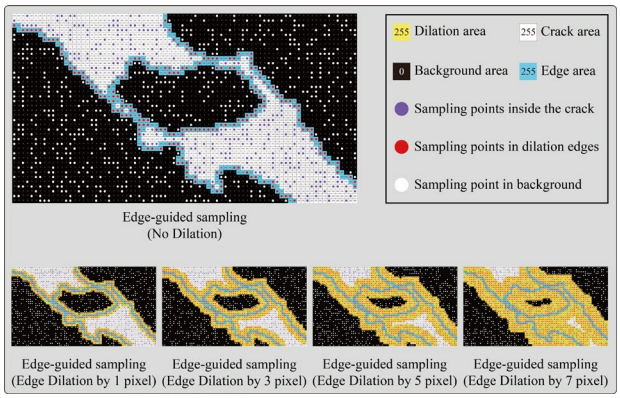

Figure 6: Visualization of boundary region-guided sampling with different dilation coefficients during the training phase

The set No.2 to set No.5 in Table 1 provide statistics on the performance of the model, which was trained in the training phase using four different widths of expanded boundaries for guiding sampling points. It can be observed that the model performs best when the dilation coefficient is 2 (corresponding to a boundary region width of 5), with mIoU, mBA, and Dice reaching 86.49%, 90.52%, and 88.54%, respectively. This best performance occurs due to the fact that the manual labeling bias range in the crack training dataset used in this study falls within this interval. Combining the experimental results in Table 2 with the visual effects of the probability heatmaps in Figure 4, the following conclusions can be drawn: dilation coefficients that are too low (0 or 1) or too high (3) result in improved performance compared to no dilation, but they respectively lead to insignificant improvements due to the inability to fully encompass error boundaries in the sampling region or the dispersion of computational resources beyond the error interval. Specifically, when the dilation coefficient is 0 or 1, the dilated boundary region is not sufficient to encompass the bias generated near the boundary during manual labeling. When the dilation coefficient is 3, the crack's main region and too many background regions without artificial labeling errors are included as ambiguous boundary regions requiring fine sampling. These unnecessary simple sample regions take away computational resources that should belong to the ambiguous boundary regions, thereby reducing the model's learning and representation capabilities for ambiguous boundary regions, resulting in a very limited improvement in recognition accuracy brought by boundary-guided sampling.

Point sampling study in the inference stage: In order to effectively improve the model's fine-grained inference performance in the boundary regions while saving computational efficiency, it is necessary to determine reasonable probability intervals for areas with uncertain prediction results concentrated around 0.5 on the probability heatmaps. A larger probability interval implies the need to sample more probability points for fine-grained rendering, which increases precision but significantly increases computational redundancy in the inference process. Conversely, a too small probability interval, while speeding up inference, may lead to ineffective fine-grained rendering of many tiny cracks and boundary details, seriously affecting the final recognition accuracy.

Specifically, two probability parameters need to be set: the critical probability value α between background pixels and boundary regions on the coarse prediction probability map, and the critical probability value β between boundary regions and crack pixels. For the critical probability value α between background and boundary regions, this study sets three different probability parameters: 0.2, 0.3, and 0.4. Similarly, for the critical probability value β between boundary regions and crack pixels, three different probability parameters are also set: 0.6, 0.7, and 0.8. By defining these nine different probability interval ranges based on the two types of critical probability values, the boundary regions are categorized. Table 2 provides statistics on the inference results of the models that use these 9 different probability intervals for sampling on the test dataset.

From Table 2, it can be seen that the models from Set No. 4, 5, and 6 (i.e., background region probability range between 0 and 0.3) achieved relatively better accuracy than other models. This is because, compared to the sampling groups with the background region probability range set between 0 and 0.4, the sampling methods under these three parameter settings encompass a wider background sampling area, which is more helpful in repairing some tiny crack details that were not detected in the background. At the same time, the sampling groups with the background region probability range set between 0.0 and 0.2 classified too many pixels originally belonging to the boundary region as background pixels, causing ambiguous boundary regions to be unable to achieve precise boundary detail repair due to insufficient sampling points, thus resulting in relatively lower mBA.

In addition, by comparing the model performance from Set No. 4 to 6, it can be observed that when the critical probability of crack internal area is set to 0.7, the model's inference accuracy is the highest, with mIoU, Dice, and mBA reaching 87.23%, 93.56%, and 89.59%, respectively.

Finally, the sampling parameter configuration set by Set No. 6 is adopted as the optimal inference stage sampling parameters to control the model's subsequent experiments.

Performance Comparison between CBPRN and the Traditional PointRend Model

Considering that the CBPRN is built based on the original PointRend model and is specifically designed for crack

segmentation, with the main improvement being the introduction of a fine-grained boundary point rendering sampling method based on the fine-grained probability heatmap in the inference phase. In order to further demonstrate the effectiveness of this approach, which involves fine-grained point sampling guidance based on the probability heatmap in the inference phase, in comparison to the traditional approach of fine-grained point sampling guidance based on coarse segmentation, this section compares the segmentation results of the original PointRend using different sources of coarse segmentation on high-resolution images collected in Section 3.1 with the corresponding results obtained by the method proposed in this study.

Specifically, the authors selected five mainstream deep learning segmentation architectures with varying levels of segmentation accuracy, including FCN-18, UNet, DeepLabV3+, PSPNet, and RefineNet, as the generating networks for the coarse segmentation required when the original PointRend architecture performs predictions. In contrast, the method proposed in this study uses probability heatmaps proposed based on enhanced crack features extracted by the encoder and the boundary guidance branch to perform boundary point sampling guidance for the fine-grained prediction head based on point rendering.

It is worth noting that all coarse segmentation architectures and fine-grained segmentation networks were trained with default optimal parameters in the same deep learning framework with the same configuration. Additionally, when using the trained coarse segmentation models for prediction, all high-resolution images were proportionally scaled down to have a long edge of 900 pixels to avoid issues related to GPU memory overflow caused by excessively high original resolutions.

The experimental results are shown in Table 3. From the table, it can be observed that there are significant differences in the prediction results generated by different coarse segmentation mask generation architectures, with differences in mIoU, mBA, and Dice ranging from 2.92%, 3.06%, to 2.36%, from the lowest accuracy of FCN-18 to the highest accuracy of RefineNet. However, after applying the original PointRend model for the refinement process, the differences between the fine-grained prediction results become less pronounced, with mIoU, mBA, and Dice for all five experimental groups fluctuating within the intervals of 83.30±0.09%, 87.14%±0.16%, and 85.27±0.06%, respectively. These results indicate that the PointRend refined segmentation method is indeed independent of specific coarse segmentation masks and exhibits good robustness to coarse-grained features from different coarse segmentation architectures. However, when comparing the final experimental results with the best-performing coarse segmentation results based on RefineNet within the PointRend group, it can be observed that the method guided by probability heatmaps further improves the segmentation accuracy. It can be noted that among the three improved metrics, mBA shows the most significant improvement, which is more than twice the improvement in mIoU and Dice, reaching 6.27%. This outstanding robust performance is largely attributed to the introduction of the edge-guided branch in the feature extraction stage of this study, which preserves sufficient information about small cracks and crack boundaries in the enhanced crack features used to generate probability heatmaps. This enables the pixels in these detail areas to be detected during inference and finely represented through point-wise dense rendering. Additionally, it should be noted that CBPRN, by avoiding the guidance of coarse segmentation from external sources and based on a customized non-uniform inference point sampling method, surpasses the original PointRend in inference speed by more than twice on average, achieving 13.45 FPS. To further demonstrate the effectiveness of the above conclusions, Figure 7 provides visualizations of the test results for five randomly selected high-resolution crack images collected in the field. It is evident that the inference model guided by probability heatmaps proposed in this study outperforms any fine-grained rendering method guided by coarse segmentation masks in terms of crack edge recognition accuracy and sensitivity to tiny cracks.

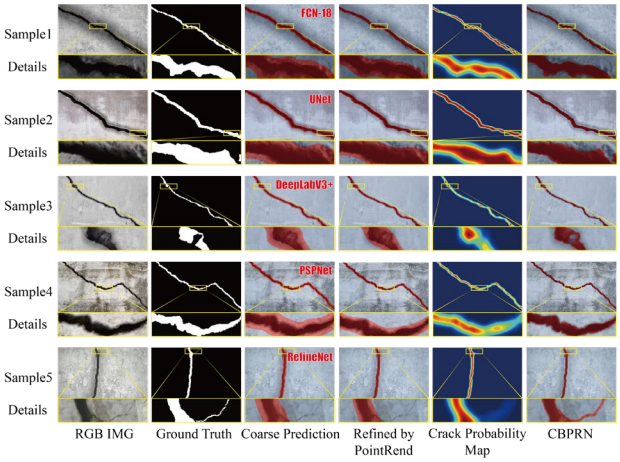

Figure 7: Visualization of fine-grained segmentation results of the PointRend architecture guided by different coarse segmentation masks and CBPRN guided by probability heatmaps

Conclusions and Future Work

In this study, a HR crack image fine-grained segmentation architecture named CBPRN is proposed. For the first time, rendering techniques from computer graphics are introduced into HR crack image segmentation tasks. Through three customized improvements, the originally designed point rendering technique for natural scene objects is adapted to effectively perform crack segmentation with fine-grained boundaries. The proficiently trained CBPRN attains a remarkable inference speed of 13.45 FPS, yielding mIoU and mBA scores of 87.23% and 93.56%, respectively, along with a Dice score of 89.59%. This performance was demonstrated on crack images exceeding 2K resolution,

thereby establishing CBPRN as the current state-of-the-art benchmark in this domain.

In future, the implementation of model pruning and quantization techniques will be advanced to facilitate the lightweight deployment of CBPRN on the UAV, aiming to provide bridge maintenance departments with a more reliable and secure method for conducting bridge crack detection in practical engineering scenarios. Additionally, this method can be adopted to the hydropower projects, similarly to detect the defects of dam structures.

Table 1: Performance comparison of traditional decoding architecture and the proposed point-rendering-based fine-grained prediction head in models trained with different parameterized feature point sampling strategies

Set No.	Decoding architecture	Sampling point extraction method for the training phase	Dilating coefficient	Width of the boundary area after dilating	IoU(%)	mBA(%)	Dice(%)
1	Traditional convolution and upsampling operations	Uniform sampling	/	/	85.54	88.67	87.31
2	Fine-grained prediction head based on point rendering	Boundary guided sampling	0	1	85.98	89.60	87.76
3			1	3	86.12	89.94	88.03
4			2	5	86.49	90.52	88.54
5			3	7	85.76	89.09	87.50

Table 2: Comparison of inference performance obtained by different boundary probability ranges

Set No.	Probability range for background area	Probability range for boundary area	Probability range for crack internal area	IoU(%)	Dice(%)	mBA(%)
1	（0.0,0.2）	（0.2,0.6）	（0.6,1.0）	86.53	90.69	88.58
2	（0.0,0.2）	（0.2,0.7）	（0.7,1.0）	86.98	92.47	89.12
3	（0.0,0.2）	（0.2,0.8）	（0.8,1.0）	86.49	90.52	88.54
4	（0.0,0.3）	（0.3,0.6）	（0.6,1.0）	86.78	91.12	88.80
5	（0.0,0.3）	（0.3,0.7）	（0.7,1.0）	87.23	93.56	89.59
6	（0.0,0.3）	（0.3,0.8）	（0.8,1.0）	86.61	90.98	88.91
7	（0.0,0.4）	（0.4,0.6）	（0.6,1.0）	86.51	90.58	88.49
8	（0.0,0.4）	（0.4,0.7）	（0.7,1.0）	86.89	92.12	88.99
9	（0.0,0.4）	（0.4,0.8）	（0.8,1.0）	85.89	89.76	87.87

Table 3: Comparison of the refined segmentation results on HR images collected onsite between the proposed probability heatmap-guided method and the original pointrend architecture guided by different coarse segmentation masks

Meticulous segmentation architecture	Source of the boundary sampling guidance		Coarse segmentation accuracy (%)			Refined segmentation accuracy (%)			Total inference speed
			mIoU	mBA	Dice	mIoU	mBA	Dice	FPS
PointRend	Coarse segmentation guidance	FCN-18	78.36	79.25	81.37	83.21	86.98	85.21	7.83
		UNet	79.47	80.09	82.46	83.30	87.11	85.24	5.77
		DeepLabV3+	80.36	80.34	82.60	83.33	87.14	85.28	3.65
		PSPNet	80.65	81.79	82.88	83.37	87.26	85.31	4.21
		RefineNet	81.28	82.31	83.73	83.38	87.29	85.33	3.49
CBPRN	Probability heat map guidance	Probability interval ∈ [0.3,0.7]		/		87.23	93.56	89.59	13.45

Acknowledgements

This work was supported by the China Scholarship Council (No. 202206130068). This work is also supported by Horizon Europe projects, D-HYDROFLEX (Project No.:101122357). INHERIT (Project No.: 101123326), and UCL-ZJU Strategic Partner Fund

References

ALI, R., CHUAH, J. H., TALIP, M. S. A., MOKHTAR, N. & SHOAIB, M. A. (2021) Automatic pixel-level crack segmentation in images using fully convolutional neural network based on residual blocks and pixel local weights. Engineering Applications of Artificial Intelligence, 104, p.pp.104391.

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, p.pp.1875-1890.

CHENG, B., GIRSHICK, R., DOLLÁR, P., BERG, A. C. & KIRILLOV, A. (2021) Boundary IoU: Improving object-centric image segmentation evaluation. Available: https://ui.adsabs.harvard.edu/abs/2021arXiv210316562C [Accessed March 01, 2021].

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. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020. Seattle, Washington, p.pp.8890-8899.

HE, K., SUN, J. & TANG, X. (2012) Guided image filtering. IEEE transactions on pattern analysis and machine intelligence, 35, p.pp.1397-1409.

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 2020. Seattle, Washington, p.pp.9799-9808.

LI, S., CAO, Y. & CAI, H. (2017) Automatic pavement-crack detection and segmentation based on steerable matched filtering and an active contour model. Journal of Computing in Civil Engineering, 31, p.pp.04017045.

LIU, H., YANG, J., MIAO, X., MERTZ, C. & KONG, H. (2023) CrackFormer network for pavement crack segmentation. IEEE Transactions on Intelligent Transportation Systems, 24, p.pp.9240-9252.

MANJUNATHA, P., MASRI, S. F., NAKANO, A. & WELLFORD, L. C. (2024) CrackDenseLinkNet: A deep convolutional neural network for semantic segmentation of cracks on concrete surface images. Structural Health Monitoring, 23, p.pp.796-817.

MUNAWAR, H. S., HAMMAD, A. W., HADDAD, A., SOARES, C. A. P. & WALLER, S. T. (2021) Image-based crack detection methods: A review. Infrastructures, 6, p.pp.1-20.

PARK, S., BANG, S., KIM, H. & KIM, H. (2019) Patch-based crack detection in black box images using convolutional neural networks. Journal of Computing in Civil Engineering, 33, p.pp.04019017.

REN, Y., HUANG, J., HONG, Z., LU, W., YIN, J., ZOU, L. & SHEN, X. (2020) Image-based concrete crack detection in tunnels using deep fully convolutional networks. Construction and Building Materials, 234, p.pp.117367.

SHI, Y., CUI, L., QI, Z., MENG, F. & CHEN, Z. (2016) Automatic road crack detection using random structured forests. IEEE Transactions on Intelligent Transportation Systems, 17, p.pp.3434-3445.

YANG, F., ZHANG, L., YU, S., PROKHOROV, D., MEI, X. & LING, H. (2019) Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. Available: https://ui.adsabs.harvard.edu/abs/2019arXiv190106340Y [Accessed January 01, 2019].

YANG, L., BAI, S., LIU, Y. & YU, H. (2023) Multi-scale triple-attention network for pixelwise crack segmentation. Automation in Construction, 150, p.pp.104853.

ZHOU, Q., QU, Z., WANG, S.-Y. & BAO, K.-H. (2022) A method of potentially promising network for crack detection with enhanced convolution and dynamic feature fusion. IEEE Transactions on Intelligent Transportation Systems, 23, p.pp.18736-18745.