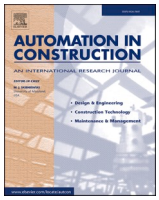
Automation in Construction 162 (2024) 105410

Contents lists available at ScienceDirect

Automation in Construction

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



Cascade refinement extraction network with active boundary loss for segmentation of concrete cracks from high-resolution images

Lu Deng [a], Huaqing Yuan [a], Lizhi Long [a], Pang-jo Chun [c], Weiwei Chen [b], Honghu Chu [a,b,*]

[a] College of Civil Engineering, Hunan University, Changsha, China
[b] The Bartlett school of sustainable construction, University College London, London, UK
[c] Department of Civil Engineering, The University of Tokyo, Tokyo, Japan

ARTICLE INFO

Keywords:
Deep learning
Crack segmentation
High resolution dataset
Multi-scale cascade operation
Boundary refinement
UAV bridge inspection

ABSTRACT

Accurate extraction of cracks is important yet challenging in bridge inspection, particularly that of tiny cracks captured from high-resolution (HR) images. This paper presents a crack-boundary refinement framework (CBRF) for meticulous segmentation of HR crack images. First, a triple-scale feature extraction module is designed to enhance the representation of miniscule-crack pixels. Then, a cascade operation involving global and local steps is adopted to conduct the refinement. In addition, an active boundary loss is introduced into the training process to solve the semantic inconsistency of crack boundary areas. The first HR crack image dataset is established to thoroughly evaluate the CBRF. Finally, an unmanned aerial vehicle (UAV)-based case study is conducted on the Yinpenling Bridge, which further confirms the practicality of the CBRF in improving the safety and efficiency of UAV-based bridge detection while ensuring the accuracy.

1. Introduction

In civil infrastructure, cracking is a predominant flaw [1–3] that may escalate under conditions such as heavy traffic, material degradation, and severe weather. As the most typical crack inspection method, manual visual inspection is critical for initial diagnosis and subsequent decision-making rehabilitation; however, it is a subjective and laborious method that requires specialized knowledge [4]. To accomplish initial inspections more efficiently, professionals and researchers have become increasingly interested in developing automated methods and the corresponding equipment.

Drawing inspiration from the automatic diagnosis of medical images [5], researchers have rigorously explored and applied traditional image processing techniques to crack detection over the past two decades. These techniques show the potential to replace the human eye to achieve high-accuracy and high-efficiency crack identification; however, they are typically sensitive to environmental noise (water spots, shadows, etc.) and are ineffective in eliminating stains caused by aging, paint splatter, and other factors [6].

Recent advancements in computational algorithms and high-performance computing [7] have notably enhanced machine learning (ML)-based approaches for crack detection, offering a more robust solution. This approach has garnered the interest of academics owing to its ability to learn and make decisions based on data with minimal human intervention. Several innovative applications of ML methods for crack pattern recognition have been documented in the literature. Among them, support vector machines [8,9] and artificial neural networks [10] are the two most prevalent techniques used in ML-based crack inspections. In addition, other classical ML techniques such as random forest and naive bayes have been employed [11,12]. All the approaches mentioned above primarily serve two purposes: crack separation and crack-type classification. Before performing these tasks, image-processing steps may be required to extract predefined features (i.e., moment features, entropy, colour, contour, edge, texture) such that crack identification from complex backgrounds is less challenging [13,14]. However, the typical ML methods used in these early studies involve only shallow learning techniques and require the manual adjustment of numerous thresholds; thus, they cannot be adapted to real-world images featuring complex backgrounds.

In contrast to the “shallow network” used in classical ML methods, deep learning (DL)-based methods with a “deep network” have been extensively used in both industry and academia owing to their wide adaptability to various real-world scenarios [15]. Among them, the deep convolutional neural network (DCNN) indicates significant potential for

* Corresponding author at: College of Civil Engineering, Hunan University, Changsha 410082, China.
E-mail address: chuhonghu@hnu.edu.cn (H. Chu).

https://doi.org/10.1016/j.autcon.2024.105410
Received 13 January 2023; Received in revised form 17 March 2024; Accepted 27 March 2024
Available online 13 April 2024

automated image processing. The term "deep network" implies that the DCNN is composed of multiple convolution and full connection layers, which allow the deep semantic information of images to be fully mined. Its advantages over previous image processing methods are primarily due to three reasons: i) it can immediately learn from raw input data and decide in an end-to-end manner, thus minimising human interference and preconceived notions; and ii) it can extract deep-level abstract features to establish deep connections that are difficult to identify behind the complex data source. The DCNN can be used to simulate the complex hierarchical cognitive laws of the visual cortex of the human brain and effectively capture the grid-like topology of a graph. Several DCNN-based image classification and object detection methods have been adopted in crack-inspection tasks and demonstrated excellent performance [16]. In crack classification tasks, standard backbone architectures such as VGGs [17], ResNets [18], GoogleNet [19], and DenseNet [20] have been applied in straightforward scenarios. For more complex situations, advanced architectures including ResNeXt [21], DarkNet [22], ShuffleNet [23], and DetNet [24] have demonstrated improved performance. In addition, some lightweight backbones, including SqueezeNet [25] and MobileNet [26], have been developed to facilitate the deployment of crack recognition models. For comprehensive crack localization within images, region-based detection algorithms have been extensively utilized or adapted for specific contexts. These algorithms include the Region-Based Convolutional Neural Network (R-CNN) [27], Spatial Pyramid Pooling Network [28], Fast R-CNN [29], Faster R-CNN [30], Feature Pyramid Network [31], You Only Look Once (YOLO) [32], RetinaNet [33], and EfficientDet [34]. Instead of identifying an entire image or an image patch, as performed in the methods above, using semantic DCNNs allows the "crack" or "non-crack" label for each pixel to be predicted, which is crucial for the safety assessment of the structures as better-detailed results are obtained [13,35]. Therefore, recent advancements in DCNN-based crack inspection have primarily focused on crack segmentation. The encoder-decoder architectures, including the SegNet [36], fully convolutional network [37], and UNet [38], are the most widely used basic architectures to achieve pixel-wise crack inspection, and almost all crack segmentation architectures were developed based on them. Unlike traditional semantic segmentation applied to natural scenes, crack segmentation presents three distinct challenges: 1. the proportion of positive and negative samples is highly unbalanced; 2. the background of the crack image is complex, and the texture is extremely rough; 3. the distribution of cracks is random, and the range of the crack scale changes widely. Notably, most studies focus on these three issues.

For the first problem, the typical solution is to introduce a loss function in the training phase to balance the positive and negative samples, such as focal loss [39] and Dice loss [40].

As for the background problem, some encoder-decoder-based networks with the embedment of hierarchical feature fusion and connected attention were customised [41], thus alleviating uneven strength from the complex background in certain scenes [42]. In addition, dense random fields and mixed pooling have been reported to be effective in accommodating the challenges associated with the complex background of long and sharp crack topologies in both pavement and tunnel scenes [43–45].

For the third issue, the multiscale supervised learning strategy and attention mechanism have been recognised as effective solutions. Xu et al. [46] proposed a new encoder-decoder structure holistically nested with a multiscale supervised learning strategy and a channel attention mechanism, which achieved better segmentation results than the basic architectures from two open-source datasets with widely varying crack sizes. Chu et al. [47] proposed the "Tiny-Crack-Net" multiscale feature fusion network with attention processes, which can be used to significantly enhance the capability of segmenting miniscule cracks. Qu et al. [48] introduced a deeply supervised convolutional neural network for crack detection, incorporating a novel multiscale convolutional feature fusion module. This module effectively captures the complex geometric structures of cracks, which are challenging to represent with single-scale features.

Despite the successful development of DCNN-based methods for crack detection, some issues remain unaddressed. As shown in Table 1, the image sizes of the current mainstream open-source crack datasets are extremely small. Most crack segmentation methods are established on these datasets and can only accommodate low-resolution (LR) crack images. Few studies have been conducted to address the problem of predicting meticulous masks for high-resolution (HR) crack images. Owing to the significant increase in the resolution of commodity cameras and displays, 4 K and even 6 K ultra-high definition images have gradually become the industry standard as they can provide more detailed information and allow more opportunities for forecasting the structural condition, planning maintenance or rehabilitation, allocating funding, etc. [49–51]. However, current DL-based semantic segmentation methods, tailored for LR images (e.g., the PASCAL [52] or COCO dataset [53]), struggle in HR settings due to computing limitations and scale discrepancies [54,55]. Notably, a model's GPU memory consumption is directly related to image resolution [56], making it impractical to train with HR images given limited resources. Additionally, applying models trained on LR images to HR ones introduces a scale gap, potentially reducing inference accuracy [57,58]. The scarcity of HR crack image datasets with precise pixel-wise annotations also hinders the development of segmentation algorithms for HR crack images [49,59]. Plausible solutions include downsampling and cropping [60–63]; however, the former removes some minor details, whereas the latter may destroy the context information of the image. Therefore, the potential of HR images for crack detection has not been fully exploited.

To overcome the challenges identified and provide an effective solution for precise segmentation of HR crack images, the authors referred to CascadePSP [64] and proposed a DL-based global and local integrated refinement crack extraction architecture named the crack-boundary refinement framework (CBRF). The proposed architecture was trained independently with the embedment of an active boundary loss (ABL) and can be added promptly to any current crack segmentation technique to improve the representative ability of the predicted crack mask. This implies that a finer and more accurate segmentation mask of the crack

Table 1
Summary of the existing open-source datasets for crack segmentation.

Dataset	No. Images	Resolution	Device	Scenes	Crack pixel (%)
CFD [65]	118	480 × 320	Iphone5	Pavement	1.6
Crack500 [66]	500	Around 2000 × 1500	LG-H345	Pavement	2.6
AEL [67]	58	311 × 462 to 700 × 1000	Car camera	Pavement	0.6
CrackLS315 [68]	315	512 × 512	Line-array camera	Pavement	0.2
Cracktree200 [69]	206	800 × 600	Area-array camera	Pavement	0.3
Cracktree260 [68]	260	800 × 600 & 960 × 720	Area-array camera	Pavement	0.4
CRKWH100 [68]	100	512 × 512	Line-array camera	Pavement	0.3
DeepCrack [70]	537	544 × 384	Unknow	Asphalt & concrete	3.5
EdmCrack600 [71]	600	1920 × 1080	GoPro Hero 7 black	Pavement	0.7
GAPS384 [72]	384	1920 × 1080	Camera	Pavement	0.3
Stone331 [68]	331	512 × 512	Area-array camera	Stone Surface	0.1

can be generated (particularly for HR crack images). Specifically, in the inference stage, the CBRF is input with a crack image and its initial mask, which can be an output of any conventional segmentation algorithm. Subsequently, the customised cascade structure in the CBRF can be used to allow the network to generate a redefined mask with fine crack boundaries for HR crack images in a coarse-to-fine manner without occupying a significant amount of graphics memory. Fig. 1 illustrates the capability of the proposed CBRF to refine and correct imprecise boundaries in coarse masks, which are outcomes of compromise methods such as cropping or downsampling high-resolution (HR) images. This demonstrates that the CBRF offers an effective solution for the precise segmentation of HR crack images.

The efficacy of the CBRF was validated via ablation studies on a self-compiled HR crack dataset, designated as HRCD-282. HRCD-282 comprises 282 on-site acquired crack images (including 100 with 2 K resolution, 100 with 4 K resolution, and 82 with 6 K resolution), and each image was finely annotated with a pixel-wise label by the authors. Finally, a case study was performed on an actual bridge to further confirm the practicality of deploying the CBRF on an unmanned aerial vehicle (UAV) for bridge crack inspection.

The main contributions of this study are threefold:

1. This paper introduces a universal CBRF that leverages coarse segmentation outcomes from various sources as preliminary data, thereby enhancing the precision of high-resolution (HR) crack image segmentation.

2. The first HR crack image dataset (abbreviated as the HRCD-282) is established and can be regarded as a benchmark dataset for evaluating future segmentation algorithms targeted towards HR crack images.

3. The CBRF offers a more reliable and efficient solution for UAV-based bridge crack inspection than traditional DL models limited to LR image processing.

2. Crack-boundary refinement framework (CBRF)

The CBRF was designed to meticulously characterise cracks on two levels. An encoder-decoder-based multi-scale refinement module was first proposed to produce three-scale refinement results for the coarse segmentation preliminary. Then two cascading operations with the embedment of the ABL were designed to generate detailed boundaries for these preliminary refinement results from both global and local aspects in a coarse-to-fine manner.

2.1. Multi-scale crack refinement module (MsCRM)

To refine the segmentation results, the capacity of the network to collect small-scale features such as crack edges and crack tails should be enhanced. The multi-scale feature fusion concept, as one of the promising methods for conducting tiny objects, was introduced in the proposed module for capturing those tiny features, which can be easily omitted in coarse segmentation architecture. As shown in Fig. 2, a raw image with four flawed segmentation masks at different scales would be extracted by the module to recover the boundary details at the finest level. To facilitate model training and eliminate the non-convergence problem due to the inconsistency of object scales, those masks would be bilinearly upsampled and then concatenated with the RGB image at the same resolution. Then, through the first two blocks of the ResNet-50 backbone and a pyramid pooling operation with the sizes of [1–3,6], these inputs will be converted into a stride 8 feature map, which contains sufficient global contextual information. At this point, the input image undergoes a series of transformative steps that include encoding and pyramid pooling, resulting in the compression of high-dimensional features abundant in semantic content. To restore the image to its original input space resolution effectively while achieving precise foreground-background separation, a gradual upsampling of features extracted by the encoder is imperative. At each stage of upsampling, feature fusion is facilitated through the use of skip connections, amalgamating encoder feature maps with upsampled feature maps of corresponding dimensions, allowing the network to integrate low-level and high-level feature information to enhance semantic representation. The connected feature maps subsequently undergo decoding processing through a series of 3×3 convolutional layers to further extract and refine features. These decoding steps are iterated four times, progressively elevating the quality and resolution of the feature maps. The final decoding layer encompasses a 3×3 convolutional layer coupled with a two-fold upsampling operation aimed at generating a crack prediction mask matching the dimensions of the original input image. Finally, besides the stride 1 prediction being obtained, the other two intermediate stride segmentation results (stride 8 and stride 4) were also collected simultaneously, which can be further employed in the cascade refinement procedure.

2.2. Global and local cascade refinement

Inspired by the human ability to perceive object details from coarse to fine levels, two cascading refinement operations were developed. These operations mimic human perceptual strategies by first conducting

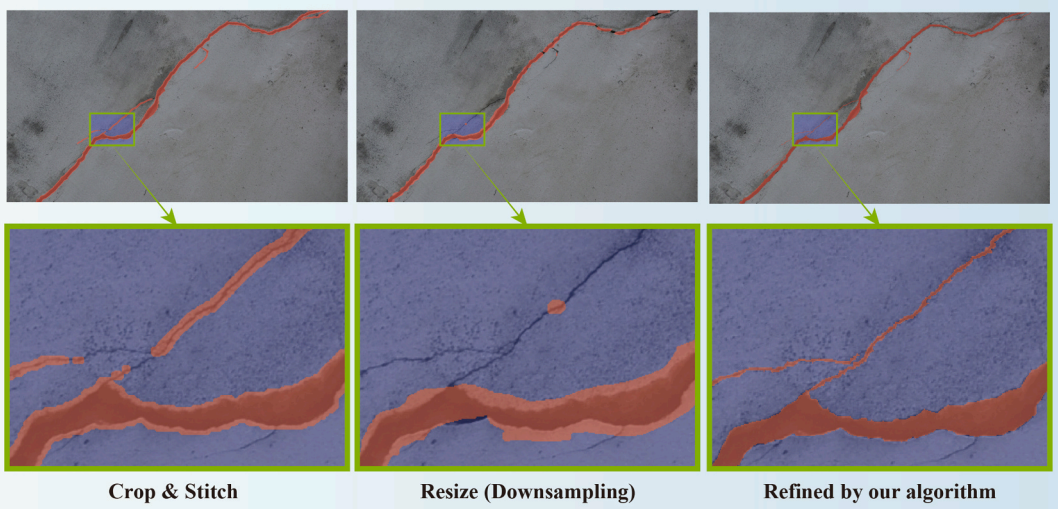


Fig. 1. Segmentation results of HR crack image (3492 × 2328). Left and Middle: Produced by Deeplab V3+, Right: Refined by CBRF.

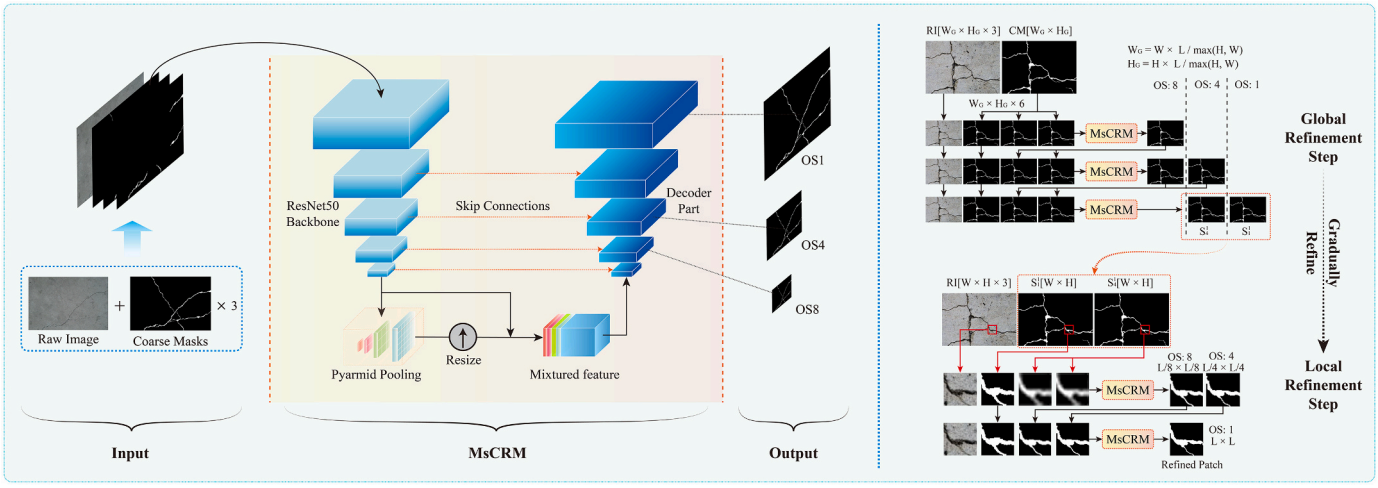


Fig. 2. Proposed crack-boundary refinement framework.

global refinement followed by local refinement, applying this logic to multi-scale preliminary results.

2.2.1. Global cascade operation

A globalized three-level cascade operation was first designed and applied to the whole image to mimic the initial stages of human-eye refined recognition, and the whole process is illustrated in Fig. 3. Since direct use of HR images for training and testing often requires a higher-configured GPU, to reduce the hardware requirements of the proposed method, the original HR input was proportionally down-sampled for the long-axis to the length of L (L is set to 900 in this paper). As can be seen in Fig. 3, the coarse mask with the same downsampling size was duplicated three times and then input into the MsCRM with the raw image.

After the first refinement operation, the coarse stride 8 output from the bilinear upsampling of two of the second-level input channels will be used instead. Then, two of the third-level input channels will be replaced with the bilinearly upsampled coarse stride 8 and stride 4 outputs from the last cascade operation. Similar input substitutions are used at each subsequent level, where the final input is made up of some pertinent outputs from earlier levels as well as the starting segmentation.

This specially designed workflow enables the Multi-Scale Cascade Refinement Model (MsCRM) to iteratively refine prediction errors while preserving information from the initial coarse mask. Crucially, the multi-scale cascade operation empowers the network to leverage the strengths of features at varying depths; specifically, it adeptly merges deep semantic insights with surface-level textural details to accurately delineate crack boundaries.

2.2.2. Local cascade operation

After completing the global refinement, a local refinement step is proposed to refine the local details of the HR image sequentially. As shown in Fig. 4, the local patch cropped from the HR image and its relevant mask patches were successively fed into the two-level cascade operation. In fact, this sequential processing of cropped patches is a good compromise to the problem that general GPUs find it difficult to handle HR images directly, both in the training and inference stages.

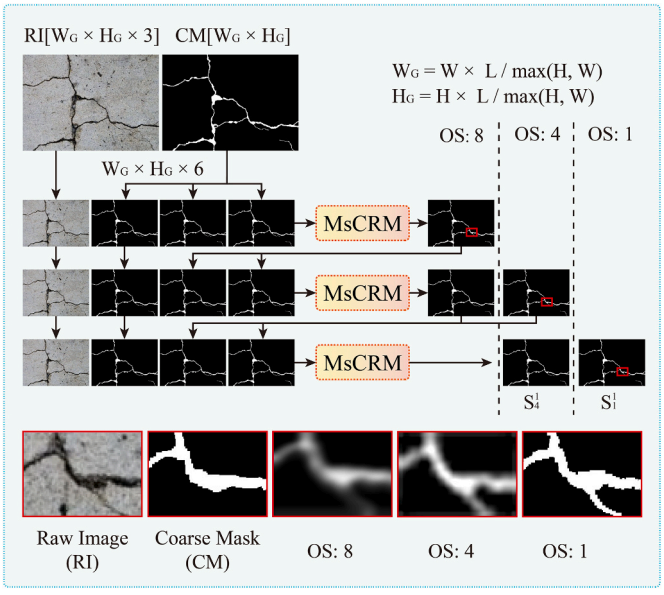


Fig. 3. Global cascade operation: the MsCRM was adopted to perform a 3-level cascade operation with output strides (OS) of 8, 4, and 1. The cascade is jointly optimized, capturing crack trunks at large output strides and crack branches at small output strides (i.e., with a meticulous boundary).

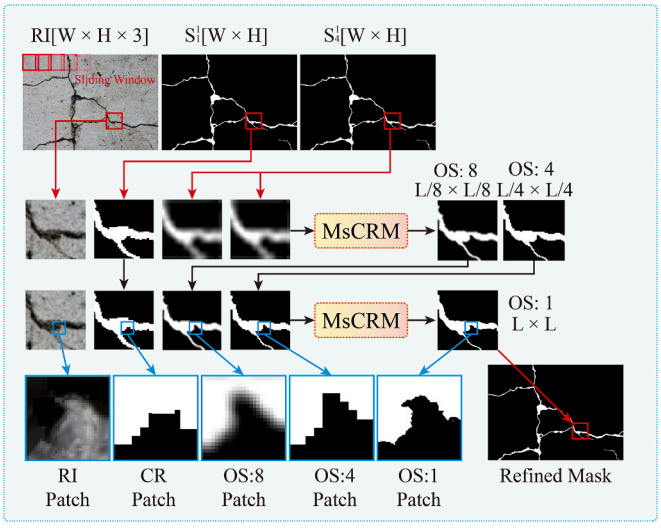


Fig. 4. Local cascade operation: the MsCRM was adopted to perform a 2-level cascade operation based on the cropped output from the global cascade operation. Blue rectangles demonstrate our visual improvement. The final output will be a fusion of all the image crop outputs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Meanwhile, since each cropped patch is determined according to the global refinement results, the global context information is largely preserved.

Specifically, two output masks (stride 1 and stride 4) from the global refinement process were bilinearly upsampled to the original image size with a resolution of W × H. Then a L × L sliding window with a stride of 4/5 × L would be employed on the image set, which consists of the original image and its masks, to conduct patch cropping. It should be noted that to avoid boundary artifacts, 24 pixels from each side of the cropped patches would be cut away. Then, the two-level local cascade operation was sequentially performed on those cropped patch sets, as can be seen in Fig. 4, which is similar to the global refinement stage. Finally, each patch prediction result will be well-organized and recombined into a HR prediction mask. Since there were some overlapping areas between the adjacent patches, prediction results in those areas may be inconsistent due to the different context information of each patch. This differentiation is eliminated by applying the mean prediction value for those overlapping areas. Through the above local steps, the boundary details of the crack are efficiently fixed in a coarse-to-fine manner by taking advantage of the corresponding scale advantage in the multi-level set.

Pseudocodes for the implementation process of the global cascade operation and the local cascade operation have been added to the appendix.

Specifically, the design of Global and Local cascade operations follows the progressive refinement principle, which means the cascade operations conduct refinement in a coarse-to-fine manner to gradually restore the crack details from the global and local perspectives, respectively. Firstly, the Global cascade operation enhances the understanding of the entire image through a three-layer structure, providing a more accurate global context for the subsequent local refinement. This three-layer-based operation logic was designed to fully utilize the three scales of features outputted by the MsCRN, thereby improving the global coarse feature through a layer-by-layer iterative approach. Then, the Local cascade operation focuses on the refinement of local areas, employing a two-layer design to enhance the capture and correction of tiny details. The reason for reducing one layer compared to the global cascade operation is that the Local cascade operation proved to be sufficient for effective refinement of local features while avoiding overfitting or excessive computational costs.

Additionally, it should be noted that in both cascade operations, multi-scale feature maps with strides of 1, 4, and 8 are ultimately selected for refinement. This selection principle is to ensure that the advantages of feature maps at different scales can be fully used to complement each other, thereby enhancing the representation of tiny crack details. Specifically, feature maps with a stride of 1 can provide ample pixel-level details, while those with strides of 4 and 8 contribute to a broader understanding of the contextual semantics and structural information of the crack. Such a design aids the model in maintaining global consistency while focusing on local details.

In summary, a cascade operation-based design paradigm for building refinement architecture is introduced in Section 2.2. This paradigm provides valuable guidance for researchers in developing specialized frameworks for HR crack image segmentation. It should be noted that the number of stages chosen for two cascade operations and the selected three-scale features in each cascade stage are the optimal configuration obtained on the authors' existing experimental equipment. Moreover, the authors maintain an open stance on the potential for more optimal architectural parameters for cascade operations. Theoretically, incorporating additional multi-scale inputs and cascade stages could potentially improve the model's predictive performance. Research on these parameters will be further explored in the future with more advanced experimental equipment.

2.3. Joint cascade loss with active boundary loss

In segmentation tasks, the utility of features decoded at various scales differs significantly. Shallow layer features, enhanced through multiple decoding and upsampling processes, offer higher resolution and are thus better suited to capturing fine edge details. In contrast, features from deeper layers provide rich semantic context due to their larger receptive fields, aiding in the identification of crack structures throughout the image. Tailoring distinct loss functions for features at each scale enables the leveraging of their unique advantages, improving segmentation accuracy from both a broad and detailed perspective.

Drawing on these principles of loss function design, a mixture of distinct loss functions is employed to supervise output features at each level to ensure that the advantages of relevant level features can be brought into full play. Based on the loss function used in the literature [64], some improvements have been carried out in this study to make the supervised procedure more suitable for the tiny crack scene. Specifically, the average of cross-entropy loss + Dice loss (Aims to fully exploit the strengths of cross-entropy loss in capturing profound semantic information while mitigating the adverse effects of class imbalance through the use of Dice loss), L1 + L2 loss (With the objective of maximizing their advantages in finely depicting low-level topological features, thereby achieving meticulous refinement of edge contour information in shallow feature maps), and the average of cross-entropy and L1 + L2 loss (Ensure comprehensive extraction of residual semantic information in the intermediate layer, as well as to provide an initial representation of edge contour details in the feature maps) were used for the coarser stride 8 output, finer stride 1 output, and intermediate stride 4 output, respectively. Moreover, ABL, which can be used for progressively encouraging the alignment between predicted boundaries and ground-truth boundaries during end-to-end training, was proposed and introduced on the stride 1 output to enhance the boundary refinement. The ABL can be written as:

$$L_{ABL} = \frac{1}{N_b}\sum_{i}^{N_b}\Lambda(M_i)CE(D_i^p, D_i^q) \tag{1}$$

The weight function Λ is computed as $\Lambda(x) = \frac{min(x,\theta)}{\theta}$, where N_b is the number of pixels on the predicted boundaries (PDBs), and θ is a hyperparameter set to 20. The closest distance to the ground-truth boundaries (GTBs) as pixel i is used as a weight to penalize its deviation from the GTBs. If M_i is 0, indicating that the pixel is already on the GTBs, this pixel will be discarded in the ABL.

$$Loss = \frac{1}{2}\left(L_{CE}^8 + L_{Dice}^8\right) + \frac{1}{2}\left(L_{L1+L2}^4 + L_{CE}^4\right) + L_{L1+L2}^1 + L_{ABL}^1 \tag{2}$$

where L_{CE}^8, L_{L1+L2}^8, and L_{ABL}^1 denote cross-entropy loss, L1 + L2 loss, and gradient loss for output stride s respectively.

3. Experiments

In this section, some preliminary preparations for the experiments were first introduced. After that, the influence of image resolution on the performance of the proposed method is mainly studied. Finally, the efficacy of the CBRF is further substantiated through ablation studies.

3.1. Datasets

For the training and preliminary evaluation of the proposed method, an open-source dataset with LR images of cracks, Deepcrack, was selected. The Deepcrack [70] consists of 537 images of cracks with a resolution of 544 by 384 pixels, including transverse, longitudinal, oblique, and alligator-shaped cracks. To better capture the actual scene, those photographs were taken in a variety of lighting conditions (including shadow, occlusion, low contrast, and noise). It should be noted that although widely used in training the crack segmentation

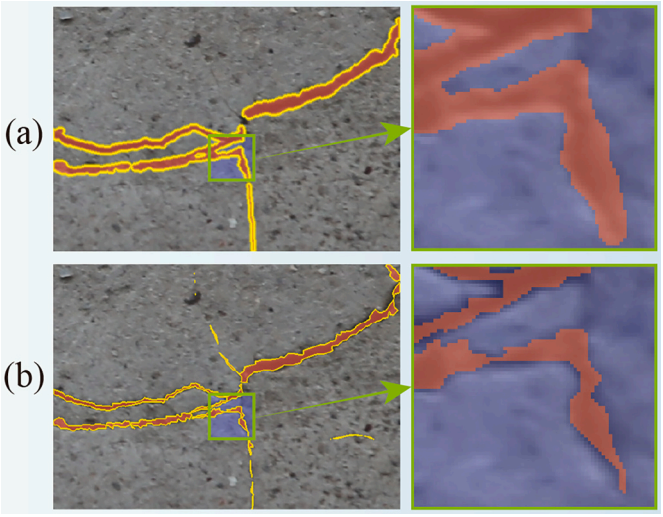


Fig. 5. Visualization of boundary details: (a) Original label of crack image from Deepcrack; (b) the same image with meticulous boundaries.

models, the Deepcrack dataset does not have pixel-perfect segmentations, and the labelled areas near the crack boundary sometimes look blurry, which may confuse the proposed refinement module in the identification of crack boundaries radically. To avoid this negative effect as much as possible, the authors finely re-labelled all the crack images contained in Deepcrack. Fig. 5 shows some relabeled examples.

To further assess the proposed method's efficacy in processing high-resolution crack images, the authors introduce the HRCD-282 dataset. This dataset comprises 282 images, with resolutions ranging from 2 K to 6 K, including 100 images each at 2 K and 4 K resolutions, and 82 images at 6 K resolution. All these original images were captured manually with the Nikon D810 in Changsha with a resolution of 5780 × 2890 pixels and then cropped to the corresponding sizes. The image acquisition details are illustrated in Fig. 6. During the shooting process, the authors held the camera about 50 cm away from the wall to ensure that tiny cracks could be unambiguously captured. To minimize image blur from hand movement, which compromises precise crack boundary labeling, the shutter speed was limited to a maximum of 1/100 s and the flash activated to maintain image brightness. Cracks that are longitudinal, transverse, oblique, mesh, and other types can be found in the HRCD-282. The collected crack photos comprise a variety of noises in addition to numerous sorts of cracks, including noises from wall joints, calcimine peeling off, electric lines, and varied lighting circumstances. Meanwhile, every image in the dataset was elaborately labelled by the authors while keeping the same guidelines as Deepcrack without the ambiguous definition of the crack boundary. Specifically, to ensure the accuracy of the data annotation, the authors controlled the quality of the annotations in the following three aspects:

1. High-Resolution Image Data Source: To ensure that annotators can perform confident and comprehensive labelling, it is essential that the crack boundaries on the images are very clear. Therefore, after the collection of crack images, two different professionals were arranged to perform two rounds of selection, removing blurry crack images caused by inaccurate focus. Additionally, it is worth noting that during the image collection process, the camera's photo storage mode was adjusted to the maximum size mode (6 K resolution). By using these HR images, it ensures that the details of small targets are presented clearly while also providing more pixel-level information due to the higher resolution, which helps in more accurate labelling of small targets.

2. Use of Professional Annotation Tool: The professional open-source annotation tool LabelMe is used to allow annotators to easily draw pixel-level labels. This annotation tool provides a convenient interface and tools to enhance the accuracy of labelling.

3. Training Multiple Annotators and Mutual Verification: Before starting the labelling process, annotators are trained, especially regarding the accuracy and consistency of annotating small targets. Trained individuals are guided to perform correct annotations based on the example images and label references provided in Fig. 6. To avoid labelling errors due to the subjective judgments of a single person, the labels after each annotation are reviewed by another trained professional annotator for timely correction of any errors.

Some examples with fine annotation in HRCD-282 can be seen in Fig. 6. The HRCD-282 dataset is available online (https://hnueducn-my.sharepoint.com/:f:/g/personal/chuhonghu_hnu_edu_cn/EgdEzOnqBpBCudDdZFgBCIMBPJNHdu4zjGg9X2Ti32R4Ng?e=A4lkps). The HRCD-282 dataset is the first real HR dataset for the crack segmentation task and can be used as the benchmark dataset for future segmentation models aimed at segmenting HR cracks.

3.2. Evaluation method

To quantitatively assess the experimental outcomes, three prevalent metrics were employed: mean Intersection Over Union (mIOU), mean Boundary Accuracy (mBA), and the Dice similarity coefficient (Dice).

The details of the mIOU and Dice can be found in the previous study [47].

The proposed method's enhancement of boundary accuracy is underscored by adopting the mean boundary accuracy metric (mBA), as introduced in CascadePSP [64], detailed in Eq. (3):

$$mBA = \frac{1}{m}\sum_{j=0}^{m}\left(\frac{1}{n}\sum_{i=0}^{n}\left(\frac{1}{\sum bd^{ij}}\sum\left\{\left[bd^{i}\cap gt\right]^{j}\otimes\left[bd^{i}\cap pd\right]^{j}+\left[1-bd^{i}\cap gt\right]^{j}\otimes\left[1-bd^{i}\cap pd\right]^{j}\right\}\right)\right) \tag{3}$$

where bd represents the crack edge matrix obtained by the morphological gradient; i denotes the sampling number of the morphological operations; j denotes the number of the crack image; gt and pd represent the ground truth matrix and the predicted mask matrix; $\otimes$ stands for the inner product of the element. The radii was sampled by 7 at $[13, 17]$ regular intervals to provide a robust estimation for cracks of various sizes.

3.3. Implementation details

Hardware equipment: The performance of the suggested technique was carried out on a desktop workstation with a GeForce RTX 3090 (24GB memory), and the system is Ubuntu 18.04 with CUDA 11.2, cudnn 11.2. To facilitate apples-to-apples comparisons, all the coarse segmentation frameworks used in this study were rebuilt with PyTorch. For the proposed method, ResNet-50 was selected as the base network.

Hyperparameters: To further improve the model's robustness, online data augmentation was applied. The batch size was set to 24, and the learning rate was established at 0.001, which was compounded by 0.9 at the end of each complete iteration. The chosen optimizer was SGD, momentum was set at 0.9, and the dropout rate was established at 0.5. The number of training epochs was also a hyperparameter, set to 1200. The total training time is around 16 h with the GeForce RTX 3090.

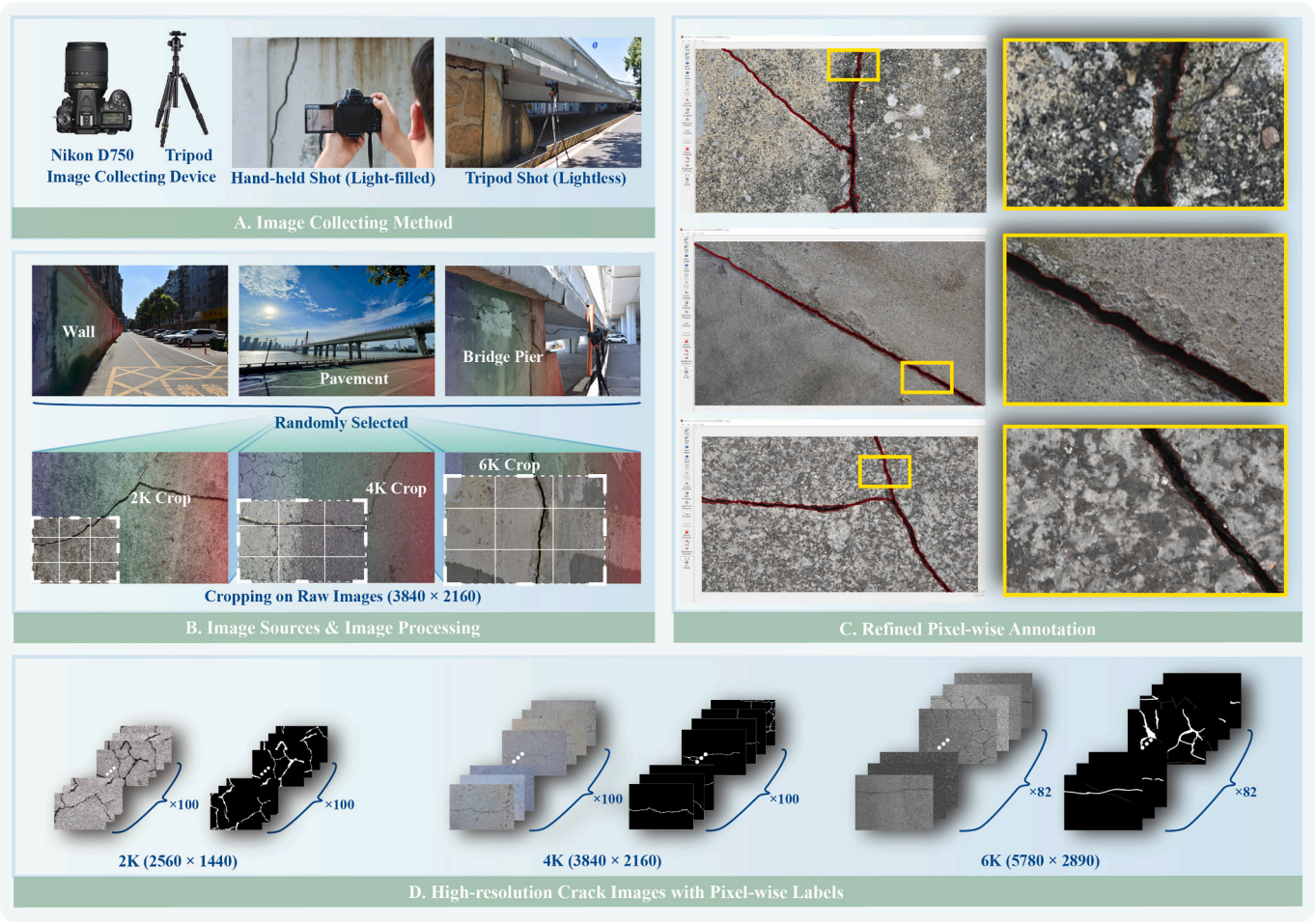


Fig. 6. Establishment of HR image dataset for crack segmentation.

It should be noted that the local cascade operation is only performed in the regions that contain cracks on mask output from the global cascade operation.

The determination of the hyperparameters presented was the culmination of an exhaustive experimental analysis, meticulously conducted to optimize the model's efficacy in detecting cracks. This process was meticulously orchestrated, prioritizing the hyperparameters based on their anticipated impact on model performance. The experimental sequence commenced with evaluations of the learning rate, optimizer, and batch size, given their paramount importance in the initial optimization phase. Within the optimal framework established by the learning rate, optimizer, and batch size previously determined, the study further evaluated the impact of momentum, dropout rate, and learning rate decay strategy on model performance. Fig. 7 visualizes the performance of models trained with all 96 different hyperparameter configurations on the HR image testing set, with the optimal parameter configuration highlighted in red.

It is crucial to underscore that during the preliminary testing phase of the three critical hyperparameters (learning rate, optimizer, batch size), all other variables, namely momentum, dropout rate, and learning rate decay strategy, were set to their most commonly used default values. This approach ensured a focused examination of the primary hyperparameters' impact, setting a robust foundation for further refinements.

Training details: The inputs used in the training procedure are concatenations of 224 × 224 patches that were cropped from the original images and their accompanying relabeled ground truth masks that were randomly disturbed to have an IOU in the range of 0.8 to 1.0 with the unperturbed ones. As in the Global step, the inputs are processed in a 3-level cascade while the loss is calculated at each stage. Although L, the crop size employed in testing, is less than the crop size, our model design helps close this gap. While the pyramid pooling module provides crucial visual context, the fully convolutional feature extractor ensures translational invariance, enabling our model to be stretched to a higher resolution without suffering appreciable performance loss. Due to the significant cost of obtaining HR training data for segmentation, we use smaller crops to speed up our training process and simplify data preparation.

3.4. Ablation study

To ensure fair comparisons and highlight the superiority of the proposed components, ablation studies were conducted in this section.

3.4.1. Ablation study for global and local refinement

The primary objective of introducing global and local cascade operations in this study is to generate high-resolution, precise masks from low-resolution, coarse segmentation of crack images. But for robust estimation, the authors conducted ablation studies on both low- and high-resolution datasets (i.e., DeepCrack and HRCD-282). The testing results under the three evaluation indicators that were introduced in Section 3.2 are shown in Table 2 and Table 3, respectively, which can be used to demonstrate the effectiveness of both global and local cascade operations. Specifically, the refinement effect was most obvious when both two cascade operations came into use, achieving improvements on mIoU, mBA, and DICE of over 9%, 18%, and 3% on the HRCD-282. When only one cascade operation was added, the effect of the global

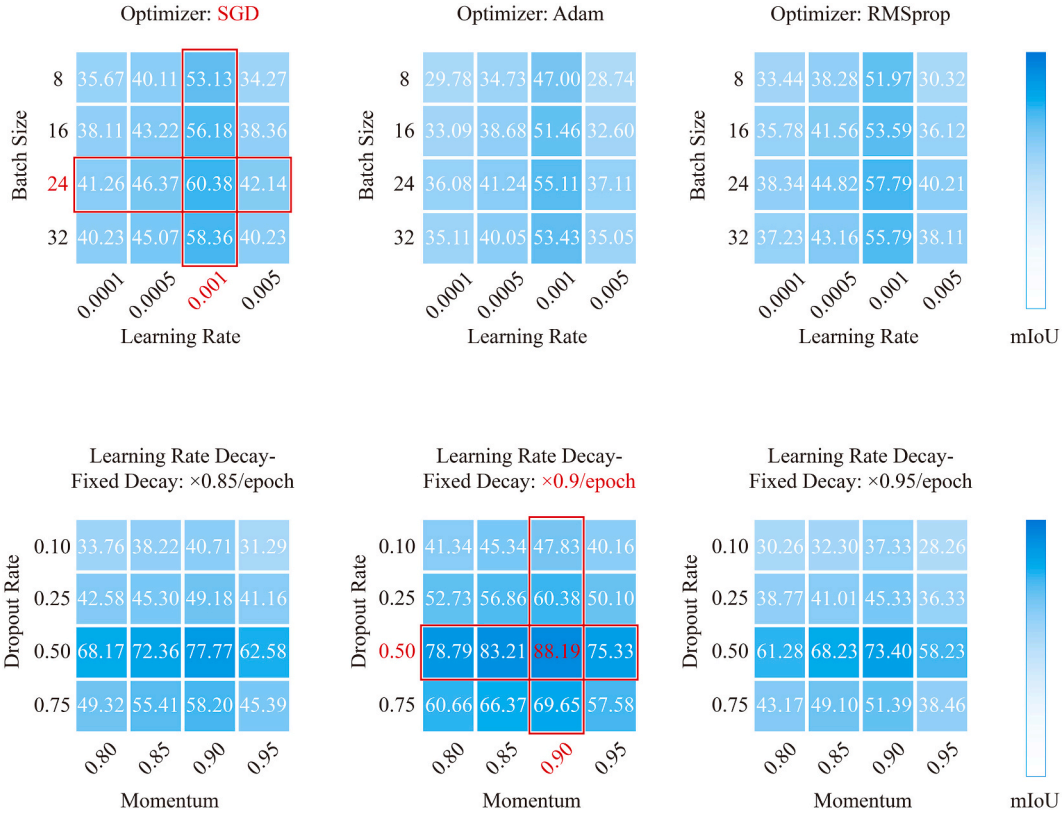


Fig. 7. Effect of different fine-tuning hyperparameters. (a) (b) (c): Testing with various optimizers, batch sizes, and learning rates; (d) (e) (f): Evaluation of different learning rate decays, dropout rates, and momentum.

Table 2
Comparison of global and local refinement based on the coarse segmentation output from the UNet (Testing on Deepcrack).

Coarse Segmentation Architecture	Refinement Operation	mIOU(%)	mBA(%)	DICE(%)
UNet	None	74.22	73.48	81.11
	Global step	$77.61_{\uparrow 3.39}$	$79.93_{\uparrow 6.45}$	$83.46_{\uparrow 2.35}$
	Local step	$74.50_{\uparrow 0.28}$	$74.66_{\uparrow 1.18}$	$82.00_{\uparrow 0.89}$
	Both step	$77.83_{\uparrow 3.61}$	$80.23_{\uparrow 6.75}$	$83.94_{\uparrow 2.83}$

Table 3
Comparison of global and local refinement based on the coarse segmentation output from the UNet (Testing on HRCD-282).

Coarse Segmentation Architecture	Refinement Operation	mIOU(%)	mBA(%)	DICE(%)
UNet	None	76.21	67.70	86.01
	Global step	$81.83_{\uparrow 5.62}$	$81.04_{\uparrow 13.33}$	$88.20_{\uparrow 2.19}$
	Local step	$83.76_{\uparrow 7.55}$	$84.15_{\uparrow 16.45}$	$88.96_{\uparrow 2.95}$
	Both step	$85.22_{\uparrow 9.01}$	$85.91_{\uparrow 18.21}$	$89.43_{\uparrow 3.42}$

step was better than that of the local step. This is because the global step is mainly designed to repair the overall object, which contributes more to the improvement of accuracy, while the local step loses some contextual information due to image cropping, which cannot be used to conduct the overall large-scale refinement operation. To further illustrate the function of the local step, a typical example of the refinement process is visualized in Fig. 4. It can be seen from Fig. 4 that the local step does have an obvious improvement on the coarse boundary, but because the boundary area accounts for a small portion of the whole object, the effectiveness cannot be well highlighted at the data level.

It is also worth noting that from Table 2, it can be seen that the proposed method is also suitable for lower-resolution images, but the effect of the local step almost disappears compared with the result obtained from the HR images. This is because the patch size in the local step is larger than the LR image itself, which makes the function of the local step similar to the global step. Therefore, the effectiveness of the local step is largely related to the size of the cropped patch and the size of the original input. In order to fully exploit the effectiveness of the local step, the segmentations produced by scaling the HRCD-282 validation set's refinement module were tested both with and without the local step. Fig. 8 shows the importance of the local step for different resolution inputs (max (H, W) varying from 300 to 3300). It can be observed that for LR crack images, only global steps are sufficient, and local steps are only effective when max (H, W) is >900.

3.4.2. Ablation study for the joint cascade loss

As mentioned in Section 2.3, different loss functions were used to supervise the output of different scales to amplify the advantages of each scale, thereby improving the accuracy of fine segmentation. Based on the architecture implemented with both steps in the previous section, ablation studies for different combinations of losses were conducted. The results were obtained in Table 4, from which it can be seen the necessity of supervising the output of each stride, and the effectiveness reflected on mBA from the embedment of ABL saw an increase of at least 8.05%. This observation can be partially explained by the fact that supervising the output of each layer allows the network to more smoothly fill the resolution gap between coarse and fine segmentation as the learning rate decays. Meanwhile, the edge of the crack can be more clearly represented with the blessing of ABL.

3.5. Robustness test of the CBRF based on some typical crack segmentation algorithms

The ablation studies previously conducted confirmed the efficacy of

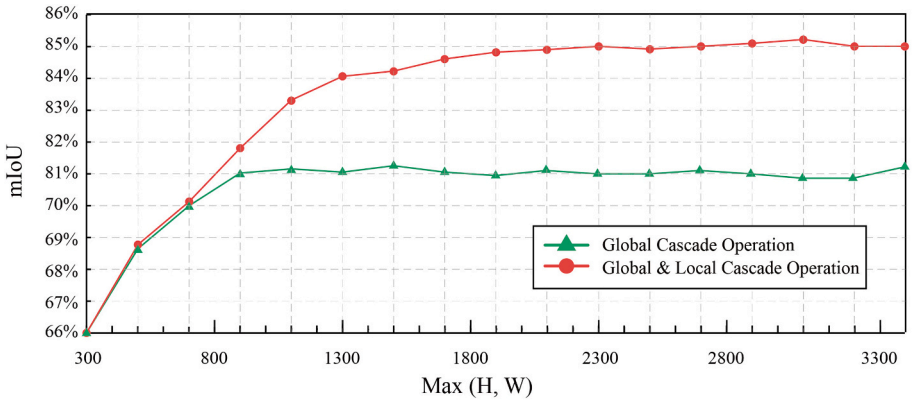


Fig. 8. Evaluation of cascade operations across different input resolutions.

Table 4
Comparison of different proportions of loss (testing on HRCD-282).

Coarse Segmentation Architecture	Loss Function	mIOU(%)	mBA(%)	DICE(%)
UNet	L^1_{L1+L2}	85.22	85.91	89.43
UNet	$\frac{1}{2}(L^4_{L1+L2}+L^4_{CE})$	$83.94_{\downarrow 1.28}$	$82.80_{\downarrow 3.11}$	$88.21_{\downarrow 1.22}$
UNet	$\frac{1}{2}(L^8_{CE}+L^8_{Dice})$	$82.94_{\downarrow 2.28}$	$81.63_{\downarrow 4.28}$	$87.55_{\downarrow 1.88}$
UNet	$L^1_{L1+L2}+\frac{1}{2}(L^4_{L1+L2}+L^4_{CE})$	$86.53_{\uparrow 1.31}$	$89.67_{\uparrow 3.76}$	$90.27_{\uparrow 0.84}$
UNet	$\frac{1}{2}(L^4_{L1+L2}+L^4_{CE})+\frac{1}{2}(L^8_{CE}+L^8_{Dice})$	$84.19_{\downarrow 1.03}$	$85.10_{\downarrow 0.81}$	$89.40_{\downarrow 0.03}$
UNet	$\frac{1}{2}(L^8_{CE}+L^8_{Dice})+\frac{1}{2}(L^4_{L1+L2}+L^4_{CE})+L^1_{L1+L2}$	$87.73_{\uparrow 2.51}$	$86.88_{\uparrow 6.97}$	$91.29_{\uparrow 1.86}$
UNet	$\frac{1}{2}(L^8_{CE}+L^8_{Dice})+\frac{1}{2}(L^4_{L1+L2}+L^4_{CE})+L^1_{L1+L2}+L^1_{ABL}$	$88.19_{\uparrow 2.97}$	$93.96_{\uparrow 8.05}$	$91.61_{\uparrow 2.18}$

Table 5
Performance of the CBRF based on the coarse masks output from five typical segmentation architectures (testing on HRCD-282).

Basic Segmentation Architecture	CBRF Operation	mIOU(%)	mBA(%)	DICE(%)
TCN	w/o	78.15	71.37	87.23
	w/	$89.61_{\uparrow 11.46}$	$94.03_{\uparrow 22.66}$	$91.98_{\uparrow 4.75}$
UNet	w/o	76.21	67.70	86.01
	w/	$88.19_{\uparrow 11.98}$	$93.96_{\uparrow 26.26}$	$91.61_{\uparrow 5.60}$
DeepLabV3+	w/o	77.03	69.37	86.56
	w/	$88.25_{\uparrow 11.22}$	$93.98_{\uparrow 24.61}$	$91.84_{\uparrow 5.28}$
FCN	w/o	71.89	59.90	81.88
	w/	$86.83_{\uparrow 14.94}$	$85.79_{\uparrow 25.89}$	$86.70_{\uparrow 4.82}$
PSPNet	w/o	76.38	68.05	86.36
	w/	$88.17_{\uparrow 11.79}$	$93.99_{\uparrow 25.61}$	$91.75_{\uparrow 5.39}$

the crack cascade refinement operation and its associated loss function, preparing the optimized CBRF for refining coarse crack masks. To showcase the robustness of the proposed CBRF, five prominent deep learning-based segmentation models—Tiny-Crack-Net (TCN), UNet, DeepLabV3+, FCN, and PSPNet—were chosen to generate coarse masks for comparative analysis. Meanwhile, to confirm the advantages of the CBRF in dealing with HR images, a parallel comparison of a compromise method (i.e., downsampling resizing) was also carried out. The testing results are listed in Table 5, and some visualization results are shown in Fig. 9.

As shown in Table 5, the average improvements of mIoU, mBA, and Dice for coarse segmentations output by the five typical methods are 12.27%, 25.01%, and 5.17%, respectively, which demonstrates the effectiveness and robustness of the proposed method. Fig. 9 shows that the results refined by the CBRF match the ground truth better, especially for the boundary areas. There were fewer false and missed detections compared with the original predictions. In addition, the CBRF can also help recover tiny cracks that were not detected in the coarse segmentation results. This is because the proposed CBRF was designed to consolidate and deepen the identification performance of crack boundaries at the physical level of the image size. Specifically, the essence of the model proposed in this study is not limited to pixel-wise prediction of black-and-white masks for RGB images. During prediction, coarse segmentation is input into the model alongside the original crack RGB images, serving as a form of prior information. The model utilizes this prior information from coarse segmentation and refines the segmentation based on the RGB images. In other words, the method proposed in this study is not traditional semantic segmentation. It is more like a refinement framework where the network learns the mapping between coarse-grained segmentation and fine-grained segmentation. Similar to super-resolution reconstruction networks that learn the mapping between blurry and clear objects, this falls under the category of low-dimensional physical learning methods. Compared to directly segmenting black-and-white crack masks from RGB images with inherent variations, the approach of using coarse segmentation as prior information to assist in identification reduces the challenges of the task. This allows computational resources to be more effectively utilized to enhance the model's robustness. The model doesn't rely on a fixed coarse segmentation architecture and doesn't require fine-tuning the model parameters based on any specific segmentation architecture. Therefore, when facing unfamiliar environments with shadows, water stains, or different materials, and so on, the cascade-based architecture does not need to be pre-finetuned, which greatly simplifies its deployment in engineering practice.

It should also be noted that although the proposed CBRF relies on the associated features obtained from the physical level to improve generalization and sensitivity in identifying crack boundaries, the lack of finetuning may still lead to false identifications in some extreme environments where there are large differences in the data distribution between the training images and the newly detected images. Some typical examples of failure segmentations are shown in Fig. 10. From Fig. 10, it can be observed that failed cases mainly fall into two categories. The first category, denoted as "Failed Case 1," is primarily attributed to two key factors: (I) the presence of impurities within the cracks and (II) the

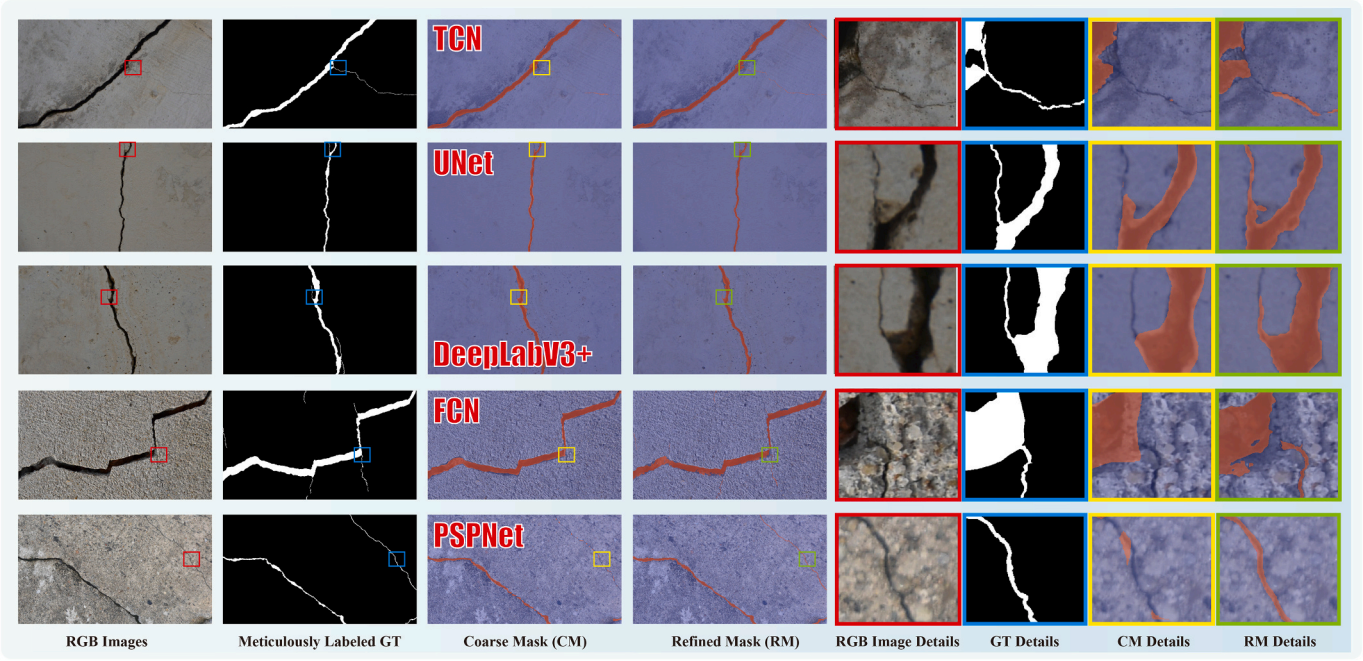


Fig. 9. Visualization of testing results of the CBRF based on five typical segmentation architectures.

existence of non-uniform light gradients along the crack edges. For cracks exhibiting significant contrast differences due to impurities within and along their edges, it's important to note that while the CBRF effectively minimizes false positives by enhancing boundary repair capabilities, its intrinsic self-recognition feature for crack characteristics may increase false negatives in these challenging areas, as highlighted in the red-marked regions of local detail in cases (I) and (II). The second category of unsuccessful cases primarily stems from regions that bear a resemblance to cracks. These regions are chiefly ascribed to two primary sources: (III) water stains and (IV) surface paint that has been subjected to prolonged exposure to sunlight and erosion by wind. Notably, the CBRF often produces false positive judgments in these areas that have a very similar appearance and data distribution pattern to cracks (the green-marked area shown in local details in cases (III) and (IV)). It is recommended that more unlabeled water stain images and paint peeling images obtained under different circumstances be added to the training dataset or a fine-tuning process be added to improve the performance of the proposed method under such circumstances.

4. Case study

To further verify the effectiveness of the proposed method in practical crack inspection tasks. A case study was carried out at the Yinpenling Bridge located in Changsha, China (see Fig. 11). The bridge has been in service for nearly 30 years, and it is still one of the pivotal projects connecting the east and west sides of Changsha. As a flexible and convenient image acquisition tool, unmanned aerial vehicles (UAVs) are expected to replace expensive and cumbersome bridge inspection vehicles to implement bridge damage detection in the future. So, in this case study, a novel industry-class UAV (DJI M300RTK) was used to capture images of the bridge. To simplify the experimental process, one of the side spans of the Yinpenling Bridge was selected for conducting this case study. The details of the on-site inspection process and the UAV system are indicated in Fig. 11. The quality of the images collected by drones directly impacts the recognition accuracy of crack models in subsequent steps. Therefore, it is essential to ensure the clarity of the collected images as much as possible. In this study, the author controls the quality of the images collected by drones through the following three aspects: Firstly, the drone must strictly follow a predefined flight path for image collection. As shown in Fig. 11, the distance between the camera and the structure is stabilized at around 1.5 m, which allows small cracks with a width of at least 0.15 mm to be adequately represented in RGB images in the form of effective pixels (at least 2 pixels). Secondly, the drone's crack detection task is scheduled for clear noon on sunny days. The direct sunlight at noon can avoid uneven shadows on the structure's surface, enabling the camera to quickly and accurately focus on small cracks. Finally, the author employs a video recording mode at 4 K resolution (30 frames per second) for crack image collection, and the drone's flight speed along both sides of the structure is controlled at 1 m per second. This allows the camera sufficient time to achieve accurate focus without missing any potential small cracks. It should be noted that the M300 RTK is equipped with a high-precision inertial measurement unit, a flight control system, a visual positioning system, and a laser rangefinder, which maintains the cruise accuracy of the drone in three-dimensional space within 2 cm. This effectively ensures the accurate execution of the collection process according to the plan, thereby guaranteeing the capture of clear crack images at a 4 K resolution. Finally, 100 4 K-resolution (4056×3040) images were captured from the H20T HD camera during a 10-min flight, containing tiny cracks with a width of at least 0.05 mm.

To further illustrate the advantages of the proposed method in processing those HR crack images in practical engineering, the authors compared the CBRF with some conventional HR image processing methods (i.e., resizing, crop & stitch, and direct processing of HR images) from the aspects of parameter quantity, operation efficiency, computational resource occupation, and accuracy. Due to the limitations of computing power based on some common GPUs, most of the segmentation models cannot be directly used for processing HR crack images. Therefore, resizing and crop & stitch were commonly used as compromise methods. Regarding the resizing method's implementation, the HR image was downsampled to a long-axis of 512, maintaining the aspect ratio, before being fed into the basic segmentation model for initial prediction. Those initial predictions were subsequently bicubic-upsampled to the original high resolution for comparison. As for the specific operation process of the crop & stitch method, the authors orderly crop the HR image into 512×512 image patches for the mask generation and then reorganize the prediction results into an HR mask according to the cropping order. Furthermore, when the GPU memory

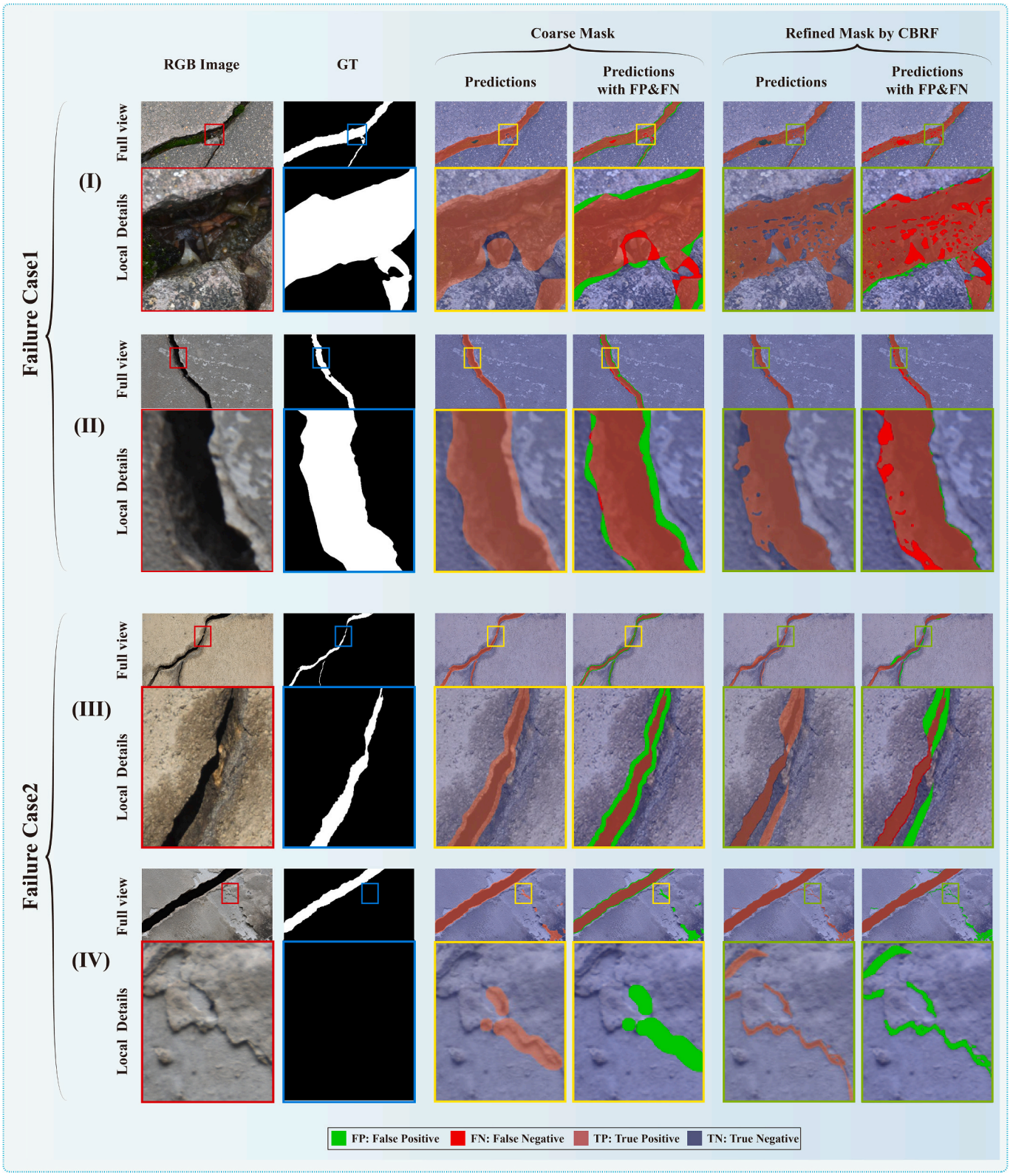


Fig. 10. Failure cases of the CBRF: Failure case1 includes (I) impurities inside of the crack and (II) colour gradient at the boundary region; Failure case2 includes (III) areas with water stains and (IV) peeling paint.

limit was released, the prediction results output by directly processing HR images were also compared. Further, this study, focusing on fine-grained segmentation of high-resolution (HR) crack images, necessitates a comparison with current state-of-the-art fine-grained segmentation methods for HR images. Notably, the PointRend architecture by Kirillov et al. [73], designed for semantic segmentation tasks, employs a fine-grained rendering strategy that substantially improves the model's accuracy in delineating image edges and complex textures. It represents the most advanced method in fine-grained segmentation, and thus, it has been used for parallel comparison with the CBRF in this study. It should

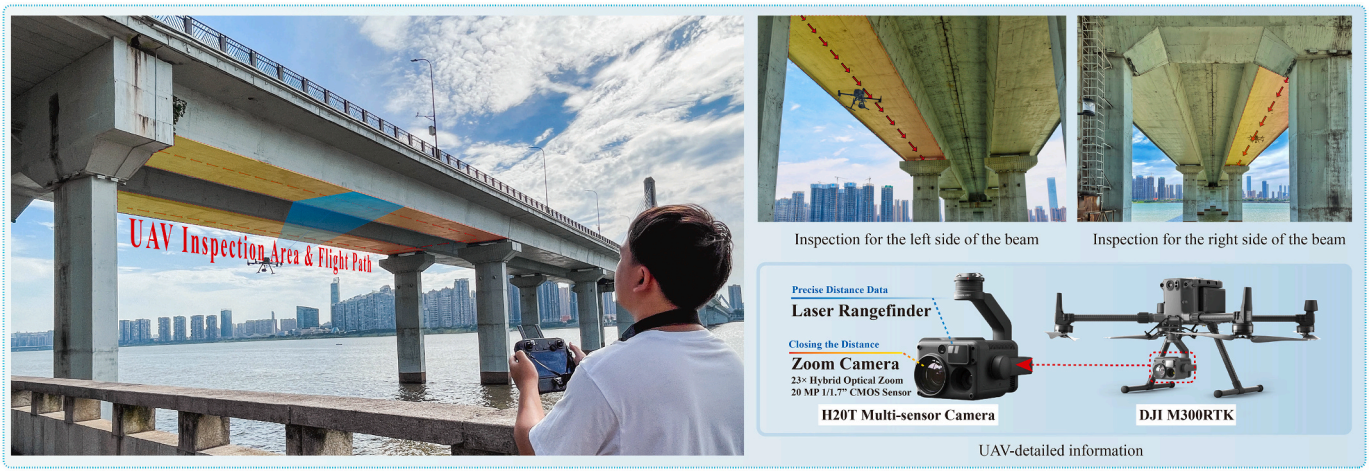


Fig. 11. UAV inspection process and equipment details.

Table 6
Performance of the CBRF and three compared methods in refine the UAV-collected HR images.

Coarse Segmentation Architecture	Operation	Total Parameters(M)	Inference Speed (FPS)	GPU Occupation (M)	mIOU(%)	mBA(%)	DICE(%)
TCN(Tiny Crack Net)	Resize	25.29	21.14	7664	75.63	68.62	82.34
	Crop & Stitch	25.29	0.33	7664	78.14	73.69	84.43
	Directly input HR IMG	25.29	0.87	33,210	79.08	75.37	85.08
	PointRend	40.76	3.32	10,813	82.76	87.18	87.21
	CBRF	34.54	12.56	8851	87.33	92.18	89.36

be noted that all the testing results and the coarse segmentation sources required for fine-grained segmentation architectures were obtained based on the best-performing basic segmentation model, "TCN," in Table 5. The "TCN" was also proposed by the authors in the previous study as one of the state-of-the-art crack segmentation networks aiming to solve the problem that tiny cracks were difficult to identify and has been proven to be effective in conducting UAV-based bridge inspections [47].

The testing results based on the HR images collected on-site are listed in Table 6. It can be seen that although the CBRF was a relatively redundant method from the aspect of parameters (34.54 M), it achieved the best accuracy of all the mIOU (87.33%), mBA (92.18%), and DICE (89.36%) while maintaining a relatively high inference efficiency (12.56 frames per second (FPS)). In contrast, PointRend focuses on improving segmentation quality by selectively refining classification at challenging regions, such as object boundaries or areas with intricate textures. While PointRend effectively enhances detail resolution at these critical points through its point-based rendering technique, it does not inherently address the integration of multi-scale contextual information as comprehensively as CBRF does. Therefore, while both CBRF and PointRend architectures offer significant contributions to the field of semantic segmentation, it is the comprehensive, multi-scale refinement strategy of CBRF that underpins its superior performance in our experiments. This architectural advantage enables CBRF to deliver more accurate and contextually coherent segmentation results, making it particularly well-suited for applications requiring high levels of precision and detail across diverse scene compositions.

To further demonstrate the advantages of the proposed CBRF over the other four methods, some of the testing results are randomly selected and illustrated in Fig. 12. As shown in Fig. 12, cracks get the finest mask at the boundary area with the proposed CBRF, but it is difficult to obtain refined crack boundaries when using resizing or crop & stitch methods. In particular, cracks at recombination seams may be discontinuous due to the loss of contextual information from cropped image edges. Meanwhile, the CBRF demonstrated the capability to reconstruct undetected cracks from coarse segmentation results, surpassing even direct processing of HR images with ultra-high-performance GPUs or utilizing cutting-edge fine-grained rendering technologies.

It was noted that accuracy and efficiency are the most important considerations in performing field inspection tasks. In structural crack inspections, some studies have utilized carrier systems with LR segmentation methods, requiring proximity to structural surfaces of no >0.5 m for effective crack detection [74,75]. This requirement significantly limits the camera's field of view (FoV), constraining the observable area and, consequently, diminishing the overall efficiency of the detection process due to the necessity for multiple passes to cover the same inspection area comprehensively.

Contrastingly, in our case study, the implementation of the CBRF marks a substantial advancement in this domain. The CBRF enables the utilization of the UAV-camera at an increased operational distance of up to 1.5 m from the beam surface while maintaining the sensitivity required to identify cracks as narrow as 0.15 mm in width. This enhancement in operational distance directly translates to a broader FoV, significantly enlarging the area captured within a single frame of the inspection. The broader coverage not only streamlines the inspection process by reducing the number of required flight paths to achieve comprehensive coverage but also substantially lowers the risk of potential collisions, thereby elevating both the safety and efficiency of UAV-based structural inspections.

Furthermore, it is essential to note that while the inference speed of the method proposed in this study in a laboratory environment reaches 12.56 FPS, this does not imply that the UAV can use it to perform very rapid crack image sampling. This is because of the possibility of bridge-induced gusts that may affect the stability of the UAV during flight, potentially leading to blurred images captured by the camera. Therefore, to reduce the risk of image blurring affecting the model's recognition accuracy and take into account the safety of the UAV during the detection process, the flight speed of the UAV should be controlled at 1 m/s when performing on-site detection tasks.

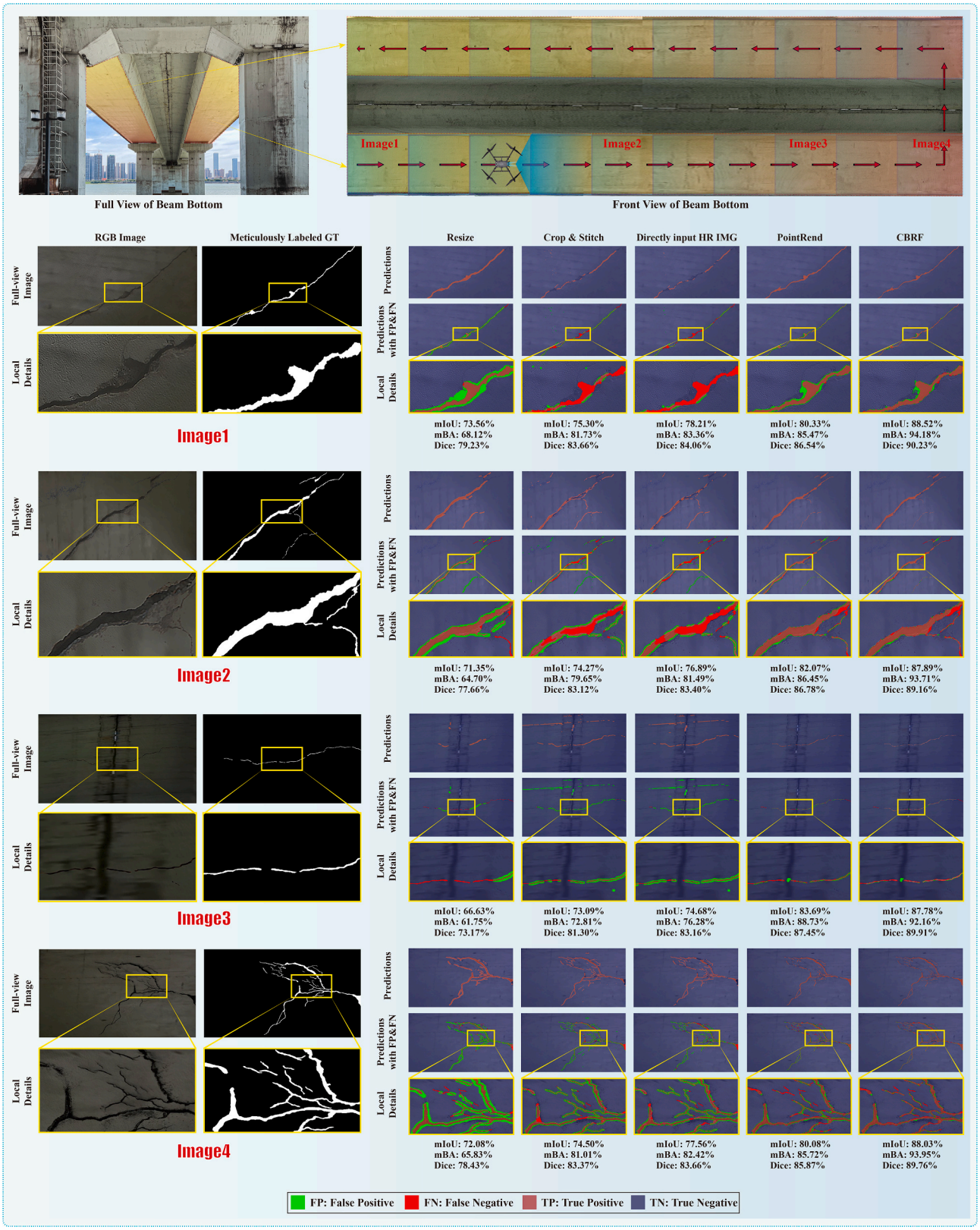


Fig. 12. Randomly selected testing samples collected on-site by the UAV and the prediction results.

5. Conclusions

This paper presented the Crack Boundary Refinement Framework (CBRF) for segmenting cracks with meticulous boundaries from high-resolution (HR) images. Three main components were included in the proposed CBRF. First, a single refinement module with triple-scale feature extraction capability was proposed to reduce the loss of miniscule crack pixels. Subsequently, a multiscale repair fusion module comprising global and local joint operations was proposed to predict the detailed contours of miniscule cracks in HR images more accurately.

Finally, an ABL was introduced in the training process to address the semantic discrepancy of feature fusion between the encoding and decoding stages. To prove that each proposed component is effective, the first HR crack image dataset (referred to herein as HRCD-282) was established to conduct ablation studies. Subsequently, a comparative experiment and a case study were conducted to further demonstrate the generalization and practicability of the well-trained CBRF. From the experimental results, the following conclusions were inferred:

1. The proposed method solved the problem of the conventional DL-based architecture, wherein cracks cannot be meticulously segmented from HR images. It achieved a mBA of 94.03% on a self-built 4 K-resolution dataset, which was at least 22.66% higher than those of five baseline references implemented using conventional architectures.
2. The CBRF was developed using a cascading operation architecture to refine crack segmentation at a physical level. This enables the enhancement of coarse segmentation results without the need for fine-tuning, allowing for the reconstruction of minute crack features not detectable in initial coarse predictions.
3. The first HR crack image dataset (dubbed HRCD-282) was established in this study and can be used as a benchmark dataset for the evaluation of future segmentation algorithms targeted towards HR crack images.
4. The application of the CBRF enables the UAV to perform crack inspection at a distance of 1.5 m from a structural surface, which implies a safer inspection. Meanwhile, the larger field-of-view resulting from the larger capturing distance will halve the inspection time.

Given the limitations of the CBRF in handling conditions such as wet surfaces, uneven lighting, and impurities resembling cracks, acquiring a comprehensive dataset covering a wide range of complex crack images with varied backgrounds is essential. This dataset will serve the purpose of advancing the training and fine-tuning of CBRF to bolster its resilience in practical engineering applications. Furthermore, directly using HR images for training and testing requires a significant amount of resources. The performance of the proposed CBRF was exploited through LR training and HR testing, which is a trade-off solution owing to the limitations of computational resources. Therefore, to bridge the gap between training and testing resolutions, manual debugging and the selection of an optimal downsampling length $\boldsymbol{L}$ are necessary for crack images of varying types and thicknesses. As this is laborious when deploying the CBRF in every case, more relevant investigations will be conducted in the future.

CRediT authorship contribution statement

Lu Deng: Supervision, Project administration, Funding acquisition. **Huaqing Yuan:** Software, Data curation. **Lizhi Long:** Software, Data curation. **Pang-jo Chun:** Writing – review & editing, Supervision. **Weiwei Chen:** Revisions of the paper and its eventual acceptance. **Honghu Chu:** Writing – review & editing, Writing – original draft, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgment

The authors would like to thank the editor, the three 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 Hunan Provincial Science and Technology Innovation Leader Project (Grant No. 2021RC4025), the National Natural Science Foundation of China (Grant No. 52278177), the Hunan Provincial Innovation Foundation for Postgraduate (Grant No. QL20210106), the Europe Commission project, HYDROFLEX (Grant No. 101122357), the Europe Commission project, INHERIT (Grant No. 101123326), and the China Scholarship Council (No. 202206130068).

Appendix A. Pseudocode of global & local cascade refinement process

Algorithm 1. Global cascade operation

Inputs:
image, coarse_mask, coarse_mask, coarse_mask
Outputs:
pred_os4, pred_os1

Start:

```
for i in range (2):
    if i == 0:
        mask_0, mask_1, mask_2 = coarse_mask, coarse_mask, coarse_mask
    elif i == 1:
        mask_0 = coarse_mask
        mask_1 = upsample(pred_os8, scale_factor = 8)
        mask_2 = mask_1.copy()
    elif i == 2:
       mask_0 = coarse_mask
        mask_1 = upsample(pred_os8, scale_factor = 8)
        mask_2 = upsample(pred_os4, scale_factor = 4)
    input = concat(image, mask_0, mask_1, mask_2, mask_3)
    pred_os8, pred_os4, pred_os1 = MsCRM (input)
```

Notes:

```
# image size: W*H*3, coarse_mask size: W*H, pred_os8 size: W/8*H/8, pred_os4 size:
    W/4*H/4, pred_os1 size: W*H
```

Algorithm 2. Local cascade operation

Inputs:
image, pred_os1, pred_os4, pred_os4
Outputs:
pred_mask

Start:

```
mask_0 = pred_os1
mask_1 = upsample (pred_os4, scale_factor = 4)
mask_2 = mask_1.copy ()
patch_pred_masks = []
for row in range (0, H, patch_h):
  for col in range (0, W, patch_w):
    patch_image = image [row:row + patch_h, col:col + patch_w]
    for i in range (2):
      if i= 0:
        patch_mask_0 = mask_0 [row:row + patch_h, col:col + patch_w]
        patch_mask_1 = mask_1 [row:row + patch_h, col:col + patch_w]
        patch_mask_2 = mask_2 [row:row + patch_h, col:col + patch_w]
      elif i= 1:
        patch_mask_0 = mask_0 [row:row + patch_h, col:col + patch_w]
        patch_mask_1 = upsample(patch_pred_os8, scale_factor = 8)
        patch_mask_2 = upsample(patch_pred_os4, scale_factor = 4)
      patch_input = concat(patch_image, patch_mask_0, patch_mask_1,
                           patch_mask_2)
      patch_pred_os8, patch_pred_os4, patch_pred_os1 =
                           MsCRM (patch_input)
    patch_pred_masks.append (patch_pred_os1)
pred_mask = merge_patch (patch_pred_masks)
```

Notes:
image size: W*H*3, pred_os4 size: W/4*H/4, pred_os1 size: W*H

References

[1] Y. Yao, S.T.E. Tung, B. Glisic, Crack detection and characterization techniques—An overview, Struct. Control. Health Monit. 21 (12) (2014) 1387–1413, https://doi.org/10.1002/stc.1655.

[2] P.-J. Chun, T. Yamane, Y. Maemura, A deep learning-based image captioning method to automatically generate comprehensive explanations of bridge damage, Comput. Aided Civil Infrastruct. Eng. 37 (11) (2022) 1387–1401, https://doi.org/10.1111/mice.12793.

[3] J. Valença, D. Dias-da-Costa, E. Júlio, Characterisation of concrete cracking during laboratorial tests using image processing, Constr. Build. Mater. 28 (1) (2012) 607–615, https://doi.org/10.1016/j.conbuildmat.2011.08.082.

[4] C.M. Yeum, S.J. Dyke, Vision-based automated crack detection for bridge inspection, Comput. Aided Civil Infrastruct. Eng. 30 (10) (2015) 759–770, https://doi.org/10.1111/mice.12141.

[5] J.S. Duncan, N. Ayache, Medical image analysis: Progress over two decades and the challenges ahead, IEEE Trans. Pattern Anal. Mach. Intell. 22 (1) (2000) 85–106, https://doi.org/10.1109/34.824822.

[6] A. Mohan, S. Poobal, Crack detection using image processing: a critical review and analysis, Alex. Eng. J. 57 (2) (2018) 787–798, https://doi.org/10.1016/j.aej.2017.01.020.

[7] Y.-A. Hsieh, Y.J. Tsai, Machine learning for crack detection: review and model performance comparison, J. Comput. Civ. Eng. 34 (5) (2020) 04020038, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000918.

[8] N.-D. Hoang, An artificial intelligence method for asphalt pavement pothole detection using least squares support vector machine and neural network with steerable filter-based feature extraction, Adv. Civ. Eng. 2018 (2018) 1–12, https://doi.org/10.1155/2018/7419058.

[9] S. Wang, S. Qiu, W. Wang, D. Xiao, K.C. Wang, Cracking classification using minimum rectangular cover–based support vector machine, J. Comput. Civ. Eng. 31 (5) (2017) 04017027, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000672.

[10] B.J. Lee, H.D. Lee, Position-invariant neural network for digital pavement crack analysis, Comput. Aided Civil Infrastruct. Eng. 19 (2) (2004) 105–118, https://doi.org/10.1111/j.1467-8667.2004.00341.x.

[11] R. Kalfarisi, Z.Y. Wu, K. Soh, Crack detection and segmentation using deep learning with 3D reality mesh model for quantitative assessment and integrated visualization, J. Comput. Civ. Eng. 34 (3) (2020) 04020010, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000890.

[12] F.-C. Chen, M.R. Jahanshahi, NB-CNN: deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion, IEEE Trans. Ind. Electron. 65 (5) (2017) 4392–4400, https://doi.org/10.1109/TIE.2017.2764844.

[13] H.S. Munawar, A.W. Hammad, A. Haddad, C.A.P. Soares, S.T. Waller, Image-based crack detection methods: a review, Infrastructures 6 (8) (2021) 115, https://doi.org/10.3390/infrastructures6080115.

[14] L. Zhang, Z. Wang, L. Wang, Z. Zhang, X. Chen, L. Meng, Machine learning-based real-time visible fatigue crack growth detection, Digit. Commun. Netw. 7 (4) (2021) 551–558, https://doi.org/10.1016/j.dcan.2021.03.003.

[15] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: a comprehensive review, Neural Comput. 29 (9) (2017) 2352–2449, https://doi.org/10.1162/neco_a_00990.

[16] C. Xiang, W. Wang, L. Deng, P. Shi, X. Kong, Crack detection algorithm for concrete structures based on super-resolution reconstruction and segmentation network, Autom. Constr. 140 (2022) 104346, https://doi.org/10.1016/j.autcon.2022.104346.

[17] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, https://doi.org/10.48550/arXiv.1409.1556 arXiv preprint arXiv:1409.1556.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 1–9, https://doi.org/10.1109/CVPR.2015.7298594.

[20] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 4700–4708, https://doi.org/10.1109/CVPR.2017.243.

[21] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 1492–1500, https://doi.org/10.1109/CVPR.2017.634.

[22] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, https://doi.org/10.48550/arXiv.1804.02767 arXiv preprint arXiv:1804.02767.

[23] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the 2018 IEEE Conference on

Computer Vision and Pattern Recognition, IEEE, 2018, pp. 6848–6856, https://doi.org/10.48550/arXiv.1707.01083.
[24] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun, Detnet: A backbone network for object detection, 2018, https://doi.org/10.48550/arXiv.1804.06215 arXiv preprint arXiv:1804.06215.
[25] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size, 2016, https://doi.org/10.48550/arXiv.1602.07360 arXiv preprint arXiv:1602.07360.
[26] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, https://doi.org/10.48550/arXiv.1704.04861 arXiv preprint arXiv:1704.04861.
[27] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 580–587, https://doi.org/10.1109/CVPR.2014.81.
[28] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916, https://doi.org/10.1109/tpami.2015.2389824.
[29] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international Conference on Computer Vision, IEEE, 2015, pp. 1440–1448, https://doi.org/10.1109/ICCV.2015.169.
[30] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, J. Inf. Process. Syst. 39 (6) (2015), https://doi.org/10.1109/TPAMI.2016.2577031.
[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017, pp. 2117–2125, https://doi.org/10.1109/CVPR.2017.106.
[32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.
[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the 2017 IEEE international Conference on Computer Vision, IEEE, 2017, pp. 2980–2988, https://doi.org/10.1109/TPAMI.2018.2858826.
[34] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114, https://doi.org/10.48550/arXiv.1905.11946.
[35] R. Ali, Y.-J. Cha, Attention-based generative adversarial network with internal damage segmentation using thermography, Autom. Constr. 141 (2022) 104412, https://doi.org/10.1016/j.autcon.2022.104412.
[36] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615.
[37] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 3431–3440, https://doi.org/10.1109/CVPR.2015.7298965.
[38] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.
[39] K. Pasupa, S. Vatathanavaro, S. Tungjitnob, Convolutional neural networks based focal loss for class imbalance problem: a case study of canine red blood cells morphology classification, J. Ambient. Intell. Humaniz. Comput. (2020) 1–17, https://doi.org/10.1007/s12652-020-01773-x.
[40] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, J. Li, Dice loss for data-imbalanced NLP tasks, 2019, https://doi.org/10.48550/arXiv.1911.02855 arXiv preprint arXiv:1911.02855.
[41] Y. Pan, G. Zhang, L. Zhang, A spatial-channel hierarchical deep learning network for pixel-level automated crack detection, Autom. Constr. 119 (2020) 103357, https://doi.org/10.1016/j.autcon.2020.103357.
[42] Z. Qu, C.-Y. Wang, S.-Y. Wang, F.-R. Ju, A method of hierarchical feature fusion and connected attention architecture for pavement crack detection, IEEE Trans. Intell. Transp. Syst. 23 (9) (2022), https://doi.org/10.1109/TITS.2022.3147669.
[43] G. Li, Q. Liu, W. Ren, W. Qiao, B. Ma, J. Wan, Automatic recognition and analysis system of asphalt pavement cracks using interleaved low-rank group convolution hybrid deep network and SegNet fusing dense condition random field, Measurement 170 (2021) 108693, https://doi.org/10.1016/j.measurement.2020.108693.
[44] Q. Zhou, Z. Qu, C. Cao, Mixed pooling and richer attention feature fusion for crack detection, Pattern Recogn. Lett. 145 (2021) 96–102, https://doi.org/10.1016/j.patrec.2021.02.005.
[45] Q. Zhou, Z. Qu, Y.-X. Li, F.-R. Ju, Tunnel crack detection with linear seam based on mixed attention and multiscale feature fusion, IEEE Trans. Instrum. Meas. 71 (2022), https://doi.org/10.1109/TIM.2022.3184351.
[46] S. Xu, M. Hao, G. Liu, Y. Meng, J. Han, Y. Shi, Concrete crack segmentation based on convolution–deconvolution feature fusion with holistically nested networks, Struct. Control. Health Monit. 29 (8) (2022) e2965, https://doi.org/10.1002/stc.2965.
[47] H. Chu, W. Wang, L. Deng, Tiny-crack-net: a multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks, Comput. Aided Civil Infrastruct. Eng. 37 (14) (2022) 1914–1931, https://doi.org/10.1111/mice.12881.
[48] Z. Qu, C. Cao, L. Liu, D.-Y. Zhou, A deeply supervised convolutional neural network for pavement crack detection with multiscale feature fusion, IEEE Trans. Neural Netw. Learn. Syst. 33 (9) (2021), https://doi.org/10.1109/TNNLS.2021.3062070.
[49] Q. Li, W. Yang, W. Liu, Y. Yu, S. He, From contexts to locality: Ultra-high resolution image segmentation via locality-aware contextual correlation, in: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, IEEE, 2021, pp. 7252–7261, https://doi.org/10.48550/arXiv.2109.02580.
[50] T. Shen, Y. Zhang, L. Qi, J. Kuen, X. Xie, J. Wu, Z. Lin, J. Jia, High quality segmentation for ultra high-resolution images, in: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2022, pp. 1310–1319, https://doi.org/10.48550/arXiv.2111.14482.
[51] J. Chen, Z. Yuan, J. Peng, L. Chen, H. Huang, J. Zhu, Y. Liu, H. Li, DASNet: dual attentive fully convolutional Siamese networks for change detection in high-resolution satellite images, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14 (2020) 1194–1206, https://doi.org/10.1109/JSTARS.2020.3037893.
[52] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2010) 303–338, https://doi.org/10.1007/s11263-009-0275-4.
[53] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755, https://doi.org/10.1007/978-3-319-10602-1_48.
[54] W. Wang, X. Xu, J. Peng, W. Hu, D. Wu, Fine-grained detection of pavement distress based on integrated data using digital twin, Appl. Sci. 13 (2023) 4549, https://doi.org/10.3390/app13074549.
[55] H. Liu, X. Miao, C. Mertz, C. Xu, H. Kong, Crackformer: transformer network for fine-grained crack detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3783–3792, https://doi.org/10.1109/ICCV48922.2021.00376.
[56] S. Lee, H. Kim, Q.X. Lieu, J. Lee, CNN-based image recognition for topology optimization, Knowl.-Based Syst. 198 (21) (2020) 105887, https://doi.org/10.1016/j.knosys.2020.105887.
[57] Y. Zhang, Y. Yuan, Y. Feng, X. Lu, Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection, IEEE Trans. Geosci. Remote Sens. 57 (8) (2019) 5535–5548, https://doi.org/10.1109/TGRS.2019.2900302.
[58] G. Chen, H. Wang, K. Chen, Z. Li, Z. Song, Y. Liu, W. Chen, A. Knoll, A survey of the four pillars for small object detection: multiscale representation, contextual information, super-resolution, and region proposal, IEEE Trans. Syst. Man Cybern. Syst. 52 (2) (2020), https://doi.org/10.1109/TSMC.2020.3005231.
[59] D. Kwon, S. Kwak, Semi-supervised semantic segmentation with error localization network, in: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2022, pp. 9957–9967. https://arxiv.org/abs/2204.02078.
[60] W. Chen, Z. Jiang, Z. Wang, K. Cui, X. Qian, Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images, in: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2019, pp. 8924–8933, https://doi.org/10.1109/CVPR.2019.00913.
[61] L. Shan, M. Li, X. Li, Y. Bai, K. Lv, B. Luo, S.-B. Chen, W. Wang, UHRSNet: A semantic segmentation network specifically for ultra-high-resolution images, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 1460–1466, https://doi.org/10.1109/ICPR48806.2021.9412819.
[62] S. Guo, L. Liu, Z. Gan, Y. Wang, W. Zhang, C. Wang, G. Jiang, W. Zhang, R. Yi, L. Ma, ISDNet: integrating shallow and deep networks for efficient ultra-high resolution segmentation, in: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2022, pp. 4361–4370, https://doi.org/10.1109/CVPR52688.2022.00432.
[63] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia, Icnet for real-time semantic segmentation on high-resolution images, in: Proceedings of the 2018 European Conference on Computer Vision (ECCV), Springer, 2018, pp. 405–420. https://arxiv.org/abs/1704.08545v2.
[64] H.K. Cheng, J. Chung, Y.-W. Tai, C.-K. Tang, Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement, in: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2020, pp. 8890–8899. https://arxiv.org/abs/2005.02551.
[65] Y. Shi, L. Cui, Z. Qi, F. Meng, Z. Chen, Automatic road crack detection using random structured forests, IEEE Trans. Intell. Transp. Syst. 17 (12) (2016) 3434–3445, https://doi.org/10.1109/TITS.2016.2552248.
[66] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, H. Ling, Feature pyramid and hierarchical boosting network for pavement crack detection, IEEE Trans. Intell. Transp. Syst. 21 (4) (2019) 1525–1535, https://doi.org/10.1109/TITS.2019.2910595.

[67] R. Amhaz, S. Chambon, J. Idier, V. Baltazart, Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection, IEEE Trans. Intell. Transp. Syst. 17 (10) (2016) 2718–2729, https://doi.org/10.1109/TITS.2015.2477675.
[68] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, S. Wang, Deepcrack: learning hierarchical convolutional features for crack detection, IEEE Trans. Image Process. 28 (3) (2018) 1498–1512, https://doi.org/10.1109/TIP.2018.2878966.
[69] Q. Zou, Y. Cao, Q. Li, Q. Mao, S. Wang, CrackTree: automatic crack detection from pavement images, Pattern Recogn. Lett. 33 (3) (2012) 227–238, https://doi.org/10.1016/j.patrec.2011.11.004.
[70] Y. Liu, J. Yao, X. Lu, R. Xie, L. Li, DeepCrack: a deep hierarchical feature learning architecture for crack segmentation, Neurocomputing 338 (2019) 139–153, https://doi.org/10.1016/j.neucom.2019.01.036.
[71] Q. Mei, M. Gül, A cost effective solution for pavement crack inspection using cameras and deep neural networks, Constr. Build. Mater. 256 (2020) 119397, https://doi.org/10.1016/j.conbuildmat.2020.119397.
[72] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, H.-M. Gross, How to get pavement distress detection ready for deep learning? A systematic approach, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2039–2047, https://doi.org/10.1109/IJCNN.2017.7966101.
[73] A. Kirillov, Y. Wu, K. He, R. Girshick, Pointrend: Image Segmentation as Rendering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9799–9808, https://doi.org/10.1109/CVPR42600.2020.00982.
[74] K. Jang, Y.-K. An, B. Kim, S. Cho, Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot, Comput. Aided Civ. Inf. Eng. 36 (2021) 14–29, https://doi.org/10.1111/mice.12550.
[75] S. Jiang, J. Zhang, Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system, Comput. Aided Civ. Inf. Eng. 35 (2020) 549–564, https://doi.org/10.1111/mice.12519.