Hindawi
Structural Control and Health Monitoring
Volume 2023, Article ID 3301106, 17 pages
https://doi.org/10.1155/2023/3301106

Research Article

Two-Stream Boundary-Aware Neural Network for Concrete Crack Segmentation and Quantification

Gaoyang Liu,[1,2] Wei Ding,[2] Jiangpeng Shu,[2,3] Alfred Strauss,[4] and Yuanfeng Duan[2]

[1]School of Civil Engineering, Shaoxing University, Huancheng West Road 508, Shaoxing 312000, Zhejiang, China
[2]College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China
[3]Innovation Center of Yangtze River Delta, Zhejiang University, Hangzhou 310058, China
[4]Institute of Structural Engineering, University of Natural Resources and Life Sciences, Vienna 1190, Austria

Correspondence should be addressed to Jiangpeng Shu; jpeshu@zju.edu.cn

Received 4 November 2022; Revised 15 March 2023; Accepted 17 June 2023; Published 30 June 2023

Academic Editor: Chia-Ming Chang

Cracks can be important performance indicators for determining damage processes in new and existing concrete structures. In recent years, deep convolutional neural networks (CNNs) have shown great potential in automatic crack detection and segmentation. However, most of the current CNNs tend to lose high-resolution details and, therefore, lead to blurry object boundaries; this results in poor performance for crack images with complex backgrounds in engineering structures. This study proposes a two-stream boundary-aware crack segmentation (BACS) network that combines semantic image segmentation with semantically informed edge detection explicitly. Firstly, a high-resolution network (HRNet) is utilized in the segmentation branch for strong high-resolution representations through repeatedly conducting multi-scale fusions across parallel convolutions. Furthermore, an edge branch is utilized for preserving fine-grained details of elongated thin cracks, which adopts a modified dynamic feature fusion (DFF) network to produce more accurate and sharper edge predictions. The proposed method is evaluated using a dataset of 1,892 images for three different scenarios. The results show that the mean intersection-over-union (mIoU) scores reach 79.26%, 68.74%, and 70.31% for pure crack, complex background, and variable-width scenarios, respectively. In addition, crack width quantification is performed to validate the accuracy in terms of engineering practice. The BACS achieves high accuracy with an average absolute error of 0.0992 mm, which corresponds to approximately two pixels in the images. In conclusion, the study provides an effective solution for the crack segmentation task, especially for the variable-width scenario, providing an accurate data foundation for the digital twin of concrete structures.

1. Introduction

Cracks and associated crack patterns can be indicators of the stress state, loss of durability and reliability, and thus of the service life of concrete structures. Therefore, certain cracks and crack patterns also provide important information about the deterioration of the concrete structure. Existing cracks can significantly accelerate corrosion and expansion of the reinforcement by the corrosion products and finally spalling of the concrete surface, which leads in the end to a reduction of the load-bearing capacity and the service life of the reinforced concrete structure. Frequent structural monitoring with reliable reporting of the condition of the inspected infrastructures is a necessary procedure to maintain their long-term service capabilities [1–4]. Manual inspections, which have been used for decades, are the most widely used methods for crack detection. However, owing to the subjective judgment of inspectors and dangerous working conditions, they are always criticized for being time-consuming and not sufficiently accurate [5, 6].

To overcome these shortcomings, several automated vision-based techniques for crack detection have been developed in the past few years. Classical computer vision approaches, such as image processing techniques (IPTs), have been introduced to the field of crack detection [7–12]. IPTs extract features from images through elaborately designed extractors and detect cracks using thresholds or trained classifiers. Still, a major concern is the reliance of prediction performance on the quality of hand-crafted extracted features. This could be inevitably limited by subjectivity and domain expertise [13]. Further, a great number of preprocessed images are required; this makes the detection process unadaptable, tedious, and inefficient. Moreover, these hand-crafted extracted features cannot distinguish between cracks and complex backgrounds in low-level image cues [14]; thus, they are less applicable in images with large variations.

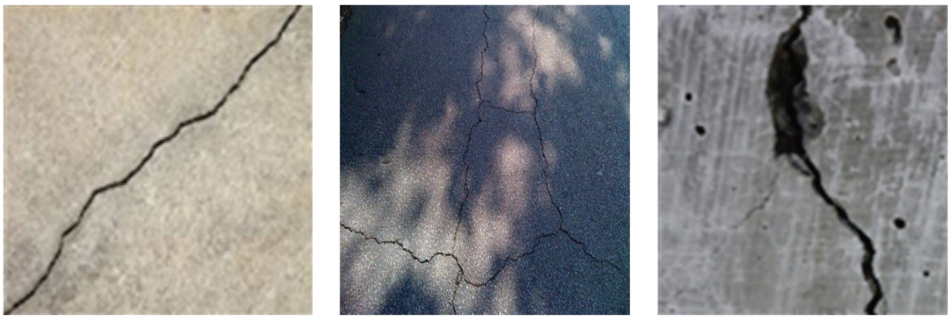
Currently, deep learning techniques are driving advances in computer vision to tackle the drawbacks of classical IPTs; they can automatically identify intricate structures of large-scale data using models with multiple processing layers [15–20]. Convolutional neural networks (CNNs) are the most widely used models for automated feature learning and supervised detection [21–26]. Multiple efforts have been made to implement CNN-related methods in pixel-level crack detection. Li et al. [27] employed a fully convolutional network- (FCN-) [28] based model for multiple damage detection, including cracks and spalling. Mei et al. [29] and Pan et al. [13] adopted DenseNet (as the backbone) together with loss functions and attention modules, respectively, to segment concrete cracks. The UNet [30] architecture has been used for concrete surface crack segmentation [31, 32]. Kang et al. [33] utilized an integrated model based on Faster R-CNN [34] and a modified IPT for crack detection and quantification. In addition, a two-level technique consisting of Faster R-CNN and Mask R-CNN [35] was devised for detecting and measuring the damage on historic glazed tiles [36]. In addition, the idea of digital twin of concrete structures has been brought up and damaged images have been considered as important data for it.

Nevertheless, cracks in engineering practice occur under various scenarios, and the current crack segmentation methods often obtain suboptimal detection results, due to the following reasons. First, most of them focus on crack detection across monotonous backgrounds, such as pure concrete surfaces and pavement surfaces. However, finding the optimal network architecture to segment cracks with such complex backgrounds and illumination is difficult, resulting in more realistic and practical problems. Secondly, the size of cracks varies dramatically, with an order of magnitude difference between small and large cracks. Furthermore, the size of certain discontinuous details and the major section of the crack differ significantly for discontinuous cracks; these differences are crucial for determining the current stable state of the fracture and whether it will continue to spread. Third, in practice, many cracks have complex topological shapes and very large differences in terms of width, as illustrated in Figure 1. The results of most existing methods tend to consist of blurry boundaries and inadequate segmentation, while accurate edge segmentation is the premise of obtaining the width crack.

The boundaries of cracks are crucial in crack segmentation, especially for width calculation. After being coupled with boundary detection, the crack segmentation can be treated as a multi-task learning (MTL) problem [37]. MTL could potentially improve segmentation performance if the associated tasks shared complementary information. The evidence has been provided in existing literature for certain pairs of tasks, i.e., detection and segmentation [34, 38], segmentation and depth estimation [39, 40], and segmentation and edge detection [41, 42]. Considering these observations, researchers started designing architectures capable of learning shared representations from multi-task supervisory signals, such as cross-stitch networks [43], multi-task attention networks [44], pattern-affinitive propagation networks [40], and multi-scale task interaction networks [45]. There have been several studies on the joint learning of semantic segmentation and boundary detection. Ding et al. [46] and Liew et al. [41] proposed to learn the boundary as an additional semantic branch to boost the segmentation performance for scene segmentation. Marmanis et al. [47] and Liu et al. [48] combined semantic segmentation with edge detection to reduce the semantic ambiguity in remote sensing tasks. These works demonstrate the validity of joint learning of segmentation and boundary detection. However, the joint learning of boundary detection and segmentation is seldom investigated in the crack segmentation task. Yamaguchi and Hashimoto [49] introduced a crack detection method for a concrete surface image based on a percolation model, yet hand-crafted features are needed with this approach. FCN and structured forests with wavelet transform (SFW) were combined to detect tiny cracks in steel beams [50]; among them, edge detection was performed using multi-scale structured forests and wavelet maximum modulus edge. Although edge detection was utilized to improve crack segmentation performance, SFW is time-consuming, and the proposed method is not an end-to-end deep learning approach.

Crack segmentation is a binary pixel-level classification task. Whether a pixel belongs to cracks largely depends on high-resolution representation, which contains low-level information of the image. Here, high resolution refers to the high-resolution representations in the segmentation neural network, like SegNet [51], U-Net [30], DeconvNet [52], and HRNet [53]. Unlike other segmentation tasks, such as medical and satellite images, high-resolution representation plays a key role in crack segmentation. However, most current CNN models tend to lose high-resolution details in complex scenes and lead to blurry object boundaries [34, 39].

To overcome these limitations and construct a novel crack assessment framework, this contribution introduces a two-stream neural network architecture for crack image segmentation under various scenarios, namely, boundary-aware crack segmentation (BACS) network. A high-resolution network (HRNet) [53, 54] is utilized in the segmentation branch with strong high-resolution representations for cracks in complex backgrounds and variable illumination. A dynamic feature fusion (DFF) network [55], which assigns different fusion weights for different input images and locations adaptively, is utilized in the edge branch to produce more accurate and sharper edge predictions. As a result, high-resolution features can be further improved with the aid of an edge branch to enhance the



(a) (b) (c)

Figure 1: Cracks in various situations: (a) pure crack; (b) complex background; (c) variable width.

quality of feature representation, contributing to a more accurate and efficient crack detection process for distinguishing the crack and non-crack at the pixel level.

This paper is organized as follows. In Section 2, the proposed BACS method composed of two branches is presented in detail. In Section 3, the BACS is tested in a concrete crack dataset, and a comparison with the state-of-the-art methods and quantification are presented. Finally, Section 4 summarizes the concluding remarks.

2. Methodology

Throughout this section, the proposed BACS for crack segmentation is presented. As depicted in Figure 2, it consists of two streams of networks. The first branch of the network, namely, segmentation branch, is HRNet. The segmentation branch is utilized to extract the overall semantic feature of images. The second branch, namely, edge branch, processes edge information in the form of semantic boundaries. The edge branch is enforced to only process boundary-related information using DFF. Semantic features from the segmentation branch are then fused with boundary features from the edge branch to produce a refined segmentation result, especially around boundaries. Next, each of the modules in our framework is described in detail. Codes and the dataset will be made available at https://github.com/GaoyangLiu/BACS.

2.1. Segmentation Branch. HRNet is adopted as the backbone of segmentation branch because it has two advantages in comparison to existing networks for segmentation tasks. First, it connects high-to-low resolution subnetworks in parallel, rather than in series, as it is done in most existing solutions. Thus, it is possible to maintain the high resolution instead of recovering the resolution through a low-to-high process, and accordingly, the predicted result is potentially more precise spatially. Secondly, most existing fusion schemes aggregate low-level and high-level representations. Instead, HRNet performs repeated multi-scale fusions to boost the high-resolution representations with the help of the low-resolution ones of the same depth and similar level, and vice versa; this results in high-resolution representations that are also rich for crack segmentation. Consequently, the predicted crack segmentation result is potentially more accurate, especially boosting performance on thin and small objects in complex backgrounds. The architecture is illustrated in Figure 3.

HRNet starts from a high-resolution subnetwork as the first stage and gradually adds high-to-low resolution subnetworks one by one; this strategy forms new stages and connects the multi-resolution subnetworks in parallel. As a result, the resolutions for the parallel subnetworks of a later stage consist of the resolutions from the previous stage and an extra lower one. The exchange units across parallel subnetworks are introduced in a way that each subnetwork repeatedly receives information from other parallel subnetworks. Below, there is an example showing the scheme for exchanging information. The third stage is divided into three exchange blocks, and each block is composed of three parallel convolution units with an exchange unit across the parallel units, which is shown in Figure 4.

In Figure 4, C_{sr}^{b} represents the convolution unit in the r^{th} resolution of the b^{th} block in the s^{sh} stage, and ε_{s}^{b} is the corresponding exchange unit. The semantic information among different branches exchanges in the exchange unit. The aggregation of information by exchange unit is illustrated in Figure 5.

The exchange units consist of upsampling and downsampling operations across various resolutions. In contrast to most existing fusion schemes that aggregate low-level and high-level representations, HRNet repeatedly performs multi-scale fusions to boost the high-resolution representations. This strategy is suitable for crack segmentation, where the high-resolution feature plays an important role with regard to boundary accuracy.

2.2. Edge Branch. The goal of edge branch is to extract object boundaries for guiding the segmentation of boundaries and thin structures of cracks. To prevent a large loss of image details due to input downsampling, and particularly for elongated thin parts, the edge stream processes the original images directly without resizing. To facilitate edge learning, the image gradient is appended in the input tensors, which can be easily computed using the Sobel filter [41, 56, 57]. The commonly used 3×3 convolution kernels are adopted to compute the horizontal and vertical gradients $\mathbf{G}_x$ and $\mathbf{G}_y$, as illustrated in the following equations:

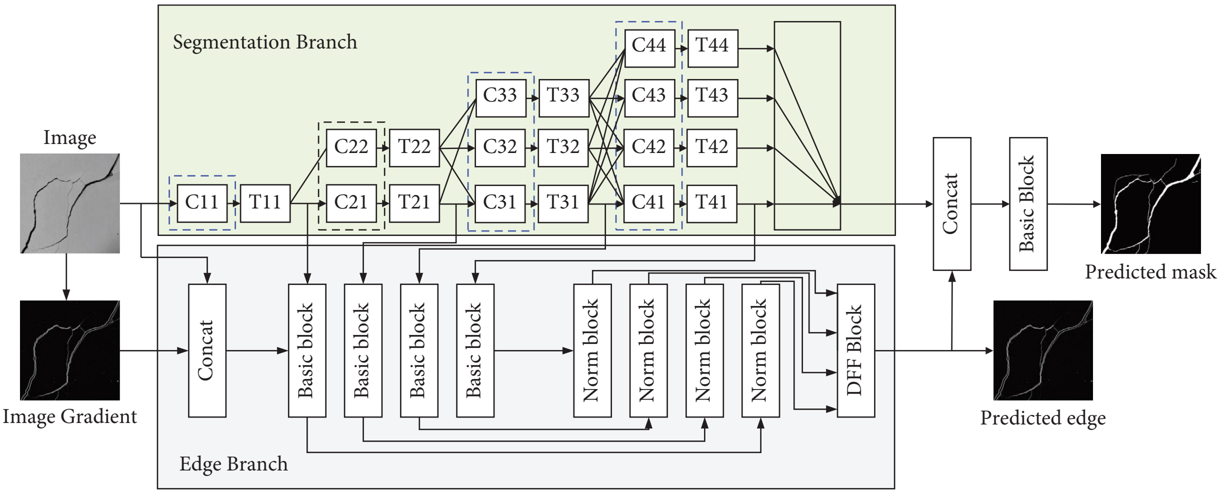


Figure 2: Architecture of the proposed BACS.

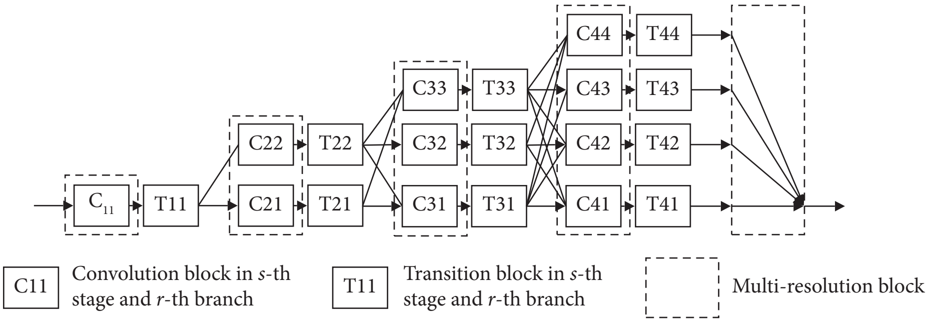

C11
T11
C21
C22
T21
T22
C31
C32
C33
T31
T32
T33
C41
C42
C43
C44
T41
T42
T43
T44
C11
Convolution block in s-th stage and r-th branch
T11
Transition block in s-th stage and r-th branch
Multi-resolution block

Figure 3: Architecture of HRNet in segmentation branch.

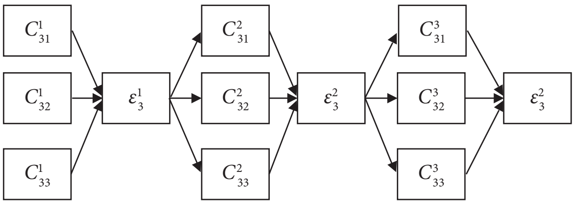


Figure 4: Exchange units of the third stage for information exchange across parallel subnetworks.

$$\mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{I}, \tag{1}$$

$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{I}, \tag{2}$$

where $\mathbf{I}$ is the input image. The magnitude of a single channel is obtained by

$$\mathbf{G}_i = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}. \tag{3}$$

This procedure is applied to each of the RGB channels. The image gradient is the square root of the gradient summation of each channel.

$$\mathbf{G} = \sqrt{\mathbf{G}_R^2 + \mathbf{G}_G^2 + \mathbf{G}_B^2}. \tag{4}$$

Finally, the gradient is normalized to the range $[0, 1]$. Some typical crack images and corresponding gradients for the Sobel filter are shown in Figure 6. It can be seen that the gradients of cracks with the clean background are with less noise. However, the Sobel filter does not consider the context of a pixel, so cracks with complex backgrounds tend to produce more noise in the gradient output. The gradient channel is appended to the RGB image as the fourth channel. The concatenated image together with four feature maps from HRNet in different stages is then fed into basic blocks to include information of different stages. Then, norm blocks are utilized to reduce the channels to the predicted categories, which is 1 in this study to denote whether a pixel is a crack or not.

The features from multiple scales can greatly benefit the semantic edge detection task if they are well fused. However, the prevalent semantic edge detection methods apply a fixed weight fusion strategy where images with different semantics are forced to share the same weights, resulting in universal fusion weights for all images and locations regardless of their

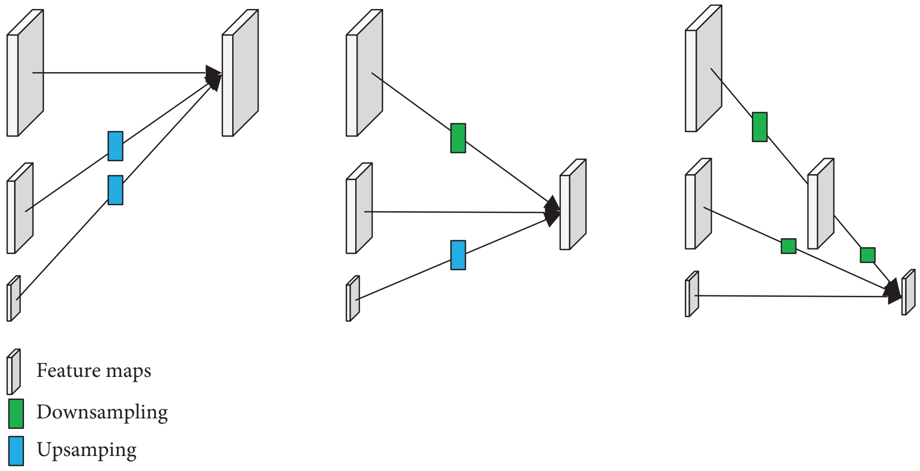
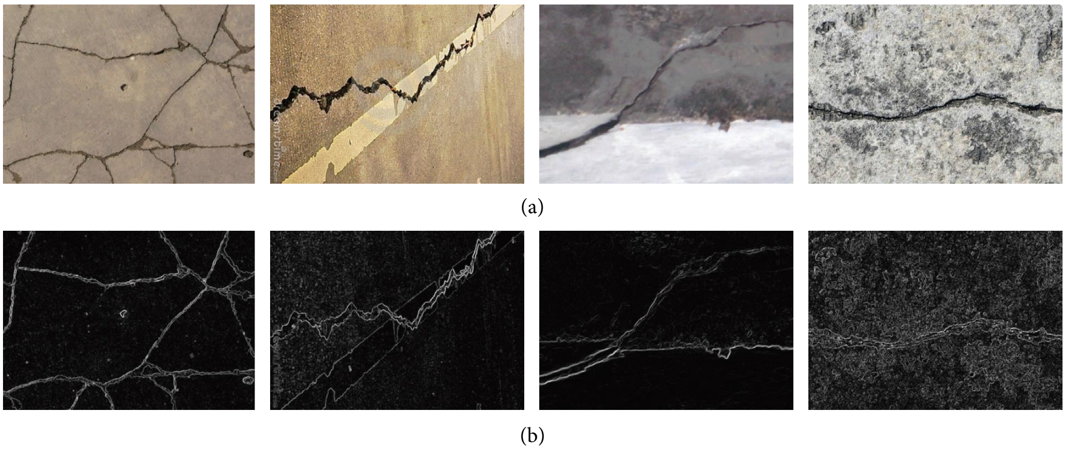


Figure 5: Aggregation of information for high, medium, and low resolutions.


(a)

(b)

Figure 6: Crack images and the corresponding image gradients from Sobel filter: (a) crack images; (b) image gradients from Sobel filter.

different semantics or local context. The DFF strategy [55], which assigns different fusion weights for different input images and locations adaptively, is adopted in this study. This is achieved using a weight learner that infers proper fusion weights over multi-level features for each location of the feature map, and it is conditioned on the specific input. In this way, the heterogeneity in contributions made by different locations of feature maps and input images can be better considered and thus help produce more accurate and sharper edge predictions. The detailed architecture of different blocks and edge detection procedure by DFF in the edge branch is illustrated in Figure 7.

The feature maps from different stages are fed into the edge branch. After being processed by basic blocks and norm blocks, four edge feature maps (F_{e1}, F_{e2}, F_{e3}, F_{e4}) of 1 channel are generated with information from the early to latter stages of segmentation branch. To better consider the heterogeneous contributions of feature maps, F_{e4} is fed into the adaptive weight learner to infer proper fusion weights over multi-stage features. $F_{\mathrm{e1}} \sim F_{\mathrm{e4}}$ are concatenated into a four-channel feature map. Then, element-wise multiplication and category-wise summation are applied to produce the final edge prediction.

2.3. *Training Loss.*

Both segmentation and edge branches are trained with a binary cross entropy (BCE) loss, since there are only two categories, crack and non-crack, in the dataset. The similarity between predicted probabilities and ground truth is measured by

$$\mathscr{L}_{\mathrm{BCE}}(\mathbf{p}, \widehat{\mathbf{p}}) = -\sum_{1}^{N} \left(p_i \ln\left(\widehat{p}_i\right) + \left(1 - p_i\right)\ln\left(1 - \widehat{p}_i\right)\right), \quad (5)$$

where $\widehat{\mathbf{p}}$ is the predicted probability, $\mathbf{p}$ is the ground truth label, and N is the total number of pixels. The ground truth of segmentation branch is the label of the entire image. The edge branch is supervised by the label of the edge only. The final loss is the summation of losses from segmentation branch and edge branch. That is to say, edge branch serves as an auxiliary task [58] for crack semantic segmentation. As a kind of additional regularization, edge branch is expected to boost the performance of the ultimately desired main task.

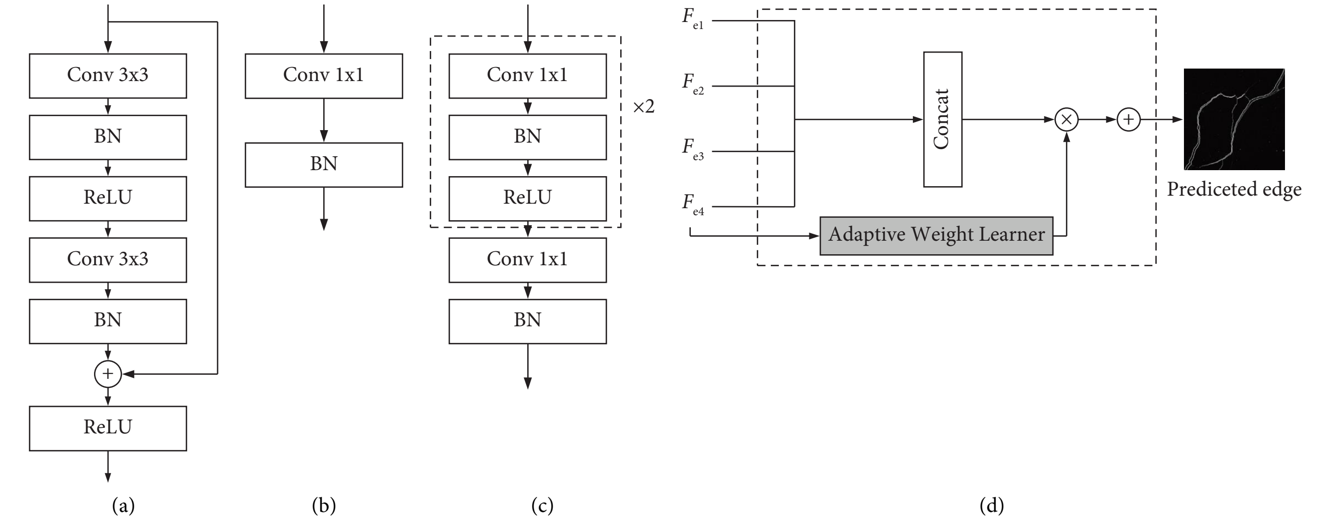


Figure 7: Edge detection procedure in edge branch relying on the DFF: (a) the architecture of basic block; (b) the architecture of norm block; (c) the architecture of adaptive weight learner in edge branch; (d) the process of feature maps in DFF.

2.4. Task Affinity between Crack Segmentation and Edge Detection. In contrast to the single-task methods, joint-task learning methods yield a promising direction to improve predictions by utilizing task correlation information to boost each other. However, the joint learning of multiple tasks can lead to negative transfer, leading to performance degradation of a single task if information sharing happens between unrelated tasks. The key point is the degree to which tasks share common structures. A statistical analysis [59] on those second-order patterns across boundary detection and crack segmentation is performed to quantify pixel affinities. Semantic pixels are considered similar when they belong to the same category. The matching number of those similar pairs is accumulated with the same space positions across the two types of corresponding images.

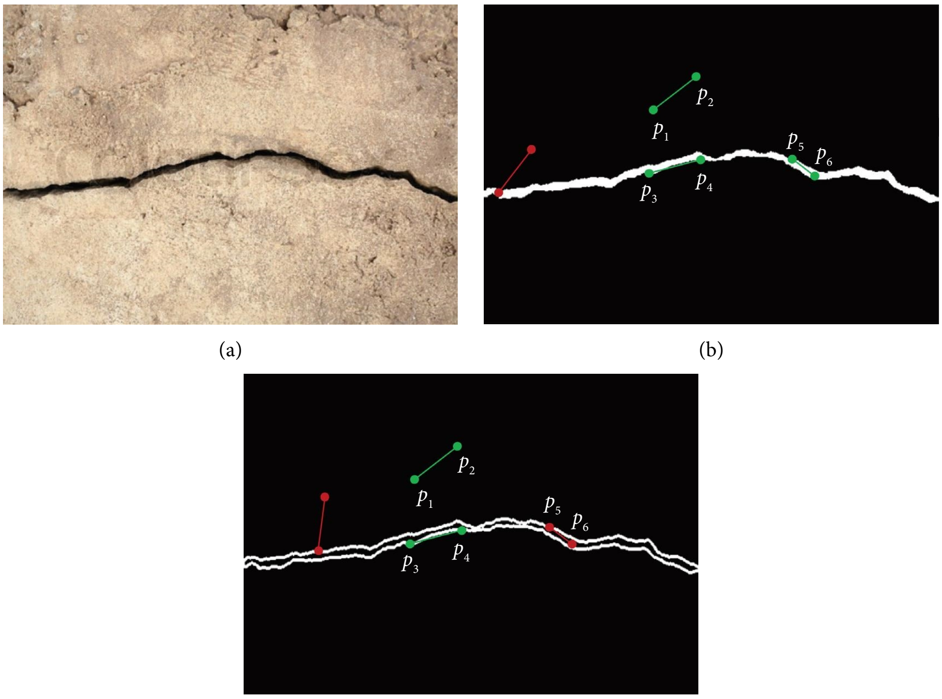
As shown in Figure 8, the affinity pairs (green points) at the common positions may exist in different tasks. Meanwhile, some common dissimilar pairs (red points) exist across tasks. Take the affinity pairs in Figure 7 for example; pixels (p_1, p_2) in the background and pixels (p_3, p_4) in the crack labels are affinity pairs in both segmentation and edge labels. Pixels (p_5, p_6) with p_5 in the edge and p_6 in the crack label are affinity pairs in segmentation labels while dissimilar pairs in edge labels. The rate of matched affinity pairs can be calculated by counting the matched pairs across segmentation and edge labels and then dividing them by the number of all pixel pairs. The rate of matched affinity pairs is 89.4% in this study. The statistical result shows that nearly all pairs across two tasks are of high affinity, which indicates that crack segmentation and edge detection share common structures in images. Therefore, the edge branch has the potential to boost the performance of the segmentation branch.

3. Results and Discussion

3.1. Dataset. To improve the generalization and demonstrate the superiority of the proposed method in various scenarios, a crack segmentation dataset consisting of three different scenarios is built for this study. The crack images that are collected from existing literature, the Internet, and taken by our team are 1,892 in total. The dataset is divided into three scenarios: pure cracks, complex background, and variable width, containing 1090, 432, and 370 images, respectively. The cracks in the pure crack scenario are relatively clear with a relatively large width, without background noise and illumination interference. Most of the images in the complex background scenario have complex backgrounds, such as spraying, water stains, honeycomb pitted surfaces, and other objects. Most of the cracks in the variable-width scenario have complex topological shapes that are difficult to segment, such as extremely thin cracks, cracks with large width differences, and other cracks with complex shapes. The images were divided into two main subsets: a training set with 1514 images and a testing set with 378 images. Each image is made available to a pixel-wise segmentation map, which operates as a mask covering the crack regions. All of the images have a fixed size of 256 × 256 pixels. Some examples of typical images corresponding to the three scenarios are illustrated in Figure 9.

To add segmentation masks to the crack images, the annotation tool “LabelMe” [60] is utilized for manual annotation. Users have the option to zoom in, zoom out, and annotate a crack by clicking along the boundary to get precise boundary labels. Figure 10 demonstrates several examples of images used in the concrete crack dataset, where the first, second, and third columns stand for original images, images with manual labels, and the corresponding ground truth, respectively.

After pixel-level annotation for the ground truth of the cracks, edge labels are extracted by applying Euclidean distance transformation. Euclidean distance transformation returns the distances to the closest background pixels. With labeled crack masks as input, the smaller distances correspond to pixels closer to the edge of the binary object. To avoid discontinuity, the width of edges is set to 2 pixels. The



(a)

(b)

(c)

Figure 8: Pixel affinities between crack segmentation and edge detection: (a) the crack image; (b) segmentation labels; (c) edge labels.

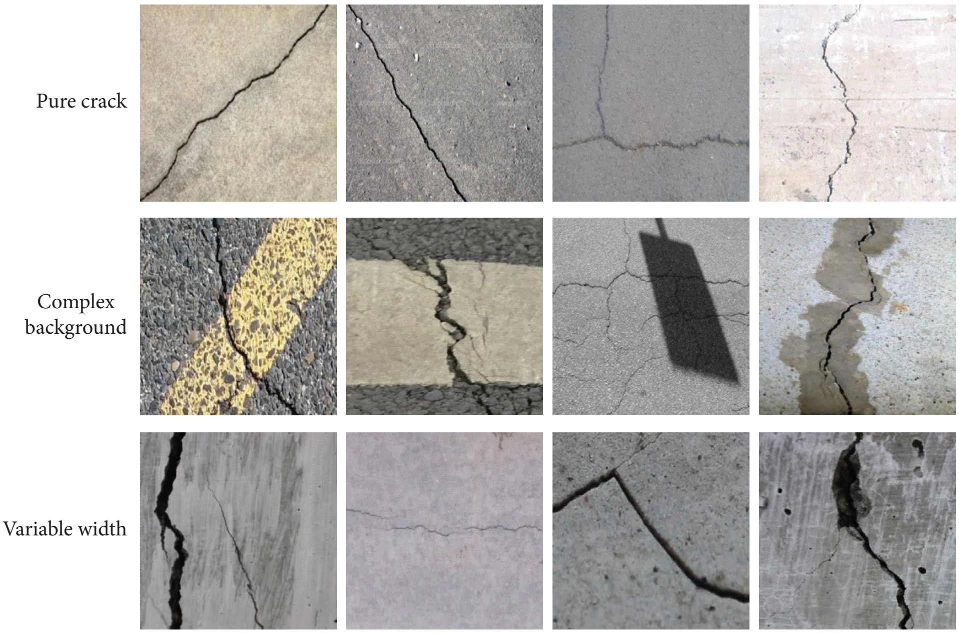


Figure 9: Images in the dataset under three different crack scenarios.

ground truth of a crack and corresponding edges is illustrated in Figure 11.

3.2. Training Process. The developed model is implemented using PyTorch and trained on Nvidia GeForce 1080TI GPU with a memory of 11 GB. Transfer learning is utilized in the segmentation branch with pretrained weights of HRNet on ImageNet to boost the performance and accelerate the training procedure. Data augmentation includes image flip, rotation, and translation. There are several hyperparameters in network training, among which the learning rate is regarded as the most important one to tune [61]. For training deep neural networks, selecting a good learning rate is essential for both better performance and faster convergence. Optimizers that adjust automatically the learning rate

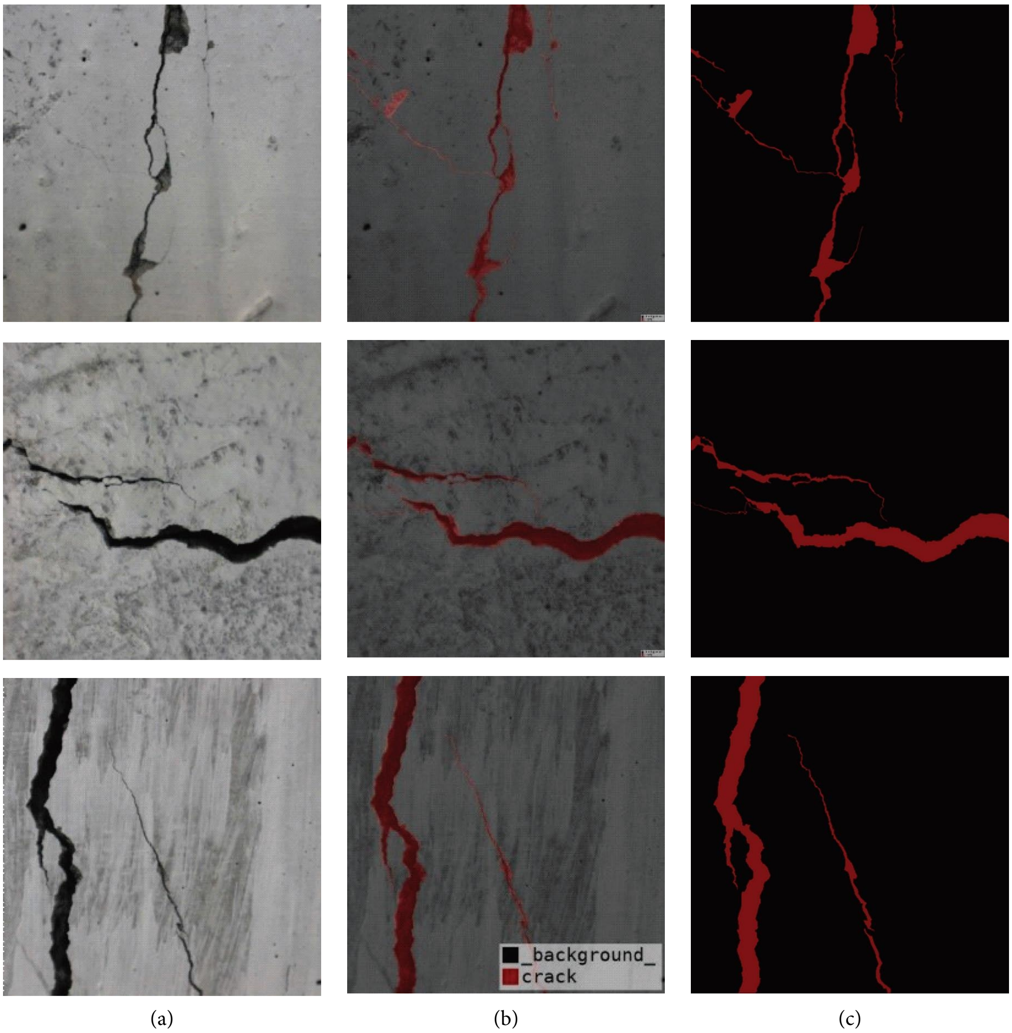


Figure 10: Examples of images used in the concrete crack dataset: (a) original images; (b) images with manual labels; (c) ground truth.

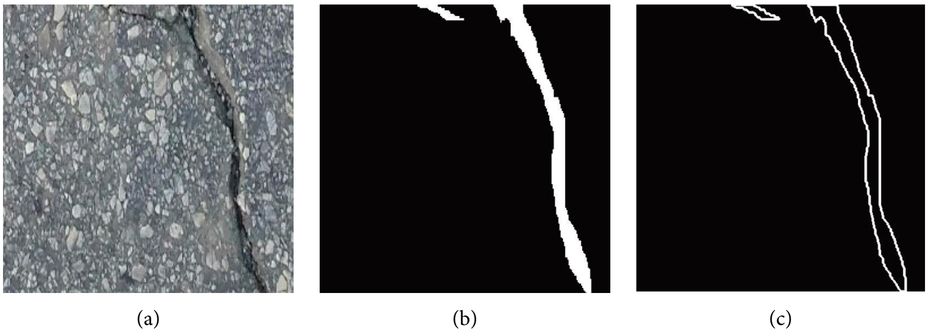


Figure 11: A crack image and its corresponding ground truth of the crack and edges: (a) the crack image; (b) the crack mask of this image; (c) the edge mask of this image.

can benefit from more optimal choices. To reduce the amount of guesswork regarding the choice of a good initial learning rate, a learning rate finder [62] is utilized in our experiments. Only one epoch is carried out starting with a very low learning rate ($10e-8$ in this study) to a very high learning rate ($10e-1$). The learning rate is increased after each processed batch, and the corresponding loss is logged as shown in Figure 12. The loss decreases at the beginning, and then it stops and goes back increasing extremely quickly. It should be noted that the learning rate that corresponds to the minimum loss value is a bit too high, since we are at the edge between improving performance and gradient explosion. The best learning rate is the point on the graph with the fastest decrease in the loss [62], which is around the middle of the steepest descending loss curve. In this study, the best learning rate is $5.2e-4$ for BACS.

With the optimal initial learning rate, Adam [63] is employed as the optimizer. The training schedule of the learning rate is multiplied by 0.8 every 50 epochs. To avoid the problem of running out of memory, the batch size is set to be eight during the training and validation processes. Figure 13 depicts the change in loss value during the training and validation processes, which shows that the training process gradually converges after about 60 epochs.

3.3. Comparison Study with Other Neural Networks. For experimental evaluation, five typical models are compared with BACS: (1) UNet [30], a traditional encoder-decoder architecture for segmentation; (2) UNet++ [64], an enhanced UNet model through a series of nested skip connections; (3) DeepLabV3+ [65], an encoder-decoder network with atrous convolution; (4) DeepCrack [66], a deep hierarchical network for crack segmentation; and (5) Crack-FPN [18], a modified FPN for crack segmentation. The mean intersection-over-union (mIoU) scores of different scenarios are shown in Table 1.

In comparison with the other four methods, BACS shows superior performance with a similar number of parameters. Under the pure crack scenario, all methods reach fine results. Yet, in complex backgrounds and variable-width scenarios, BACS obtains much better mIoU than the other methods. Specifically, the BACS is 7.13% and 12.03% higher than the lowest UNet model under complex background and variable-width scenarios, respectively. Latency in the right column denotes the seconds per image. It should be noticed that latency is highly dependent on the network architecture and hardware. Networks with complex architecture and skip connections, such as DeepLabV3++ and UNet++, are more likely to have large latencies. DeepCrack obtains the lowest latency (79 ms) among all the models. The latency of BACS, 172 ms, is larger than the backbone HRNet-w32 due to the edge branch. Overall, the latency is acceptable in a single 1080TI GPU. It is shown that benefiting from HRNet in the segmentation branch, the fused high-resolution features can be treated as neutralization which aggregates the multiple-level features from coarse to fine. Additionally, the edge branch with the DFF can maintain the boundary information, which is critical for the crack segmentation task. Therefore, the results of BACS achieve a significant performance improvement, especially in the complex background and variable-width scenarios.

To further investigate the performance of the proposed BACS compared with DeepCrack, they are trained and validated on the original DeepCrack dataset. The result shows that BACS achieves mIoU of 82.31%, 6.89% higher than that of DeepCrack.

Some sample images from the three different scenarios and the results for all methods are shown in Figures 14–16. The mIoU values (%) are shown on the top of each prediction. Under the pure crack scenario, all methods perform well.

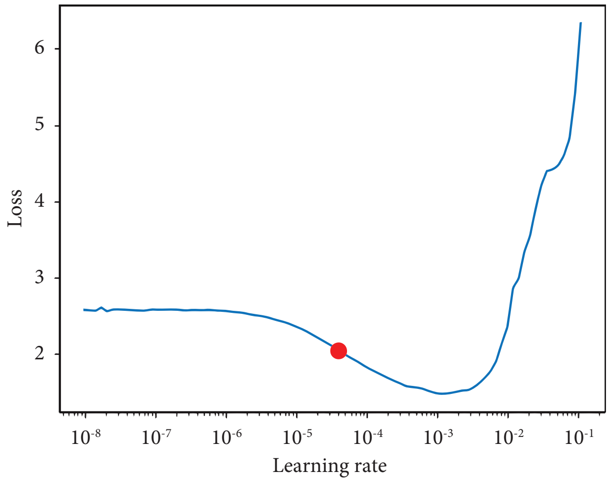


Figure 12: Loss variation as the change of learning rate to get the optimal initial learning rate.

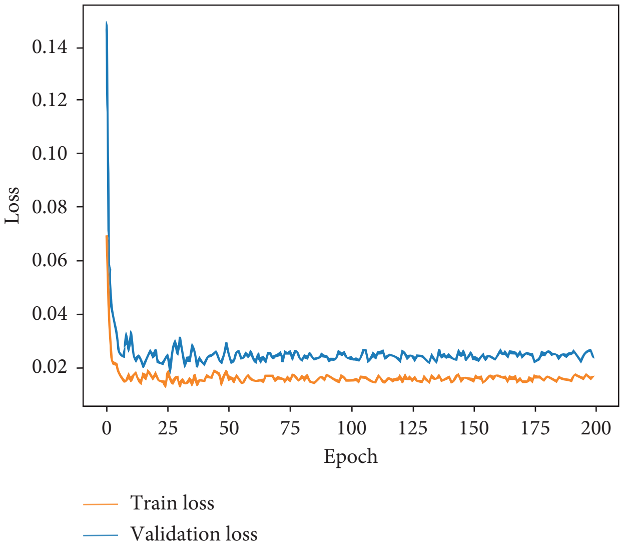


Figure 13: Decline of loss in training and validation process.

Under the complex background scenario, the BACS performs better than the other methods. Owing to the rich high-resolution features of the HRNet, the BACS produces fewer false-positive predictions. Moreover, the edge of the predicted crack is sharper and crisper, benefiting from the edge branch fused information. With the aid of the edge branch and DFF, BACS produces crisper and more precise segmentation results, especially along thin cracks. In Figure 17, a typical image under the variable-width scenario is illustrated; for better presentation, only the results of BACS and DeepCrack are presented. The width of cracks ranges from 1 to 32 pixels, while crack widths of 2 and 15 pixels are shown in Figure 17(a). It can be seen that BACS is able to detect very narrow cracks.

3.4. Validity Analysis of the Edge Branch

3.4.1. Ablation Study. In this section, the edge branch, DFF block, and edge supervision are thoroughly analyzed to further understand their operation.

Table 1: Segmentation results of mIoU on the test set of the crack dataset.

Models	Backbone	No. of parameters (million)	mIoU (%)			Latency (ms)
			Pure crack	Complex background	Variable width	
UNet	ResNet50	32.46	73.51	61.61	58.28	147
UNet++	ResNet50	34.92	73.69	63.84	62.24	192
DeepLabV3+	ResNet50	26.68	74.91	65.17	62.26	274
DeepCrack	VGG16	25.27	72.68	61.74	65.04	**79**
Crack-FPN	ResNeXt50	34.71	74.98	64.56	64.90	166
HRNet-w32	—	31.17	76.62	66.38	67.84	121
BACS	**HRNet-w32**	**32.82**	**79.26**	**68.74**	**70.31**	172

The bold values are the optimal ones in each column.

In the segmentation branch, the HRNet maintains high-resolution representations by connecting high-to-low resolution convolutions in parallel and repeatedly conducting multi-scale fusions across parallel convolutions. The resulting high-resolution representations are robust and spatially precise. Thus, the baseline mIoUs are relatively high in all three scenarios, as shown in Table 2.

When comparing models with and without the edge branch, the edge information plays a crucial role in the variable-width scenario. The edge stream helps guide the segmentation of thin cracks. Nevertheless, one may argue that the performance gain from adding the edge branch is partially due to the increased number of parameters. Therefore, the two sources of performance boost are disentangled by comparing with a baseline whose network architecture is the same as the BACS, while the edge supervision is replaced with mask segmentation supervision. The ground truth of the edge branch is the same as segmentation branch with mask labels. As illustrated in Section 2.3, the final loss is the summation of BCE losses of segmentation branch and edge branch. In the experiment of edge supervision, the loss of edge branch is calculated by the BCE loss of predictions and the mask labels. This procedure aims to validate the performance improvement of the edge branch compared with the network with the same number of parameters. A comparison study to investigate the effect of the Sobel filter is also conducted. Without the Sobel filter, the model performs worse than that with the Sobel filter, particularly in the pure crack scenario. This is due to the fact that the Sobel filter tends to yield image gradients with less noise in the pure crack scenario, which provides effective additional information to the edge branch. A slight performance drop is noticed in all three scenarios in the dataset when removing edge supervision. This verifies our finding that the edge branch information is essential to addressing the crack segmentation task.

3.4.2. Feature Maps among the Two Branches. To further illustrate the BACS in detail and validate its effectiveness qualitatively, some intermediate feature maps are presented in Figure 18. Crack images are fed into the segmentation branch, yielding four output feature maps from the high-resolution branch of the HRNet. Four feature maps of each stage are shown in the downside of the segmentation branch. The different stages of the convolutional layers obtain different features with various levels of information. Low layers kept more low-level information; thus, the boundary of the crack and other dots is clear in the first two feature maps from the segmentation branch. However, low layers focus more on the sharp contrast of images, thereby containing a large amount of noise. The top layers obtained more abstract and global features, which are composed of low-level features. These global features contain much more semantic and context information, which helps determine whether a pixel belongs to the crack or background. Since cracks are often long and thin in noisy backgrounds, the segmentation performance is more sensitive to low-level information compared with some other segmentation tasks. Hence, after maintaining high resolution and repeatedly performing multi-scale fusions, the HRNet in the segmentation branch is suitable and superior for the crack segmentation task.

Subsequently, four feature maps are fed into the edge branch together with the concatenation of crack images and gradient. The norm blocks of the edge branch produce raw edge outputs, which are further fine-tuned by the DFF block, as illustrated in Figure 18. The edge prediction of the DFF block is much better than the raw edge branch output. The final loss is a summation of the segmentation and edge loss. As an MTL problem, combining the segmentation and edge losses can boost the performance of crack segmentation, which is validated in Section 3.4.1. Considering the importance of low-level information in crack segmentation, the segmentation and edge branch outputs are concatenated together and passed to a basic block to produce the crack segmentation result. The concatenated tensor with five channels is fed into a basic block to produce the final one-channel prediction with the same resolution. With the aid of the edge branch, the BACS yields more accurate crack segmentation results and precise crack boundaries.

3.5. Crack Width Quantification. To validate the accuracy of the proposed method in engineering practice, crack widths on various concrete surfaces are calculated based on BACS and compared with the widths obtained by a crack width observer. These crack images are obtained using a smartphone in different locations. The crack images and other measured widths are shown in Figure 19.

There are mainly two steps for quantifying the chosen cracks, i.e., obtaining the pixel widths and mapping them to actual widths in the measurement unit. In the first step, the crack images are fed into the BACS to get the predicted

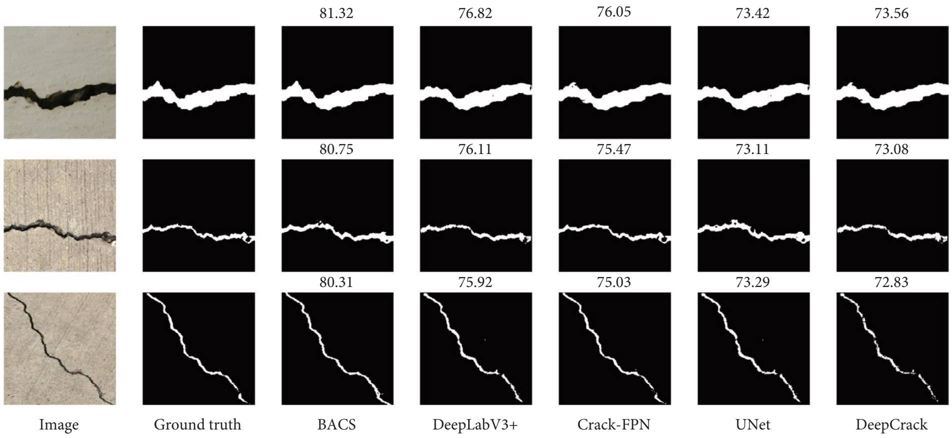


Figure 14: Comparison between different models under pure crack scenario.

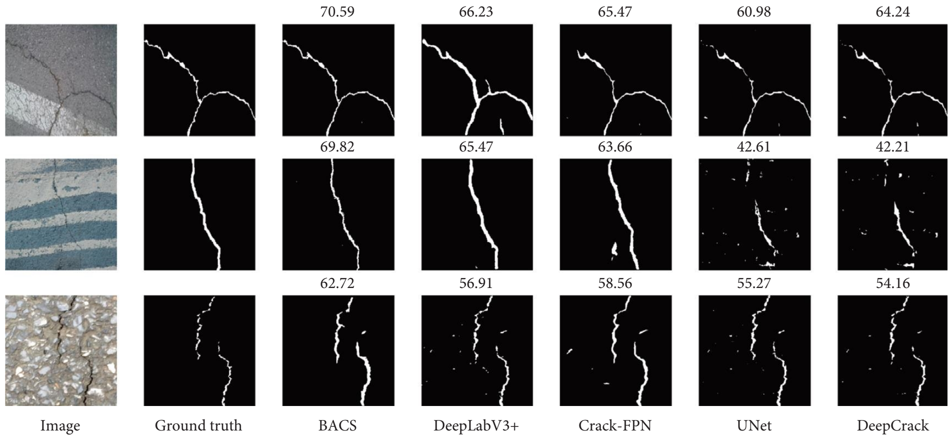


Figure 15: Comparison between different models under complex background scenario.

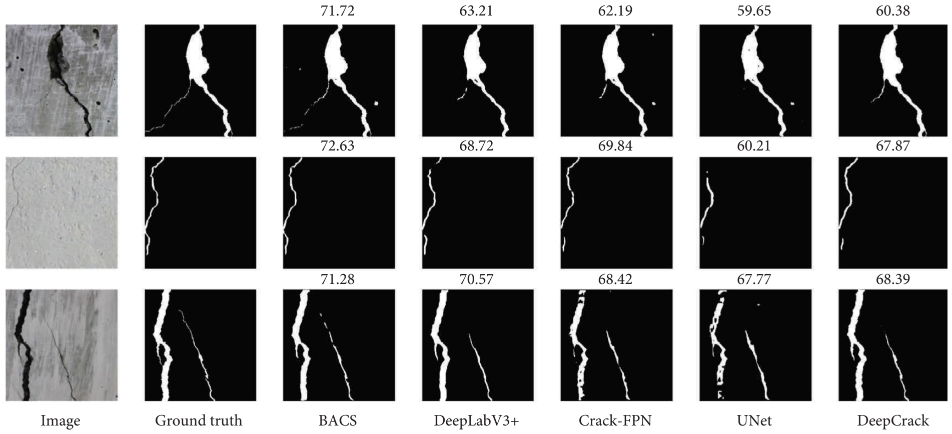


Figure 16: Comparison between different models under variable-width scenario.

cracks, as shown in the second column in Figure 20(b). Then, each crack instance is skeletonized using the medial axis thinning algorithm to extract a one-pixel-wide centerline. Since the width of a crack often varies along the crack, the crack widths are evaluated at specific pixels on the centerline. For a query pixel on the crack centerline, the crack widths are computed as shown in Figure 20: (1) the orientation of the crack at the centerline is calculated by fitting a line to the

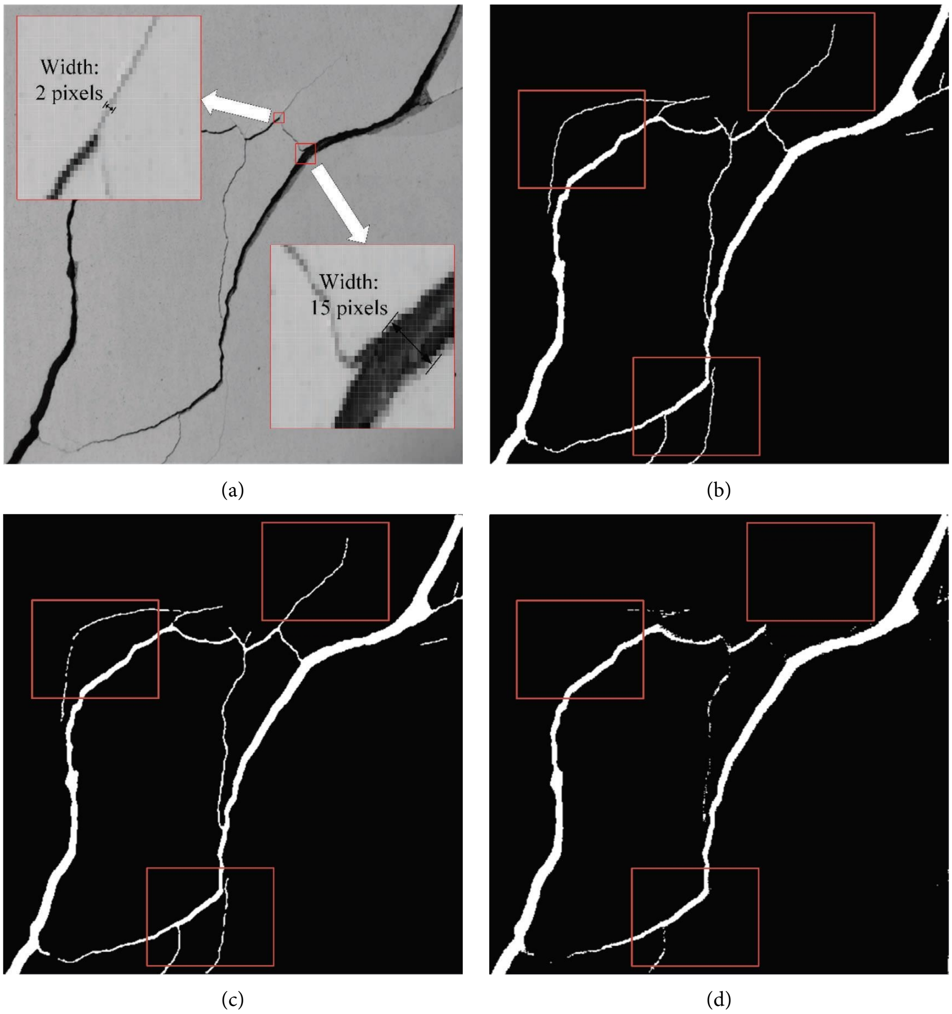


Figure 17: A typical image under the variable-width scenario: (a) image; (b) ground truth; (c) BACS; (d) DeepCrack.

Table 2: Results of ablation study for the edge branch and edge supervision.

Models	Edge branch	DFF	Edge supervision	Sobel filter	mIoU (%)		
					Pure crack	Complex background	Variable width
HRNet	No	No	—	—	75.62	66.38	66.84
BACS	Yes	No	Edge	Yes	75.71	67.82	69.69
BACS	Yes	Yes	Mask	Yes	77.96	68.02	69.84
BACS	Yes	Yes	Edge	No	77.02	67.93	68.62
BACS	Yes	Yes	Edge	Yes	**79.26**	**68.74**	**70.31**

The bold values are the optimal ones in each column.

pixel and its neighboring pixels on the centerline; (2) a line normal to the crack orientation is then created; (3) at both sides of the crack centerline, the crack boundary pixel that is closest to the line is extracted; and (4) the distance between the two pixels is calculated as the crack width in pixels.

The second step is to map pixel widths to actual widths in the measurement unit, where the pixel ratio R (pixel/mm) between the number of crack pixels and the actual width of the crack is necessary. Considering that the ratio often changes over the distance of the smartphone camera from the surface of the detected concrete, the relation between the ratio and distance is calibrated under laboratory conditions. As shown in Figure 21, to calibrate the relation between pixel width and distance from the smartphone to the surface of the detected target, experiments are performed in a quasi-static process. The fitted curve of the relation between the pixel ratio (R) and distance is illustrated in Figure 22, with which the pixel ratio of any working distance could be obtained.

Finally, the actual crack widths are calculated using the following formula:

$$w(\mathrm{mm}) = \frac{p(\mathrm{pixel})}{R(\mathrm{pixel/mm})}, \tag{6}$$

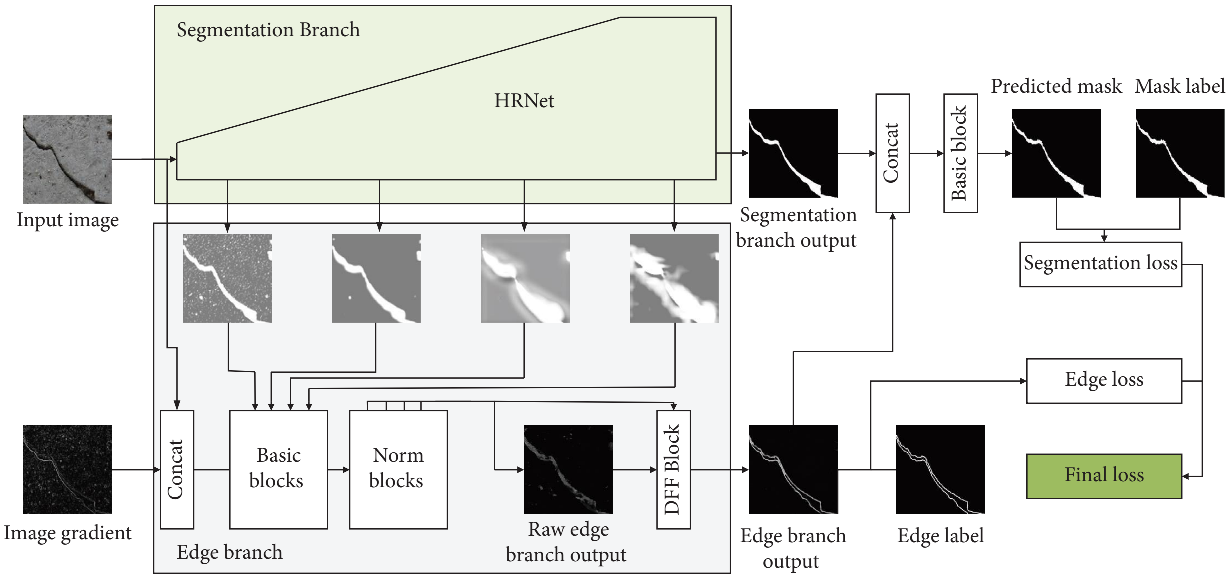
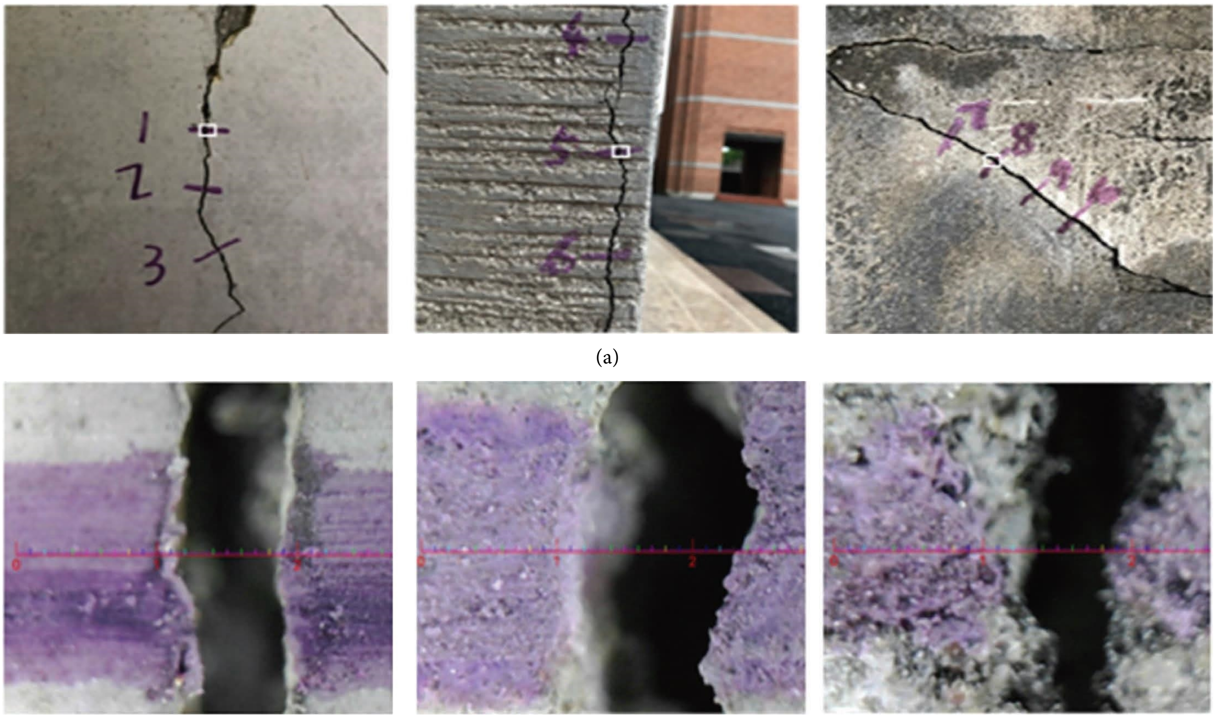


Figure 18: Graphical representation of feature maps in the proposed BACS.



(a)

(b)

Figure 19: Crack images and widths measured by crack width observer: (a) images; (b) widths by crack width observer.

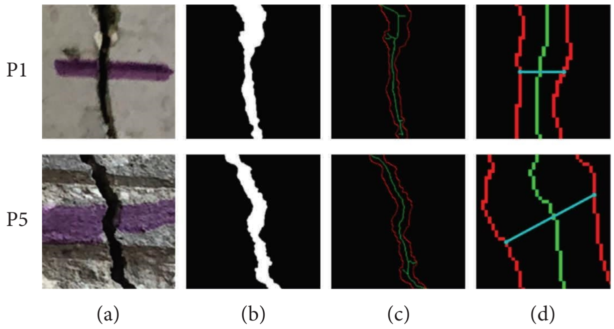


Figure 20: Crack quantification in pixels of point 1 (P1) and point 5 (P5): (a) input crack images; (b) predicted cracks; (c) centerlines; (d) crack widths.

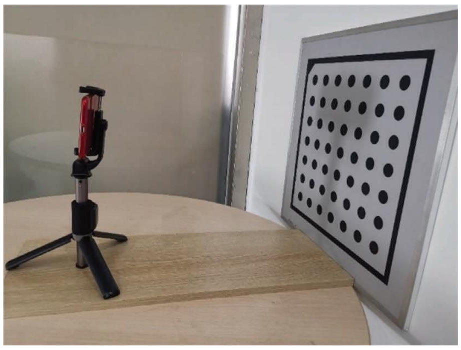


Figure 21: Calibration experiment apparatus.

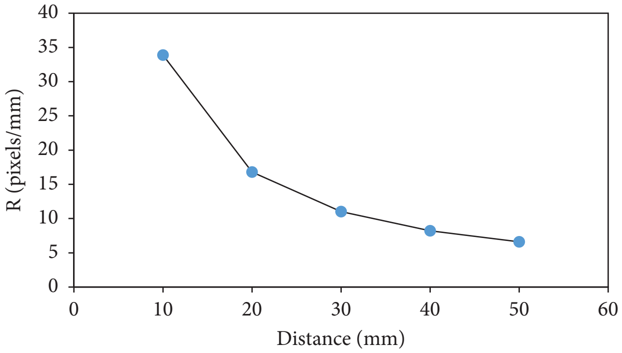


Figure 22: Fitted curve of the relation between ratio and distance.

Table 3: Crack width measurement results.

Point index	Actual width (mm)	BACS pixel width (pixel)	BACS width (mm)	Absolute error (mm)	Error rate (%)
1	1.046	30	1.167	0.121	11.57
2	0.865	25	0.989	0.124	14.34
3	0.783	22	0.884	0.101	12.90
4	1.157	28	1.103	0.054	4.67
5	1.354	36	1.438	0.084	6.20
6	1.021	24	0.958	0.063	6.17
7	0.817	23	0.924	0.107	13.10
8	0.904	26	1.018	0.114	12.61
9	1.883	51	2.011	0.128	6.80
10	2.102	56	2.198	0.096	4.57
Average	1.1932	32	1.269	0.0992	9.29

where w is the actual width of the crack and p is the pixel width.

In the experiment, crack widths at ten points of three different cracks are calculated. With a working distance of 150 mm, the pixel ratio is 25.35 pixels/mm as estimated by the curve shown in Figure 22. The pixel widths predicted by BACS and actual widths obtained by equation (6) are shown in Table 3.

Table 3 shows that BACS achieves high accuracy with an error rate of 9.29%. The average absolute error of the BACS is 0.0992 mm, which is approximately two pixels in the images.

4. Conclusions

A novel two-stream boundary-aware crack segmentation (BACS) network is proposed in this study, which combines semantic segmentation with semantically informed edge detection explicitly. The segmentation branch using HRNet aims to acquire strong high-resolution representations for cracks in complex backgrounds in engineering practice. Additionally, a modified dynamic feature fusion (DFF) network is adopted as the edge branch to boost the performance in elongated thin cracks. The mIoU in a crack dataset consisting of different scenarios indicates that the edge branch significantly improves semantic segmentation. The conclusions are summarized as follows [49]:

(1) With the aid of HRNet in the segmentation branch that maintains high resolution instead of recovering the resolution through a low-to-high process, BACS reaches high performance in crack segmentation with both clean and complex backgrounds.

(2) Edge branch in BACS that integrates DFF preserves fine-grained details, especially for elongated thin cracks. Based on the evaluation metric mIoU, BACS yields the best value of 70.67% under the variable-width scenario.

(3) BACS is a feasible and precise deep learning model for crack quantification at arbitrary working distances. With the crack segmentation results, the widths obtained by our approach are close to the actual values, with an error rate of 9.29%. The average absolute error of BACS is 0.0992 mm, which is approximately two pixels in the images.

(4) The proposed method shows superior performance for the crack segmentation task under difficult conditions, especially in the variable-width scenario. The findings show a new way of structural inspection and safety assessment of concrete structures, providing an accurate data foundation for the digital twin of concrete structures.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to gratefully acknowledge the China Postdoctoral Science Foundation (2021M702866), the National Natural Science Foundation of China (52108179), and the Key R&D Program of Zhejiang (2023C01161).

References

[1] X. Zhang, D. Rajan, and B. Story, "Concrete crack detection using context-aware deep semantic segmentation network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 11, pp. 951–971, 2019.

[2] A. C. Neves, J. Leander, I. González, and R. Karoumi, "An approach to decision-making analysis for implementation of structural health monitoring in bridges," *Structural Control and Health Monitoring*, vol. 26, no. 6, Article ID e2352, 2019.

[3] P. Spyridis and N. Mellios, "Tensile performance of headed anchors in steel fiber reinforced and conventional concrete in uncracked and cracked state," *Materials*, vol. 15, no. 5, p. 1886, 2022.

[4] K. Agathos, K. E. Tatsis, K. Vlachas, and E. Chatzi, "Parametric reduced order models for output-only vibration-based crack detection in shell structures," *Mechanical Systems and Signal Processing*, vol. 162, Article ID 108051, 2022.

[5] C. Z. Dong and F. N. Catbas, "A review of computer vision–based structural health monitoring at local and global levels," *Structural Health Monitoring*, vol. 20, no. 2, pp. 692–743, 2020.

[6] S. Bhowmick, S. Nagarajaiah, and A. Veeraraghavan, "Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos," *Sensors*, vol. 20, no. 21, p. 6299, 2020.

[7] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics*, vol. 25, no. 3, pp. 507–515, 2011.

[8] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino, "Concrete crack detection by multiple sequential image filtering," *Computer-Aided Civil and Infrastructure Engineering*, vol. 27, no. 1, pp. 29–47, 2012.

[9] W. Wang, A. Zhang, K. C. P. Wang, A. F. Braham, and S. Qiu, "Pavement crack width measurement based on laplace's equation for continuity and unambiguity," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 2, pp. 110–123, 2018.

[10] D. Zhang, Q. Li, Y. Chen, M. Cao, L. He, and B. Zhang, "An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection," *Image and Vision Computing*, vol. 57, pp. 130–146, 2017.

[11] Z. Huang, Y. Tu, S. Meng, C. Sabau, C. Popescu, and G. Sas, "Experimental study on shear deformation of reinforced concrete beams using digital image correlation," *Engineering Structures*, vol. 181, pp. 670–698, 2019.

[12] M. Mahal, T. Blanksvärd, B. Täljsten, and G. Sas, "Using digital image correlation to evaluate fatigue behavior of strengthened reinforced concrete beams," *Engineering Structures*, vol. 105, pp. 277–288, 2015.

[13] Y. Pan, G. Zhang, and L. Zhang, "A spatial-channel hierarchical deep learning network for pixel-level automated crack detection," *Automation in Construction*, vol. 119, Article ID 103357, 2020.

[14] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proceedings of the 2016 IEEE International Conference on Image Processing*, pp. 3708–3712, (ICIP), Quebec, Canada, September 2016.

[15] F. C. Chen and M. R. Jahanshahi, "ARF-Crack: rotation invariant deep fully convolutional network for pixel-level crack detection," *Machine Vision and Applications*, vol. 31, no. 6, p. 47, 2020.

[16] F. C. Chen and M. R. Jahanshahi, "NB-CNN: deep learning-based crack detection using convolutional neural network and naïve bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, 2018.

[17] T. Ghosh Mondal, M. R. Jahanshahi, R. T. Wu, and Z. Y. Wu, "Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance," *Structural Control and Health Monitoring*, vol. 27, no. 4, Article ID e2507, 2020.

[18] W. Zhao, Y. Liu, J. Zhang, Y. Shao, and J. Shu, "Automatic pixel-level crack detection and evaluation of concrete structures using deep learning," *Struct Control Health Monit. n/a*, , Article ID e2981.

[19] A. C. Neves, I. González, R. Karoumi, and J. Leander, "The influence of frequency content on the performance of artificial neural network–based damage detection systems tested on numerical and experimental bridge data," *Structural Health Monitoring*, vol. 20, no. 3, pp. 1331–1347, 2021.

[20] O. B. Olalusi and P. Spyridis, "Machine learning-based models for the concrete breakout capacity prediction of single anchors in shear," *Advances in Engineering Software*, vol. 147, Article ID 102832, 2020.

[21] J. Zhang and J. Zhang, "An improved nondestructive semantic segmentation method for concrete dam surface crack images with high resolution," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5054740, 14 pages, 2020.

[22] H. Chen, Y. Su, and W. He, "Automatic crack segmentation using deep high-resolution representation learning," *Applied Optics*, vol. 60, no. 21, pp. 6080–6090, 2021.

[23] Y. Zhang, J. Fan, M. Zhang, Z. Shi, R. Liu, and B. Guo, "A recurrent adaptive network: balanced learning for road crack segmentation with high-resolution images," *Remote Sensing*, vol. 14, no. 14, p. 3275, 2022.

[24] J. Shu, W. Ding, J. Zhang, F. Lin, and Y. Duan, "Continual-learning-based framework for structural damage recognition," *Structural Control and Health Monitoring*, vol. 29, no. 11, Article ID e3093, 2022.

[25] J. Shu, C. Zhang, Y. Gao, and Y. Niu, "A multi-task learning-based automatic blind identification procedure for operational modal analysis," *Mechanical Systems and Signal Processing*, vol. 187, Article ID 109959, 2023.

[26] W. Ding, H. Yang, K. Yu, and J. Shu, "Crack detection and quantification for concrete structures using UAV and transformer," *Automation in Construction*, vol. 152, Article ID 104929, 2023.

[27] S. Li, X. Zhao, and G. Zhou, "Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 7, pp. 616–634, 2019.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 Ieee Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.

[29] Q. Mei, M. Gul, and M. R. Azim, "Densely connected deep neural network considering connectivity of pixels for automatic crack detection," *Automation in Construction*, vol. 110, Article ID 103018, 2020.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Springer International Publishing, Berlin, Germany, pp. 234–241, 2015.

[31] Z. Liu, Y. Cao, Y. Wang, and W. Wang, "Computer vision-based concrete crack detection using U-net fully convolutional networks," *Automation in Construction*, vol. 104, pp. 129–139, 2019.

[32] H. Bae, K. Jang, and Y. K. An, "Deep super resolution crack network (SrcNet) for improving computer vision-based automated crack detectability in in situ bridges," *Structural Health Monitoring*, vol. 20, no. 4, pp. 1428–1442, 2020.

[33] D. Kang, S. S. Benipal, D. L. Gopal, and Y. J. Cha, "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning," *Automation in Construction*, vol. 118, Article ID 103291, 2020.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[35] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," pp. 2961–2969, 2017, https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html.

[36] N. Wang, X. Zhao, Z. Zou, P. Zhao, and F. Qi, "Autonomous damage segmentation and measurement of glazed tiles in historic buildings via deep learning," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 3, pp. 277–291, 2020.

[37] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3614–3633, 2022.

[38] R. Girshick, "Fast R-CNN," pp. 1440–1448, 2015, https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html.

[39] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," 2020, http://arxiv.org/abs/1411.4734.

[40] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "PAD-net: multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing ArXiv180504409 Cs," 2021, http://arxiv.org/abs/1805.04409.

[41] J. H. Liew, S. Cohen, B. Price, L. Mai, and J. Feng, "Deep interactive thin object selection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 305–314, IEEE, Waikoloa, HI, USA, January 2021.

[42] Z. Chen, H. Zhou, J. Lai, L. Yang, and X. Xie, "Contour-Aware loss: boundary-aware learning for salient object segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 431–443, 2021.

[43] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," pp. 3994–4003, 2016, https://openaccess.thecvf.com/content_cvpr_2016/html/Misra_Cross-Stitch_Networks_for_CVPR_2016_paper.html.

[44] S. Liu, E. Johns, and A. J. Davison, "End-to-End multi-task learning with attention," 2021, http://arxiv.org/abs/1803.10704.

[45] S. Vandenhende, S. Georgoulis, and L. Van Gool, "MTI-net: multi-scale task interaction networks for multi-task learning," 2021, http://arxiv.org/abs/2001.06902.

[46] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-Aware feature propagation for scene segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6819–6829, November 2019, https://openaccess.thecvf.com/content_ICCV_2019/html/Ding_Boundary-Aware_Feature_Propagation_for_Scene_Segmentation_ICCV_2019_paper.html.

[47] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: improving semantic image segmentation with boundary detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 158–172, 2018.

[48] S. Liu, W. Ding, C. Liu, Y. Liu, Y. Wang, and H. Li, "ERN: edge loss reinforced semantic segmentation network for remote sensing images," *Remote Sensing*, vol. 10, no. 9, p. 1339, 2018.

[49] T. Yamaguchi and S. Hashimoto, "Automated crack detection for concrete surface image using percolation model and edge information," in *Proceedings of the IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, pp. 3355–3360, Paris, France, November 2006.

[50] S. Wang, Y. Pan, M. Chen, Y. Zhang, X. Wu, and S. F. W. Fcn, "Steel structure crack segmentation using a fully convolutional network and structured forests," *IEEE Access*, vol. 8, pp. 214358–214373, 2020.

[51] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[52] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the 2015 Ieee International Conference on Computer Vision*, pp. 1520–1528, Iccv, Santiago, Chile, June 2015.

[53] K. Sun, Y. Zhao, and B. Jiang, "High-resolution representations for labeling pixels and regions," 2021, http://arxiv.org/abs/1904.04514.

[54] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," 2021, http://arxiv.org/abs/1902.09212.

[55] Y. Hu, Y. Chen, X. Li, and J. Feng, "Dynamic feature fusion for semantic edge detection," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 782–788, Macao China, August 2019.

[56] M. Vardhana, N. Arunkumar, S. Lasrado, E. Abdulhay, and G. Ramirez-Gonzalez, "Convolutional neural network for biomedical image segmentation with hardware acceleration," *Cognitive Systems Research*, vol. 50, pp. 10–14, 2018.

[57] D. Sharifrazi, R. Alizadehsani, and M. Roshanzamir, "Fusion of convolution neural network, support vector machine and Sobel filter for accurate detection of COVID-19 patients using X-ray images," *Biomedical Signal Processing and Control*, vol. 68, Article ID 102622, 2021.

[58] L. Liebel and M. Körner, "Auxiliary tasks in multi-task learning," 2018, https://arxiv.org/abs/1805.06334.

[59] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," 2019, http://arxiv.org/abs/1906.03525.

[60] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[61] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade: Second Edition. Lecture Notes in Computer Science*, G. Montavon, G. B. Orr, and K. R. Müller, Eds., Springer, Berlin, Germany, pp. 437–478, 2012.

[62] L. N. Smith, "Cyclical learning rates for training neural networks," 2017, https://arxiv.org/abs/1506.01186.

[63] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2021, https://arxiv.org/abs/1412.6980v9.

[64] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: a nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 3–11, Springer, 2018.

[65] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *ArXiv180202611 Cs*, http://arxiv.org/abs/1802.02611, 2021.

[66] Y. Liu, J. Yao, X. Lu, and R. Xie, "A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, 2019.