Hindawi
Structural Control and Health Monitoring
Volume 2024, Article ID 4863177, 21 pages
https://doi.org/10.1155/2024/4863177

Research Article

Weakly Supervised Fatigue Crack Detection in Steel Bridge Girders Using a Proposed Two-Stage Network Training with a Segmentation Refinement Module

Fei Jiang,[1] Youliang Ding,[1] Yongsheng Song,[2] Fangfang Geng,[3] and Zhiwen Wang[4]

[1]*Key Laboratory of Concrete and Prestressed Concrete Structures of Ministry of Education, Southeast University, Nanjing 210096, China*
[2]*School of Architecture Engineering, Jinling Institute of Technology, Nanjing 211169, China*
[3]*School of Architecture Engineering, Nanjing Institute of Technology, Nanjing 211167, China*
[4]*Shenzhen Express Engineering Consulting Co. Ltd, Shenzhen 518000, China*

Correspondence should be addressed to Youliang Ding; civilchina@hotmail.com and Fangfang Geng; fangfang_civil@hotmail.com

Received 22 August 2023; Revised 24 October 2023; Accepted 22 December 2023; Published 6 January 2024

Academic Editor: Yong Xia

Existing semantic segmentation methods for fatigue cracks in steel bridge girders are fully supervised and thus demand manual annotation of pixel-level labels, which is time-consuming. Recently, there have been remarkable developments in semantic segmentation under image-level tag supervision. However, these weakly supervised approaches are still inferior to the fully supervised manner in terms of accuracy. To mitigate this gap, this paper commits to improving the correlation between high-level semantics to low-level appearance. A two-stage training manner with a segmentation refinement module for progressively refining pseudolabels and training the segmentation network was proposed. First, an activation modulation and recalibration scheme was recommended, which leverages a spotlight branch and a compensation branch to locate both the discriminative and less-discriminative object regions. Then, the generated pseudolabels were used as supervision to train the segmentation network in the proposed two-stage manner. In the first stage, the network was pretrained to learn all essential information and provide a basic segmentation performance, aiming to facilitate network convergence in the following training. To develop the inference quality, in the second stage, the pretrained network was further trained recursively with the designed segmentation refinement module to improve the labels using two postprocessing algorithms between each iteration. Overall, our method achieves comparable inference results to fully supervised approaches while significantly reducing annotation workload, which improves the efficiency of routine bridge inspection.

1. Introduction

Steel box girders have been widely applied in long-span cable-stayed and suspension bridges in light of their advantages of low weight, high torsional stiffness, and rapid construction. Such structures are suffering from fatigue cracks owing to initial defects and residual stresses related to fabrication and construction processes. Under repeated vehicle loads, the development of fatigue cracks continues to reduce the stiffness and integrity of the local welded connections, thereby decreasing the reliability and durability of bridges. To support accurate decision-making on bridge maintenance, it is essential to detect and monitor the fatigue cracks in a periodical or even real-time manner.

Nondestructive testing (NDT) techniques, such as traditional ultrasonic testing or advanced phased-array ultrasonic testing [1], are usually used to approach the problem of local damage detection, which can obtain both the inner and surface damage characteristics. Nevertheless, the accuracy of these NDT-based methods is limited by measurement noise and highly depends on skilled inspectors or expensive instruments. Compared with NDT, the vision-based methods are relatively inexpensive to implement and adaptive for surface cracks [2]. Currently, human-based visual inspection

still plays a crucial role in the routine fatigue maintenance of steel bridges. However, its consistency in quantitative evaluation and accessibility cannot be guaranteed considering environmental and human factors [3, 4]. Furthermore, the inspection by human inspectors is often labor-intensive and time-consuming.

With the rapid development of computer technology, methods for crack inspection based on computer vision have emerged. Most of these studies have focused on image processing techniques (IPTs). A significant advantage of IPTs is that almost all surface defects may be identified [3]. However, this method is limited by its reliance on subjectively chosen parameters and filters, which indicates a lack of generality under different scenarios [5]. Therefore, some researchers tried to improve the robustness of the IPT-based method in real-world situations by machine learning (ML) [6–9]. However, these improvements are strictly limited by the feature extraction capacity of IPTs, and despite the improvements, these optimized methods still require some pre- and postprocessing techniques that are time-consuming [10, 11].

Owing to the development of convolutional neural network (CNN), deep learning (DL)-based methods have been proposed for image-based crack detection in computer vision. Recent studies on DL-based crack detection have mainly involved methods of image classification, object detection, and semantic segmentation. Image classification methods focus on obtaining image-level class information, while object detection methods concentrate on getting the class and general location information. Neither of the above methods can provide sufficiently accurate information in terms of crack width, length, and direction, which, however, play fundamental roles in fatigue maintenance design. Consequently, semantic segmentation methods based on supervised learning which could provide pixel-level semantic and localization information have been used for fatigue crack detection.

However, the utilization of supervised semantic segmentation demands a large number of annotated pixel-level labels [12], which calls for enormous human labor and time costs during the preparation process [13, 14]. To approach this difficulty, weakly supervised learning (WSL) with image-level labels, which indicates the existence of the object of interest, has been implemented in the training of segmentation networks [14, 15]. Compared with pixel-level annotation, the labeling cost at the image level can be significantly reduced. Nevertheless, the network performance by WSL is lower than that by fully supervised learning (FSL), and there remains room for performance improvement.

This paper proposed an improved WSL-based method for high performance of fatigue crack detection with low labeling cost. The main contributions of the proposed method are as follows:

(i) A pixel-level detection method was proposed for the segmentation of fatigue cracks, which only used image-level classification labels but achieved state-of-the-art performance in WSL-based methods.

(ii) To realize customized optimization for the segmentation characteristics, the activation modulation and recalibration (AMR) scheme was adopted to generate refined pseudolabels, which approached the problem that only the most discriminative regions were highlighted in state-of-the-art WSL-based methods.

(iii) To the best of the author's knowledge, this paper is the first to propose a two-stage training method for weakly supervised fatigue crack segmentation. After learning the semantic features in refined pseudolabels, the segmentation network was trained recursively with a segmentation refinement module. This takes both the advantages of deep learning and the morphological knowledge of cracks.

The remainder of this paper is organized as follows. Section 2 introduces the current literature on crack detection. Section 3 outlines the methodology of our proposed method. Section 4 presents the experimental results. Lastly, conclusions are given in Section 5.

2. Related Works

2.1. Conventional IPT-Based Crack Detection Methods. Conventional IPT-based methods have been widely utilized for crack detection over the past two decades. Early studies rely on intensity-thresholding methods due to their simplicity and efficiency, assuming that crack pixels exhibit lower intensity than the background [16–18]. However, these methods face challenges in unevenly illuminated images, as a single threshold is applied to the entire image.

Edge detection, leveraging edge-like and texture features associated with cracks, has also been a common technique [4, 19–23]. Despite its popularity, edge-detection methods often yield disjoint crack fragments instead of complete profiles [24]. Researchers have attempted to address this limitation through fragment-linking techniques [25–28]. Texture-analysis methods, such as those employing local binary patterns, have been applied to detect textured cracks, like in pavement crack detection [29–31]. However, these methods may struggle with cracks exhibiting intensity inhomogeneity.

To overcome challenges related to real-world image variations, IPT-based methods have been enhanced by integrating several ML classifiers such as support vector machine (SVM) and k-nearest neighbor (KNN) algorithms [6, 7, 32]. Nevertheless, these improvements are constrained by the feature extraction capacity of IPTs. Notably, prior IPT-based studies predominantly focused on cleaner surfaces, such as pavement or concrete, and may not be robust enough for crack detection in steel bridge girders, where obstacles such as marker curves and weld line edges exist.

2.2. Deep Learning-Based Crack Detection Methods. With the development of high-performance graphics processing units (GPUs) and parallel computing, DL-based techniques are

gaining prominence in computer vision-based surface damage detection. CNNs do not require manual construction of features or prior knowledge of crack shape, texture, or contextual information. DL-based crack detection methods have been widely used in civil engineering and are categorized into two main types: patch-level and pixel-level methods [2, 33].

Patch-level methods typically employ sliding window techniques or crop small patches for crack detection. A CNN-based workflow utilizing sliding windows allows the detection of cracks in images larger than those used for training [3]. Faster region-based CNN (faster R-CNN) has been proposed for real-time detection of cracks [10], and transfer learning from a benchmark CNN enhances robust crack classification with limited crack images [11]. While these methods provide satisfactory crack region detection, they may lack morphology information related to cracks. Postprocessing algorithms, such as edge detection and dilation operations, are employed to segment detected patches at the pixel level [34, 35]. However, postprocessing accuracy depends on the optimal patch size, which can be challenging to determine.

Semantic segmentation has emerged as an effective way to detect cracks at the pixel level, offering quantification properties. Fully convolutional network (FCN) retains spatial information, enabling pixel-level segmentation. Various end-to-end segmentation models, including FCN, VGG16, InceptionV3, ResNet152, feature pyramid networks (FPN), and U-net, have been applied for crack detection, achieving high accuracy [36–48]. Recent innovations include FCS-Net, a deep FCN-based network integrating ResNet50, atrous spatial pyramid pooling (ASPP), and batch normalization (BN) for fine fatigue crack segmentation [48].

While these approaches have advanced automated crack detection, patch-level methods are quick but limited in extracting crack information. Pixel-level methods provide accurate segmentation but are labor and time-intensive for pixel-level label preparation. Addressing this trade-off, an efficient crack detection method with reduced annotation burden needs to be proposed.

2.3. Image-Level Weakly Supervised Semantic Segmentation (WSSS). Since the generation of fully annotated datasets is laborious, alternative learning methods based on unlabeled or weakly labeled visual data have become prevalent in recent years. Various forms of weak labels have been proposed in previous studies, such as bounding boxes, points, scribbles, and image-level supervision. Among these, image-level weak supervision is favorable for its simplicity and reliability [49]. Therefore, this study focuses on the image-level weakly supervised crack segmentation.

Recently, image-level WSSS works mostly employ class activation maps (CAMs) [50] as initial pseudolabels for the training of segmentation networks. CAMs provide a heatmap representation that highlights the interested object regions. However, CAMs generated by classification networks tend to highlight the most discriminative regions; hence, the obtained pseudolabels may only cover a part of the target objects. These coarse discriminative object regions may not meet the requirement of pixel-level semantic segmentation and thus harm the network performance. The efforts to alleviate this issue can be classified into two aspects: refining the pseudolabels based on CAMs and modifying the segmentation training procedures.

To obtain finer initial pseudolabels, most studies have focused on refining the seeds or response regions of initial CAMs. Wei et al. [51] used dilated convolution with different dilate rates to enlarge the receptive fields. Kolesnikov and Lampert [52] proposed three principles: seed, expand, and constrain (SEC) to refine the seeds. Ahn and Kwak [53] predicted semantic affinity between pixels by AffinityNet and propagated local activations using a random walk algorithm. Chang et al. [54] enforced the network to learn better response regions by exploiting the subcategory information. Lee et al. [55] randomly selected the hidden units in the feature map to make the activated regions better characterize the object. However, these methods were developed in an interactive and random manner, which may lose essential information. To approach this issue, Qin et al. [56] recently proposed a novel activation modulation and recalibration (AMR) scheme, which leverages a spotlight branch and a compensation branch to provide complementary and task-oriented CAMs for WSSS.

In addition to refining the initial pseudolabels, several studies have focused on improving the WSSS performance by modifying the segmentation training procedures. Most of them trained segmentation networks in a recursive manner along with a refinement module that exploits the prior information at its best. Khoreva et al. [57] proposed a recursive training enhanced by denoise techniques that improved the labels between each round with object priors. Li et al. [58] designed a new superpixel conditional random field (superpixel-CRF) model to refine generated masks, based on which the segmentation model was trained iteratively. Recently, for WSSS of power lines, Choi et al. [59] introduced a broken line connection algorithm to provide refined segmentation labels for recursive segmentation training.

To the authors' knowledge, no studies have been conducted on WSSS of fatigue cracks in a noisy background, let alone the corresponding improvement methods for WSSS performance. This paper is also based on the concept of CAM and utilizes a recursive training procedure. A critical difference from previous studies is our proposed two-stage training procedure, which first trains the segmentation network based on refined pseudolabels generated by AMR and then performs recursive training with a designed refinement module to denoise the crack segmentation. Therefore, the proposed method can benefit from both the refinement of the initial pseudolabels and the modification of the segmentation training procedure.

3. Methods

3.1. Overview of the Proposed Method. Single classification networks can localize the most discriminative object regions, which is however far from the requirement for pixel-level segmentation. To provide refined pseudolabels, the AMR

method is recommended to integrate less-discriminative regions. In addition, to further enhance the performance of WSSS, a two-stage training procedure is proposed, aiming to continuously refine training labels by leveraging the features of deep learning and fatigue crack morphologies.

As shown in Figure 1, our framework consists of two parts: the first part deals with the training of AMR branches and the generation of initial pseudolabels; the second part involves the proposed weakly supervised two-stage training of the segmentation network. Specifically, the first part involves a systematic four-step process. First, the input images (X_{in}) undergo preprocessing, wherein they are cropped into small patches, and each patch is annotated with four image-level labels: background (L_b), crack (L_c), marker (L_m), and the combination of crack and marker (L_{cm}). Subsequently, AMR is trained using these annotated patches, enabling it to effectively highlight crack and marker regions of interest. The trained AMR is then utilized to generate CAMs of patches for new input images. Importantly, the images used for AMR training are distinct from those used for generating the CAMs, avoiding overestimating the trained AMR performance. Finally, the dense conditional random field (DenseCRF) [60] is employed to process the probability maps derived from CAMs, producing fine segmentation masks as initial pseudolabels for the following training of the segmentation network.

In the second part, the generated pseudolabels are used to train the segmentation network within certain epochs to obtain fine-enough basic segmentation performance in stage I. After that, the pretrained network is further trained in a recursive manner in stage II, and in each iteration, a designed refinement module refines the predicted masks from the prior iteration and gets more complete and precise labels to train the segmentation network in the current iteration.

3.2. Activation Modulation and Recalibration Method. A conventional CAM of a specific category highlights the discriminative regions used by multilabel classification networks to determine that category. Given an input image $I \in R^{3\times H\times W}$, global average pooling (GPA) is used to identify the importance of the feature maps $F(I) \in R^{C\times H\times W}$ (C is the channel of the feature maps) extracted from the last convolution layer. Then, the conventional CAM can be simply obtained by computing a weighted sum matrix of the extracted feature maps:

$$M(I) = w_N^T F(I), \tag{1}$$

where $M(I) \in R^{N\times H\times W}$ is the obtained CAMs and w_N^T is the weight of the fully connected layer for N classes.

However, the conventional CAMs are classification-oriented and lack some minor but essential features for the segmentation tasks. To solve this problem, Qin et al. [56] proposed a novel AMR scheme for WSSS, which outperformed state-of-the-art WSL-based methods on the PASCAL VOC 2012 dataset. However, its effectiveness for highlighting crack regions under the interference of edge-like features has not been validated. As shown in Figure 2, similar to previous studies, a spotlight branch based on common CNNs is utilized to highlight the most discriminative object regions and generate the corresponding spotlight CAMs M_s. Besides that, the main contribution of the AMR method is the implementation of a parallel compensation branch, which leverages a spatial-channel attention module to focus on those essential but easily overlooked regions. The obtained compensation CAMs M_{c} are used to recalibrate the spotlight CAMs M_s to generate the final weighted CAMs M_{w} [56]:

$$M_{\text{w}}(I) = \xi M_{\text{s}}(I) + (1-\xi) M_{\text{c}}(I), \tag{2}$$

where ξ is the recalibration coefficient.

During the training process, the AMR method is optimized with a two-part combination loss L_{all}, which can be expressed as

$$L_{\text{all}} = L_{\text{cls}} + L_{\text{cps}}. \tag{3}$$

The first loss prat, L_{cls}, is the averaged classification loss of the two branches and can be calculated as

$$L_{\text{cls}} = \frac{1}{2}\left(L_{\text{cls}}^s + L_{\text{cls}}^c\right), \tag{4}$$

where L_{cls}^s and L_{cls}^c are the multilabel soft margin losses for supervision on the spotlight branch and the compensation branch, respectively.

The second loss part, L_{cps}, aims to provide a cross pseudosupervision on the spotlight branch and the compensation branch. It can be regarded as the semantic similarity regularization of each branch and can be represented as

$$L_{\text{cps}} = \left\| M_s - M_c \right\|_1. \tag{5}$$

This paper aims to use the AMR method to generate high-quality pseudolabels from image-level annotations. The whole generation process of the pseudolabels can be summarized as follows. ResNet50 is used as the backbone to design the multilabel classification branches of AMR, and the spatial-channel attention module with the Gaussian function is plugged into the compensation branch. The process then utilizes the image-level annotations to train the AMR under the supervision of equation (3). After the training is completed, the weighted CAMs can be obtained by using the discriminative localization technique described in equation (2). Finally, DenseCRF is used to process the CAM probability map to obtain the synthetic labels used to train the segmentation network.

3.3. Two-Stage U-Net Training. U-net [61] was originally designed for semantic segmentation of biomedical images with some edge-like features, which make it much more straightforward for crack detection. The adopted skip-connection promotes the aggregation of spatial and semantic information, which makes U-net outperform the conventional FCN. Furthermore, U-net also performs well while little training data are available. All the advantages mentioned above make U-net a good fit for the detection of

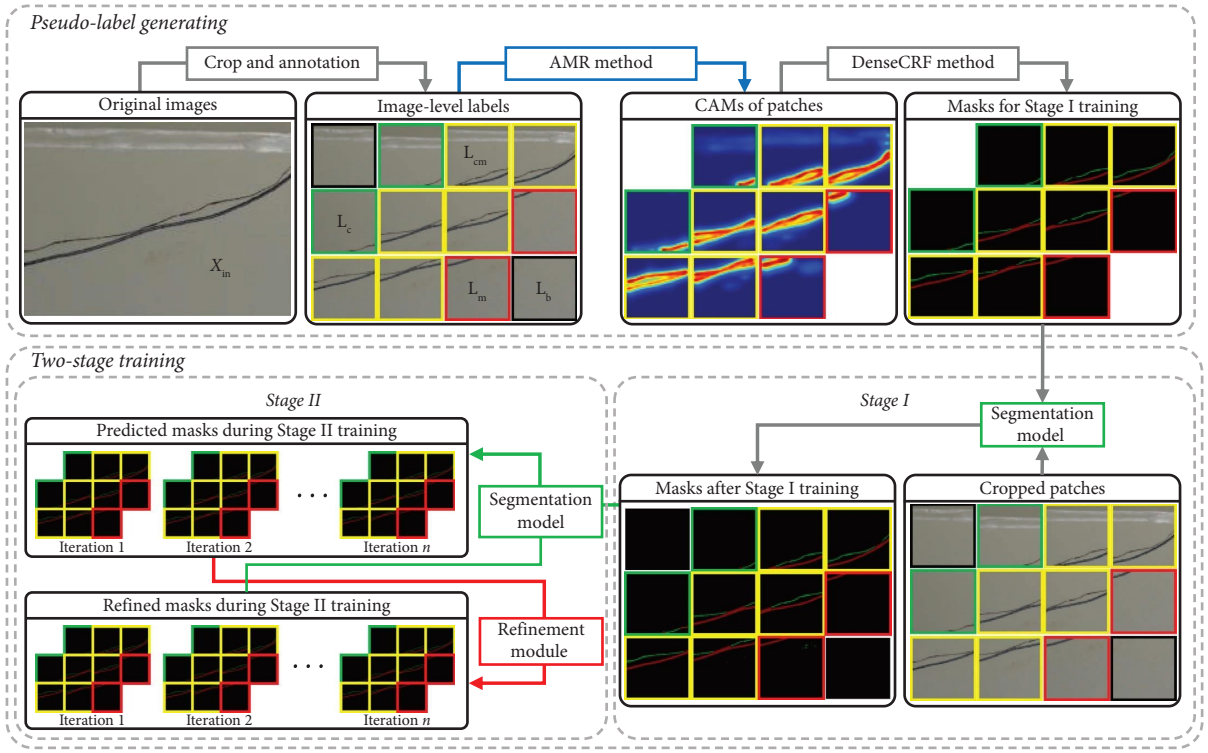

Figure 1: Overview of the proposed method.

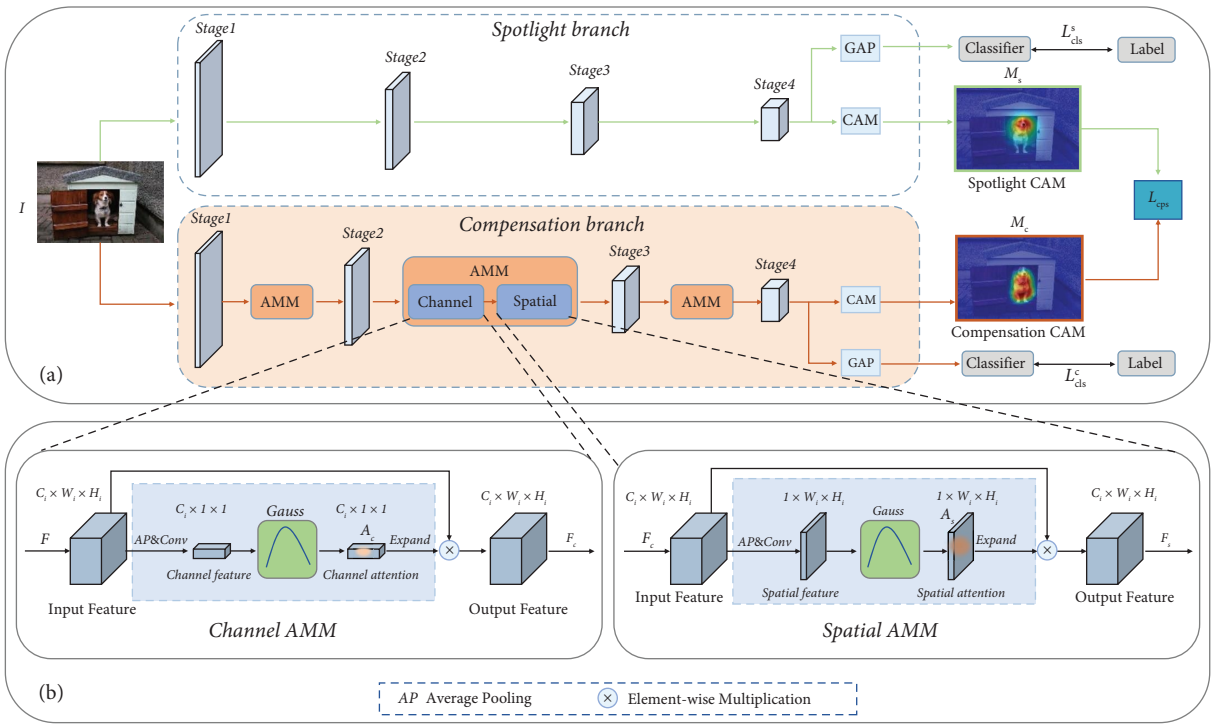

Figure 2: The framework of the AMR method [56]. (a) AMR. (b) AMM.

fatigue cracks. In this study, the original U-net architecture was slightly modified to meet the input and output image size requirements. The detailed operations for each layer are listed in Table 1, where Conv represents the convolutional operation; BN indicates the batch normalization operation; ReLU is the ReLU activation function; Maxpooling

Table 1: Detailed operations for each layer in U-net.

Layers		Feature size	Operation	Filter size	Filter number	Stride	Padding
Encoder	Input	$3 \times 512 \times 512$	Conv1 + BN + ReLU	$3 \times 3 \times 3$	64	1	1
	L1	$64 \times 512 \times 512$	Conv2 + BN + ReLU	$64 \times 3 \times 3$	64	1	1
	L2	$64 \times 512 \times 512$	Maxpooling	2×2	—	2	—
	L3	$64 \times 256 \times 256$	Conv3 + BN + ReLU	$64 \times 3 \times 3$	128	1	1
	L4	$128 \times 256 \times 256$	Conv4 + BN + ReLU	$128 \times 3 \times 3$	128	1	1
	L5	$128 \times 256 \times 256$	Maxpooling	2×2	—	2	—
	L6	$128 \times 128 \times 128$	Conv5 + BN + ReLU	$128 \times 3 \times 3$	256	1	1
	L7	$256 \times 128 \times 128$	Conv6 + BN + ReLU	$256 \times 3 \times 3$	256	1	1
	L8	$256 \times 128 \times 128$	Maxpooling	2×2	—	2	—
	L9	$256 \times 64 \times 64$	Conv7 + BN + ReLU	$256 \times 3 \times 3$	512	1	1
	L10	$512 \times 64 \times 64$	Conv8 + BN + ReLU	$512 \times 3 \times 3$	512	1	1
	L11	$512 \times 64 \times 64$	Maxpooling	2×2	—	2	—
	L12	$512 \times 32 \times 32$	Conv9 + BN + ReLU	$512 \times 3 \times 3$	1024	1	1
	L13	$1024 \times 32 \times 32$	Conv10 + BN + ReLU	$1024 \times 3 \times 3$	1024	1	1
Decoder	L14	$1024 \times 32 \times 32$	TransConv1 + concate L11	$1024 \times 2 \times 2$	512	2	—
	L15	$1024 \times 64 \times 64$	Conv11 + BN + ReLU	$1024 \times 3 \times 3$	512	1	1
	L16	$512 \times 64 \times 64$	Conv12 + BN + ReLU	$512 \times 3 \times 3$	512	1	1
	L17	$512 \times 64 \times 64$	TransConv2 + concate L8	$512 \times 2 \times 2$	256	2	—
	L18	$512 \times 128 \times 128$	Conv13 + BN + ReLU	$512 \times 3 \times 3$	256	1	1
	L19	$256 \times 128 \times 128$	Conv14 + BN + ReLU	$256 \times 3 \times 3$	256	1	1
	L20	$256 \times 128 \times 128$	TransConv3 + concate L5	$256 \times 2 \times 2$	128	2	—
	L21	$256 \times 256 \times 256$	Conv15 + BN + ReLU	$256 \times 3 \times 3$	128	1	1
	L22	$128 \times 256 \times 256$	Conv16 + BN + ReLU	$128 \times 3 \times 3$	128	1	1
	L23	$128 \times 256 \times 256$	TransConv4 + concate L2	$128 \times 2 \times 2$	64	2	—
	L24	$128 \times 512 \times 512$	Conv17 + BN + ReLU	$128 \times 3 \times 3$	64	1	1
	L25	$64 \times 512 \times 512$	Conv18 + BN + ReLU	$64 \times 3 \times 3$	64	1	1
	L26	$64 \times 512 \times 512$	Conv19 + softmax	$64 \times 1 \times 1$	3	1	0
	Output	$3 \times 512 \times 512$					

represents the max pooling operation; TransConv represents the deconvolution operation; Softmax is the Softmax activation function; Concat means the concatenation of the encoder and decoder layers by skip connection.

To accelerate the network convergence and further improve the performance of WSSS, the segmentation network is proposed to be trained in a two-stage manner, which can be summarized as follows. In the first stage, U-net is pretrained for certain epochs to learn all the essential information indicated by the initial pseudolabels. This training stage aims to provide a basic segmentation performance and facilitate network convergence in the following training process. Although the AMR method is used, the initial pseudolabels are still incomplete since they are generated by the network only using image-level labels. To develop the inference quality, the pretrained U-net is further trained in a recursive manner in the second stage. It is expected that the segmentation performance with noisy labels could be developed by itself via recursive training with a segmentation refinement module.

3.4. Segmentation Refinement Module: Assimilation and Connection. In the proposed method, a segmentation refinement module is designed to provide continuously optimized labels in the recursive training of U-net. The refinement module aims to exploit the available morphology information related to fatigue cracks and the surrounding markers at their best. The information is integrated in the following two cues:

> C1. Cracks and markers are generally separated. Therefore, there are no discrete marker segments on the crack path, and likewise, there are no discrete crack segments on the marker path.
>
> C2. Fatigue cracks mostly initiate near the substrate surface, and in the propagation phase, the crack penetrates the substrate surface, forming a continuous damage path. Therefore, surface fatigue cracks are usually continuous.

The recursive training is enhanced by denoising the network outputs using the morphology information. Following the above two cues, the labels can be improved by two postprocessing algorithms between each iteration.

A1. An assimilation algorithm follows cue C1 to assimilate the false-detected discrete segments into their categories. Algorithm A1 supports the proposed assimilation process, as illustrated in Figure 3. The input img is a synthetic mask containing segmented crack and marker domains. When U-net is used for pixel classification, given the similar features between cracks and markers, they could be misclassified into each other, which results in the intermingling of cracks and markers. Statistical analysis of the

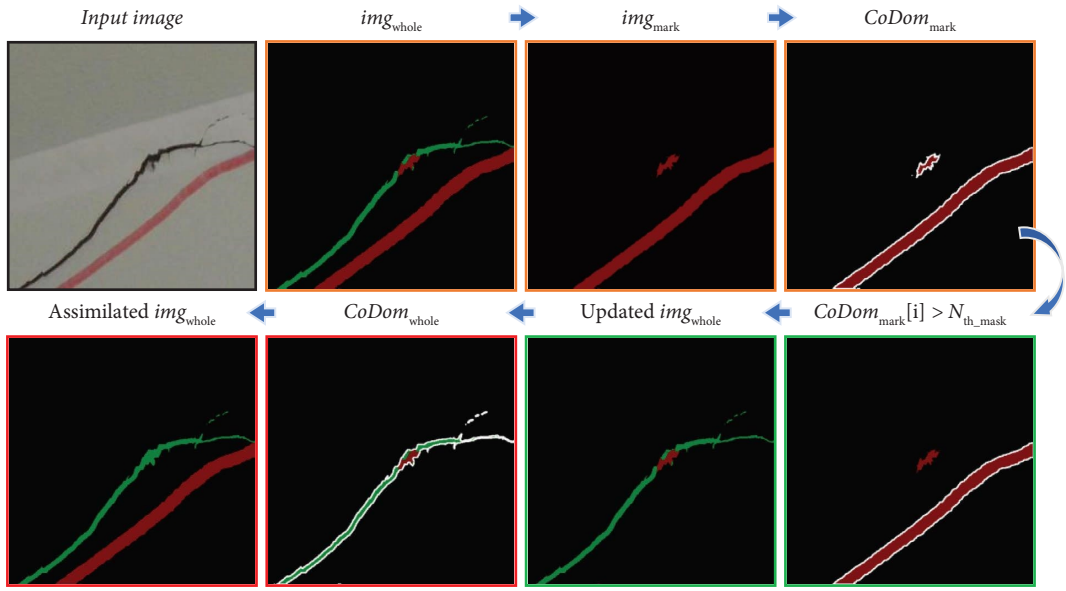

Figure 3: Example images of the proposed assimilation process. For illustration purpose, only the crack assimilation process is shown here.

predicted results reveals that the number of pixels being false-detected as another category in a connected domain is generally smaller than that being correctly detected. Based on this, crack assimilation and marker assimilation are designed to correct the misidentified crack and marker pixels, respectively. Each assimilation part consists of three steps. For crack assimilation, first, marker pixels img_{mark} are stripped from the synthetic label copy $\text{img}_{\text{whole}}$ and connected domains of markers $\text{CoDom}_{\text{mark}}$ can be obtained. Second, the marker pixels in each connected domain are counted and further removed from $\text{img}_{\text{whole}}$ if the pixel number exceeds the threshold $N_{\text{th\_mask}}$. The reason for the removal procedure is that those larger connected domains are less likely to be misidentified, and the removal of them facilitates the following assimilation process. Third, connected domains of the updated synthetic label copy $\text{CoDom}_{\text{whole}}$ are obtained, and each obtained domain is checked to assimilate the misidentified crack pixels into its category by comparing the pixel numbers. Similar steps are implemented for mask assimilation as well.

A2. A connection algorithm follows cue C2 to connect discrete crack segments into a whole. Given the influence of uneven illumination inside dim bridge girders, some background noise could be wrongly identified as small crack points during the segmentation process. Before the crack connection, these misidentified noises are filtered beforehand according to the highlighted regions by CAMs. Algorithm A2 supports the connection process, as shown in Figure 4. The proposed crack connection part consists of three steps. First, crack segments are extracted from the assimilated mask $\text{img}_{\text{whole}}$, and the corresponding connected domains $\text{CoDom}_{\text{crack}}$ are further found. Subsequently, the extreme points ExtrmPts_i are found for each domain, which is a list containing top-most, bottom-most, right-most, and left-most points (Figure 4). After that, the Eulerian distance between every extreme point of a connected domain and that of every other contour is compared to obtain two endpoints (Pt_1, Pt_2) with the least distance. Finally, the two endpoints are connected using an assumed crack line in $\text{img}_{\text{whole}}$. During the recursive training process, the discrete crack segments are gradually connected into a whole.

4. Experiment

4.1. Dataset and Experimental Setup

4.1.1. Dataset. The original dataset employed in this paper was granted by the organizing committee of the 1st International Project Competition for Structural Health Monitoring (IPC-SHM 2020) [62]. Specifically, a total of 200 images with the size of 4,928 × 3,264 or 5,152 × 3,864 pixels were provided, and these images were collected from steel bridge girders under different camera parameters and environment conditions during routine inspection. The dataset acquisition details can be found at https://www.schm.org.cn/#/IPC-SHM,2020/project1.

Based on the original dataset, two subdatasets were further generated to train and evaluate the proposed method. The first subdataset, called the AMR dataset, aims to train the AMR branches for generating high-quality pseudolabels. Thus, AMR-dataset is a multilabel image classification dataset. 80 high-resolution images were selected from the original dataset and resized to multipliers of 512. These resized images were further cropped into small patches of 512 × 512 pixels. The cropping process was performed to improve the training and testing efficiency and has been widely adopted in previous studies. Considering the category-imbalance problem, the final generated AMR-dataset contains 800 images with cracks, 800 images with markers, and 800 background images, which all have manually annotated image-level labels.

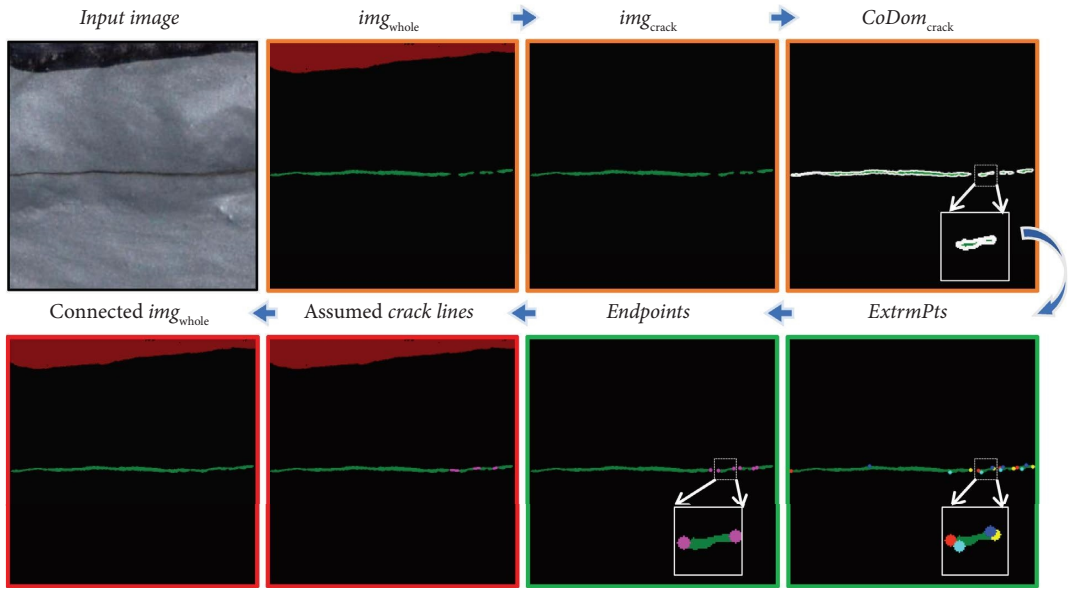

Figure 4: Example images of the proposed connection process.

The remaining 120 images in the original dataset were randomly divided into three subsets: the training set, the validation set, and the testing set using a split ratio of 6 : 2 : 2. Therefore, the second subdataset, called the U-net-dataset, was constructed for training and evaluating the segmentation network. During training, the original high-resolution images were resized and cropped into small patches of 512×512 pixels, and the trained AMR was used to produce their corresponding synthetic labels. Note that there is no element overlap between the AMR-dataset and the U-net-dataset, thus avoiding overestimating the AMR performance.

4.1.2. Evaluation Metrics. In evaluating the segmentation performance of the proposed method, three key metrics are employed. Annotation time measures the labeling efficiency before model training, offering insights into the pretraining annotation workload and cost-effectiveness. After model training, the model's prediction accuracy is assessed using the mean Intersection-over-Union (mIoU) metric, a standard measure of segmentation performance. Furthermore, the novel efficiency metric (Images/s) quantifies the prediction speed of the trained model, reflecting the number of images processed per second. This comprehensive set of metrics provides a thorough evaluation, addressing annotation costs, segmentation accuracy, and computational efficiency, offering a well-rounded perspective on the effectiveness of the proposed segmentation approach.

4.1.3. Training Configuration. For producing attention maps, the AMR classification branches were trained for 20 epochs with a batch size of 16 images and an initial learning rate of 0.001. A stochastic gradient descent algorithm was leveraged for network optimization using a 0.0001 weight decay. Some data augmentations were also implemented on the training samples to improve the training efficiency. After obtaining the pseudolabels, the U-net segmentation network was trained in a two-stage manner, which first pretrained the network for 50 epochs using the initial pseudolabels and then further developed the segmentation performance using a recursive network training for 50 iterations. The initial learning rate was 0.0001, and the weighted cross-entropy loss was used to approach the unbalanced data. All the tasks described were performed on a workstation (CPU: double Intel® Xeon® CPU E5-2680 v4 @ 2.40 GHz, RAM: 64 GB, GPU: ASUS GeForce RTX 2060 D6 12G).

4.2. Study of the Annotation Workload. An experiment was designed to contrast the annotation workload of the proposed weakly supervised method with that of traditional fully supervised methods. In this experiment, our method employed image-level annotation, which simply involved placing different categories of images into different category folders. For the fully supervised method, three main representative pixel-level annotation tools were selected to scrutinize the annotation workload: Adobe® Photoshop (PS), LabelMe [63], and an online annotation tool EasyDL [64].

As illustrated in Figure 5, PS utilizes the magic wand and lasso tools, especially effective for objects in sharp contrast to the background. LabelMe, an open-source Python tool, extracts annotations with control points along object boundaries. EasyDL, an algorithm-assisted tool, requires users to add or remove anchor points, automatically identifying object and background regions for pixel-level labels.

Twenty images from the dataset were selected for the annotation workload experiment. Each experiment was repeated three times under consistent conditions by individuals with varying proficiency levels. The recorded

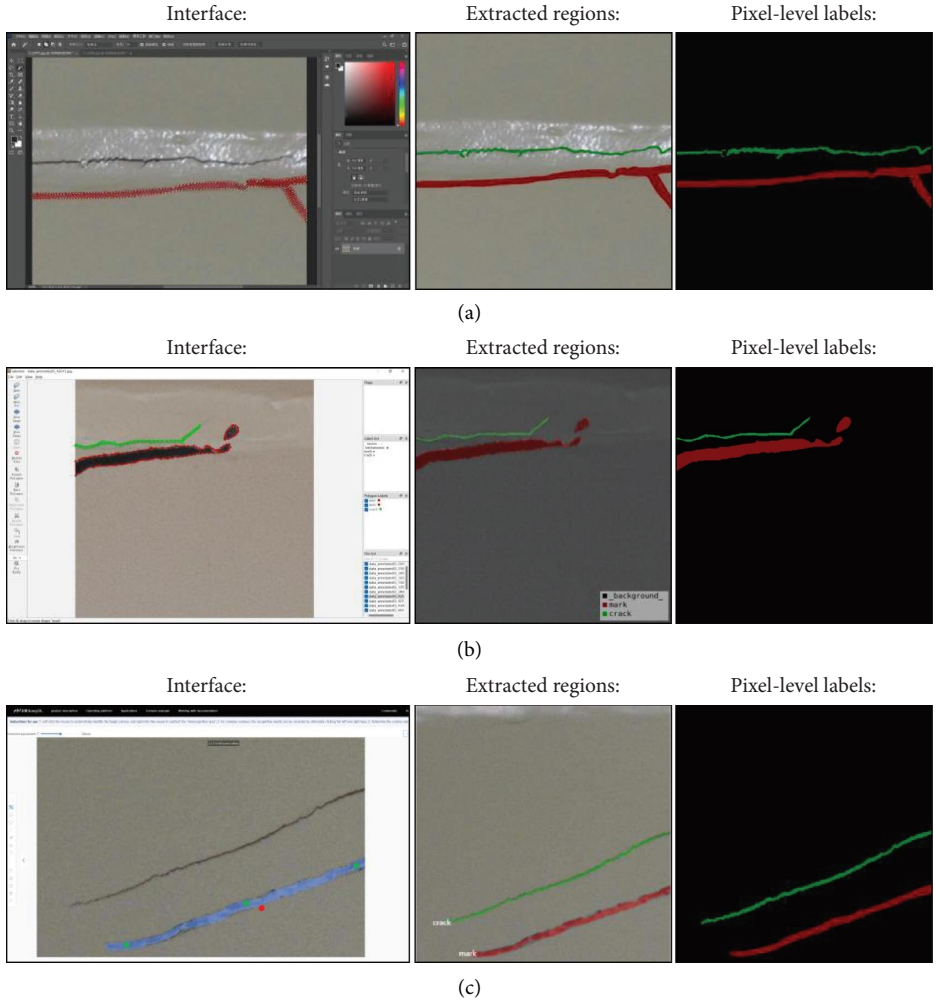

Figure 5: Interface and labeling process of three different pixel-level annotation tools: (a) PS, (b) LabelMe, and (c) EasyDL.

annotation time characterized the overall workload, and the results for different annotation methods are summarized in Figure 6.

Figure 6 reveals that LabelMe exhibits the highest annotation time, attributed to the complexity of placing boundary control points. In contrast, PS, utilizing the magic wand tool, proves quicker than LabelMe. Among pixel-level tools, EasyDL records the lowest annotation time and deviation. However, EasyDL's pixel-level annotation time is approximately seven times longer than annotating image-level labels. These results suggest the significantly reduced annotation workload of our weakly supervised method compared to conventional fully supervised methods.

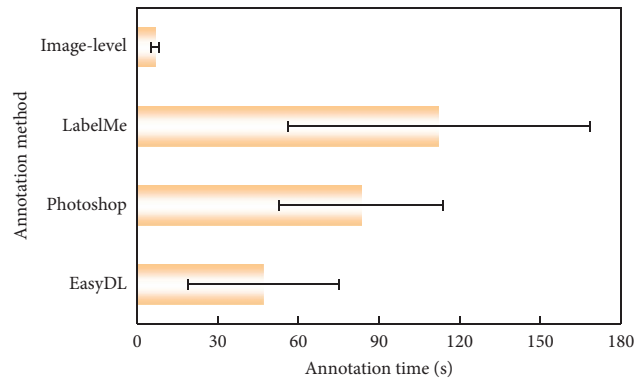

Figure 6: Comparison of the annotation time using different methods.

4.3. Comparison between AMR and Grad-CAM for CAM Generation. The paper proposes using AMR in Section 3.2 to generate effective CAMs for providing semantic and localization cues in segmentation. To assess the AMR's effectiveness in activating complete object regions, a comparison was conducted with the state-of-the-art method, gradient-weighted Class Activation Mapping (grad-CAM) [65]. Grad-CAM used gradient information to assign weights to feature maps extracted from the last convolutional layer of ResNet50 in this study. This process allowed the creation of CAMs without the need for retraining, preserving the existing model structure and parameters. A set of thresholds $th = \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ was defined to convert CAMs into synthetic labels. For semantic

class k, if the value of CAM $A_k^{x,y} \geq \text{th}_n$, the class of these pixels at the spatial location (x, y) was annotated as k, otherwise background.

Table 2 shows the comparison of IoU metrics on the training subset of the built U-net-dataset for different th values. "Crack," "Marker," and "Background" denote IoUs corresponding to these three different classes, and "All" denotes the mean values of IoUs for the three classes, namely, mIoU. The results of the grad-CAM were reproduced using their publicly available implementation, and all the results in Table 2 were obtained without DenseCRF postprocessing. The results indicate that the adopted AMR produces more accurate CAMs than grad-CAM for all different thresholds. The performance of our proposed CAMs is generally improved by about 2%, and when th = {0.5, 0.6}, "All" achieves the best results. Figure 7 shows typical examples of the CAM results obtained using AMR and grad-CAM, and it can be observed that the former provides more complete activation regions than the latter. These results demonstrate that the implementation of AMR in WSSS of fatigue cracks is promising, but given the limited training data, the authors recommend further validation when more data are available.

4.4. Ablation Study. The configuration of the proposed method contains three main components: the AMR-based CAM generation, a segmentation refinement module, and the proposed two-stage training. To investigate the effectiveness of each component, ablation studies were conducted and the corresponding results are listed in Table 3. The mIoU metric during training was compared for each experiment configuration, as shown in Figure 8. For configurations C1, C3, and C4, they have the same mIoU trend in the first 50 epochs and overlap each other before 50 in Figure 8. Some example images according to the two-stage training phase are illustrated in Figure 9, where ground truth refers to the true labels for the input images in this study.

In the ablation studies, direct segmentation training based on the initial pseudolabels (Configuration C1) was adopted as the baseline. As shown in Table 3 and Figure 9, the initial pseudolabels are very coarse and the performance of the baseline was 71.9%. By applying the two-stage training with the segmentation refinement module, the segmentation performance improves gradually during the recursive training (50~100 iterations in Figure 8) and finally achieves 76.5%. This demonstrates that our method is effective.

Experiments were extensively conducted to verify the effectiveness of the proposed two-stage training by comparing the performance of configurations C2 and C4. For the direct training method, it started from improving initial pseudolabels using the segmentation refinement module and then used these refined labels to train the segmentation network for 100 epochs. Figure 8 shows the comparison results. With the iterations, the performance only increases at the beginning and stabilizes at a low mIoU, while in the proposed two-stage training, the performance continues to increase at the second stage and reaches a much higher mIoU. This result demonstrates that our two-stage training method is effective. This training scheme progressively mines common object features from previous masks and then expends more reliable object regions with the assistance of the segmentation refinement module; thus, the performance can increase rapidly to a quite satisfactory result.

In some cases, the initial pseudolabels produced by AMR are still incomplete. To mine the whole regions of objects, a segmentation refinement module is incorporated into the recursive training. To evaluate the effectiveness, an experiment was conducted on the training framework without refinement (configuration C3) and the performance was compared with that of configuration C4. From Figure 8, without the refinement module, some misidentified object regions may grow gradually, and thus the performance decreases continuously during the recursive training. Figure 9 shows how the prediction improves with iterations according to the refinement module. By exploiting the available morphology information, some false-detected segments are assimilated into their correct categories (Figures 9(b) and 9(d)), and the broken cracks are gradually connected into a whole (Figures 9(a), 9(c) and 9(e)). These results demonstrate the effectiveness of our proposed segmentation refinement module.

The performance differences among configurations C1 to C4 are intricately linked to the recursive training process and the role of the segmentation refinement module. Throughout recursive training, the model's predictions from the previous iteration serve as training labels for the next iteration. In the absence of segmentation refinement, as observed in C3 (Figure 8), errors accumulated during iterative training can amplify, causing a gradual decline in performance. On the contrary, configuration C4, leveraging segmentation refinement, undergoes an iterative correction process. This module guides the model through training cycles, progressively rectifying errors and enhancing performance. This iterative refinement proves pivotal in steering the model toward improved predictions, countering the cumulative degradation seen in C3. However, in configuration C2, where the segmentation refinement module optimizes initial pseudolabels only once at the beginning of training, its impact is more restrained. This limited optimization, compared to the continuous refinement in C4, restricts the model's exposure to refined and contextually rich labels, resulting in a more modest improvement, as shown in Table 3.

4.5. Comparison of the Trained Segmentation Performance. The proposed method was further compared with fully supervised methods and some other weakly-supervised methods. The fully supervised learning was directly conducted on the manually labeled ground-truth data of the pixel level. The other weakly supervised segmentation followed the workflow that first generated pseudolabels and then used the synthetic labels to directly train the segmentation network. Other than the proposed U-net, several popular architectures were chosen as alternative segmentation networks to provide a more comprehensive comparison. All of the models were trained to converge using the

Table 2: Comparison of the accuracy in terms of IoU (%) for CAMs generated by AMR and grad-CAM on the training subset of the built U-net dataset.

Threshold	AMR				Grad-CAM			
	All	Crack	Marker	Background	All	Crack	Marker	Background
0.2	41.07	10.09	36.77	76.36	39.29	9.62	34.30	73.94
0.3	47.42	13.97	44.13	84.16	45.81	13.22	41.73	82.47
0.4	53.30	18.13	52.52	89.26	52.12	17.53	50.47	88.34
0.5	58.32	22.53	59.71	92.72	57.65	22.41	58.24	92.31
0.6	58.70	25.34	56.71	94.06	57.16	25.31	52.50	93.66
0.7	53.51	26.19	40.32	93.86	51.41	24.98	35.73	93.51
0.8	45.85	23.28	20.20	93.17	44.65	21.66	19.16	93.12

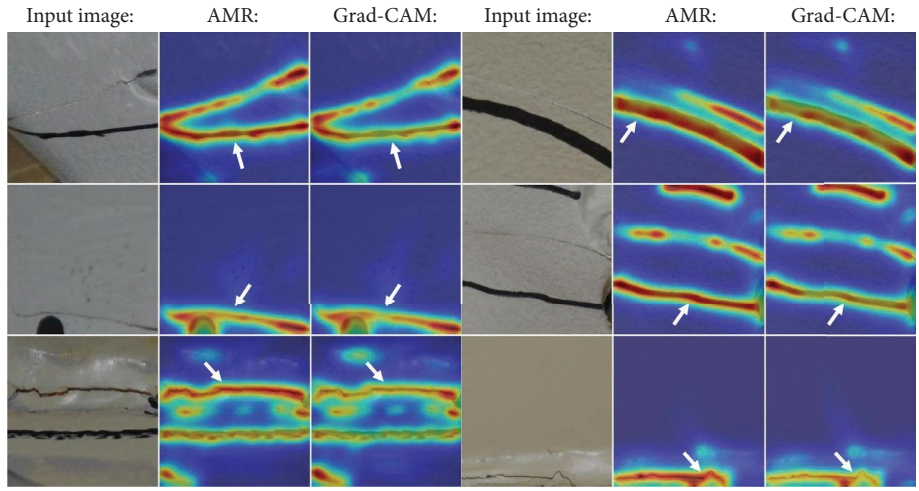

Figure 7: Examples of CAMs generated by AMR and grad-CAM methods on the training subset of the built U-net dataset.

Table 3: Results of the ablation study.

Configuration	Pseudolabel	Direct training	Segmentation refinement	Two-stage training	mIoU on validation set (%)
C1	✓	✓			71.9
C2	✓	✓	✓		72.3
C3	✓			✓	51.4
C4	✓		✓	✓	76.5

same training parameters, and the trained models were evaluated according to the test set. The evaluation metrics are listed in Table 4.

As shown in Table 4, the weakly supervised methods are obviously lower than the fully supervised methods in terms of mIoU, which is attributed to the incompleteness of the initial pseudolabels. Compared with the FCN-based methods, U-net achieves better prediction performance. This is due to the fact that U-net has a finer upsampling process with more channels, and rather than being simply added as in FCN, the same-level encoder and decoder parts are concatenated in U-net. Although soft attention is implemented at the skip connections in attention U-net (AttU-net), the mIoU metrics of AttU-net and U-net are very close under both fully supervised and weakly supervised learning configurations. For the current test samples, our method with U-net achieves a higher mIoU value than the other weakly supervised methods and is only slightly lower than the U-net-based fully supervised method by 1.6%. Besides, the higher efficiency value obtained by our method indicates its capability to handle more images in a given time frame.

A comparison of segmentation results for typical damage images is shown in Figure 10. In Figures 10(a) and 10(b), image patches with different background colors under normal conditions are used to test the trained models. All models provide good prediction results for them. However, the FCN-based prediction is relatively rough, which may be owing to the lack of enough localization information during the upsampling process. Figures 10(c) and 10(d) show prediction results for very thin cracks, and our method makes better predictions than the other weakly supervised methods whose inferences of fatigue cracks are incomplete. To evaluate our model's robustness to surface interference,

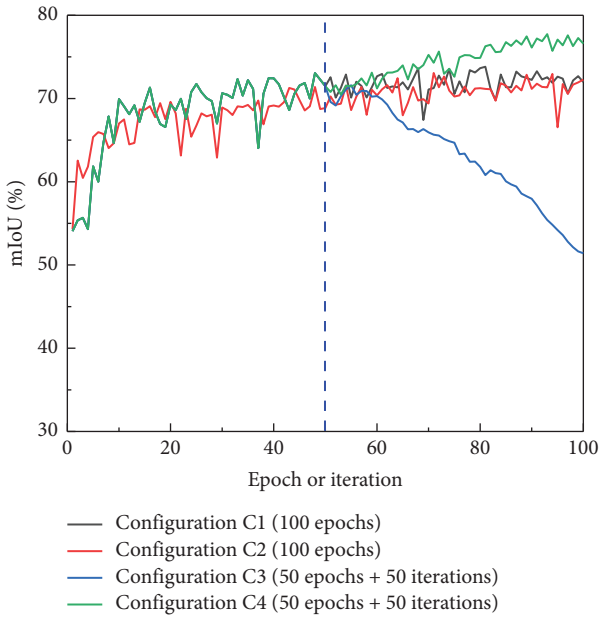

FIGURE 8: Change of mIoU on the validation set during the training process. Configurations C1, C3, and C4 have the same mIoU trend in the first 50 epochs and thus overlap each other before 50.

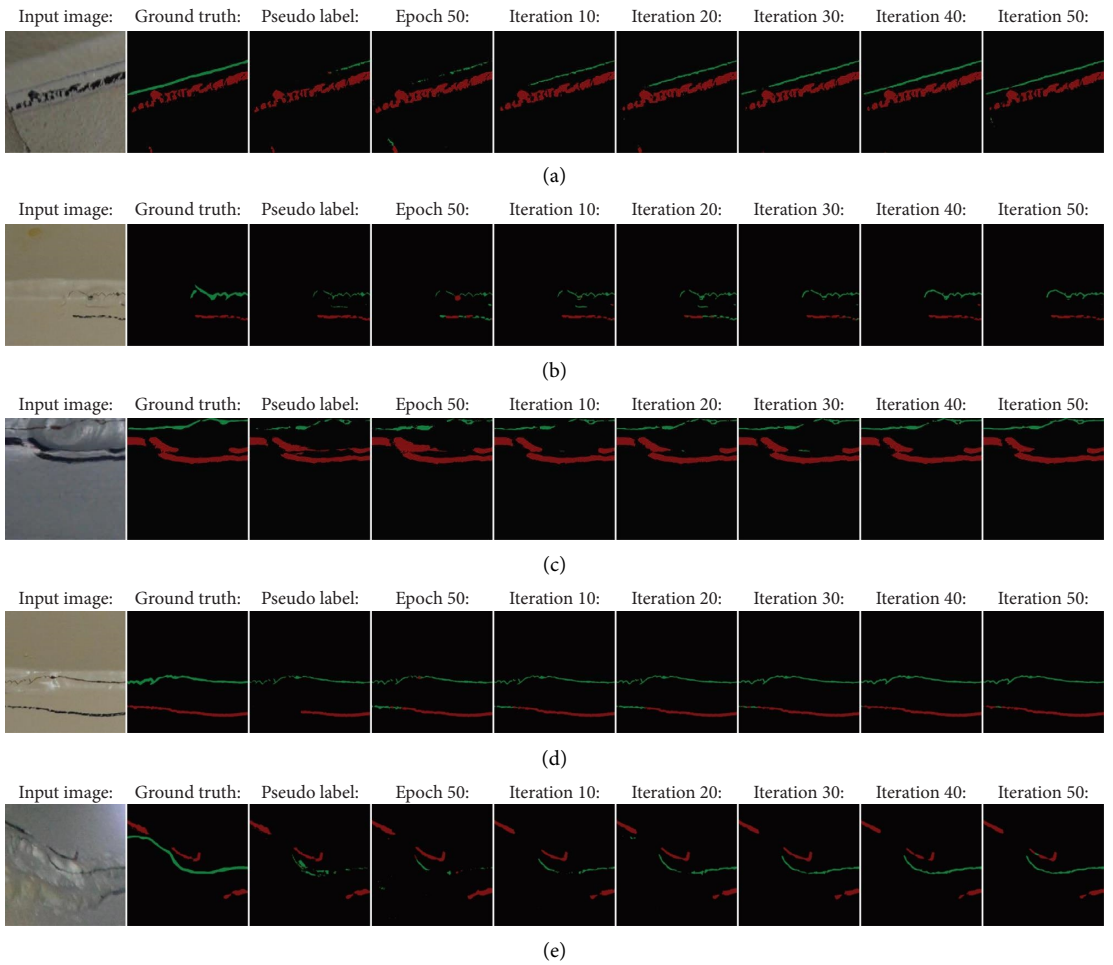

FIGURE 9: Example segmentation results during the two-stage training process with the proposed refinement module.

Table 4: Comparison of the evaluation metrics on the test set.

Supervision	Method	mIoU on test set (%)	Efficiency (images/s)
Full	FCN-8s	74.9	12
	U-net	77.9	20
	AttU-net	77.0	13
Weak	FCN-8s	68.2	12
	U-net	71.1	20
	AttU-net	71.9	13
	Our method	76.3	20

AttU-net means attention U-net.

the image patch with cracks on a contaminated surface is fed into the trained models, as shown in Figure 10(e). Our proposed model is able to discriminate stains with minor errors, but its prediction of fatigue cracks is not as accurate as that of the fully supervised methods. As shown in Figure 10(f), the crack-like weld edges are correctly identified as background by all models, and among the weakly supervised methods, the proposed method provides more satisfactory results. However, the models fail in some cases with confusing construction lines and tiny markers, as illustrated in Figure 10(g).

Overall, the promising nature of our proposed method is demonstrated by the higher mIoU results compared to other weakly supervised methods, and the accurate segmentation outcomes achieved for typical damage image patches, showcasing its potential for effective crack detection.

4.6. Assessment of Model Performance under Complex Real-Bridge Conditions. Section 4.5 shows the promising performance of our proposed method. However, only cropped image patches with sizes of 512×512 pixels are visually illustrated in Figure 10. In this section, the trained model performance is further visualized using original images with the size of $4{,}928 \times 3{,}264$ or $5{,}152 \times 3{,}864$ pixels. These larger images are deemed to better capture and reflect the complexity inherent in real-bridge environments, offering a comprehensive assessment of the trained model's strengths and limitations. Four typical real-bridge conditions are considered as follows.

4.6.1. Ideal Inspection Conditions. Under ideal inspection conditions, where the crack background is clean and free of distractions, and the crack markers are neatly applied, the model's performance is evaluated. In Figure 11, the segmentation results demonstrate the accurate detection of cracks with varying widths. Our method also successfully extracts most of the crack markers; only very few pen strokes are identified as cracks, as indicated by the dashed-line frames. This can be attributed to the resemblance of pen strokes to cracks in terms of color and shape. Overall, these findings provide a baseline understanding of the model's accuracy and segmentation quality under optimal conditions.

4.6.2. Varying Lighting Conditions. The results presented in Figure 12 reveal that our proposed method successfully detects and extracts the position and morphological information of cracks under varying lighting conditions. However, the dynamic nature of lighting introduces slight errors in the model's performance, as indicated by the dashed-line frames. For instance, in Figure 9(a), the smooth top plate-fillet weld appears brighter than the surrounding base material due to reflections. Along the boundaries where the weld intersects with areas of varying brightness, crack-like features are occasionally formed, leading to misclassifications by the algorithm. Similarly, in Figure 9(c), the presence of shadows creates a contrast with the bright background, resulting in several pixels along the boundaries being unfortunately identified as cracks. However, in Figure 9(d), where dim lighting conditions are present, the model successfully avoids misclassifications near shadow boundaries, attributing to the lack of strong contrasts and intensity-gradient changes. Moreover, in Figure 12(b), the proposed method effectively identifies most of the crack and marker regions even under dim lighting conditions.

4.6.3. Cluttered Backgrounds. The prediction performance of our method is further assessed under the challenging condition of cluttered backgrounds. In Figure 13, genuine cracks and markers are accurately identified. Figure 13(a) shows the detection of some background pixels as cracks at primer color transition areas, attributed to visual complexities caused by gradients and borders between different primer colors. In Figure 13(b), occasional false detections of cracks occur due to complex thin and light markings that visually resemble cracks. Figure 13(c) demonstrates instances where some dot-like stains are sometimes misclassified as markers due to their visual similarity. Finally, in Figure 13(d), the needle-like stains are identified as cracks as expected, given their elongated and thin characteristics resembling crack-like features so much. To mitigate the above minor errors, the authors recommend calculating the area of connected components in the predicted mask and removing some false positives corresponding to smaller connected component areas, such as the smaller dot-like stains. Despite these challenges, our model performance remains satisfactory overall.

4.6.4. Obstacles or Confusions Caused by Irrelevant Objects. Figure 14 showcases the model's performance when faced with obstacles or confusions caused by irrelevant objects. While genuine cracks and markers are accurately detected, there are some prediction errors. In Figures 14(a) and 14(b),

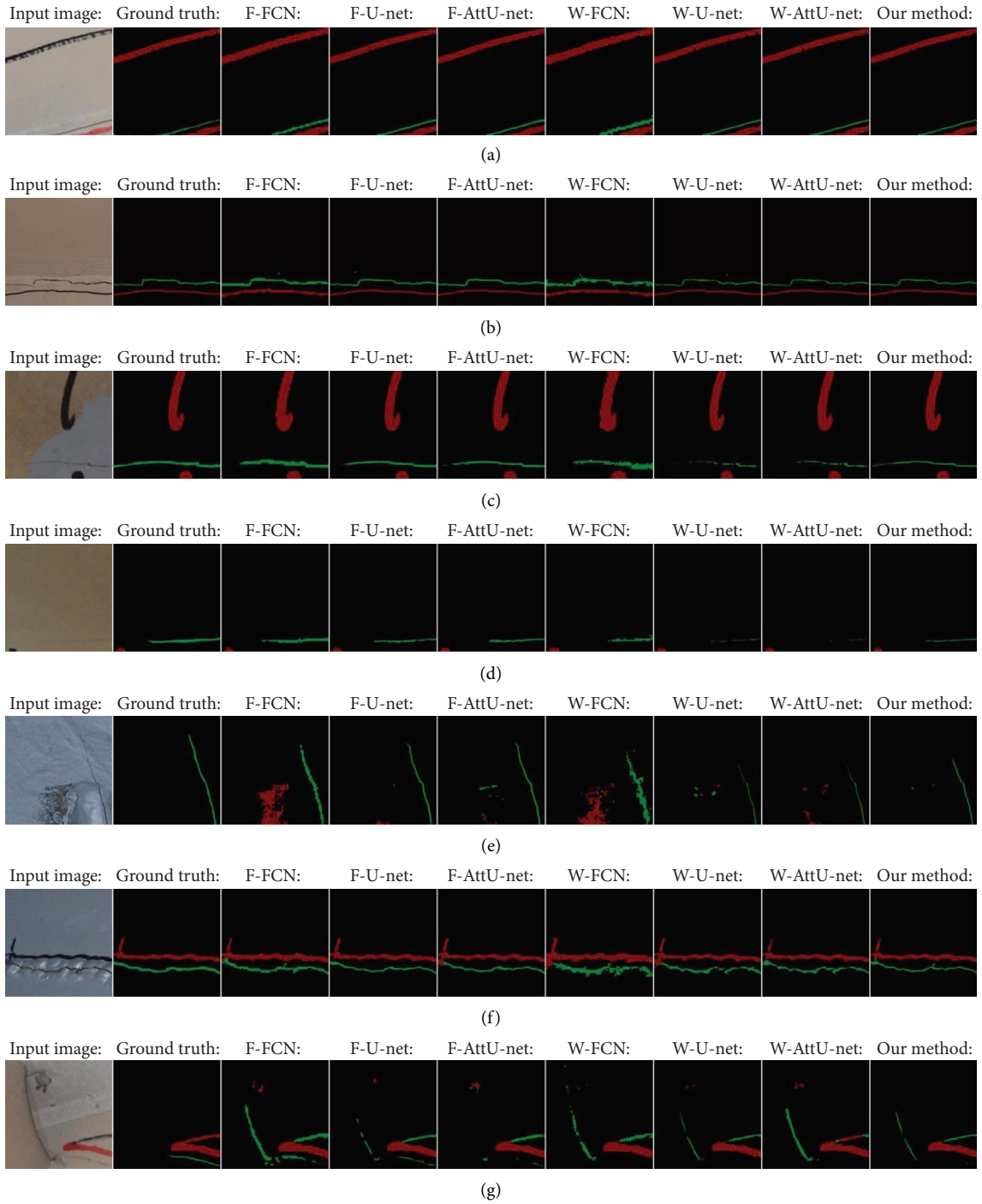

FIGURE 10: Example segmentation results generated by different networks under full or weak supervision: (a, b) normal condition, (c, d) tiny crack, (e) contaminated surface, (f) weld line edges, and (g) misidentified example. F and W represent fully supervised and weakly supervised, respectively.

curved hole edges serve as obstacles, and the prediction results differ due to varying primer colors. The strong intensity-gradient change between the black curved hole edge and the light background primer color in Figure 14(a) makes it more susceptible to being erroneously detected as cracks compared to Figure 14(b) with a darker primer color. In Figures 14(c) and 14(d), rare scenarios involving irrelevant objects, such as transparent tapes, sensor boxes, and electronic wires, reveal instances where these edges are sometimes mistaken as cracks or markers. These misclassifications arise due to the visual similarity between these objects and genuine cracks/markers, the presence of texture resembling cracks/markers, and the limited exposure of the model to such scenarios during training (only Figures 14(c)

FIGURE 11: Original-image segmentation results for idea inspection conditions: (a) subtle crack, (b) fine crack, (c) medium crack, and (d) coarse crack.

and 14(d) have transparent tapes and sensor boxes in the dataset), resulting in a lack of contextual information for accurate differentiation.

Given that this study aims to enhance the accuracy of traditional WSL-based crack segmentation and reduce the annotation burden of FSL-based methods, the proposed method successfully achieves this goal, as evidenced by comparable mIoU and visualized segmentation results, along with reduced annotation time compared to previous studies [13, 33, 42, 48, 66]. Overall, the proposed method

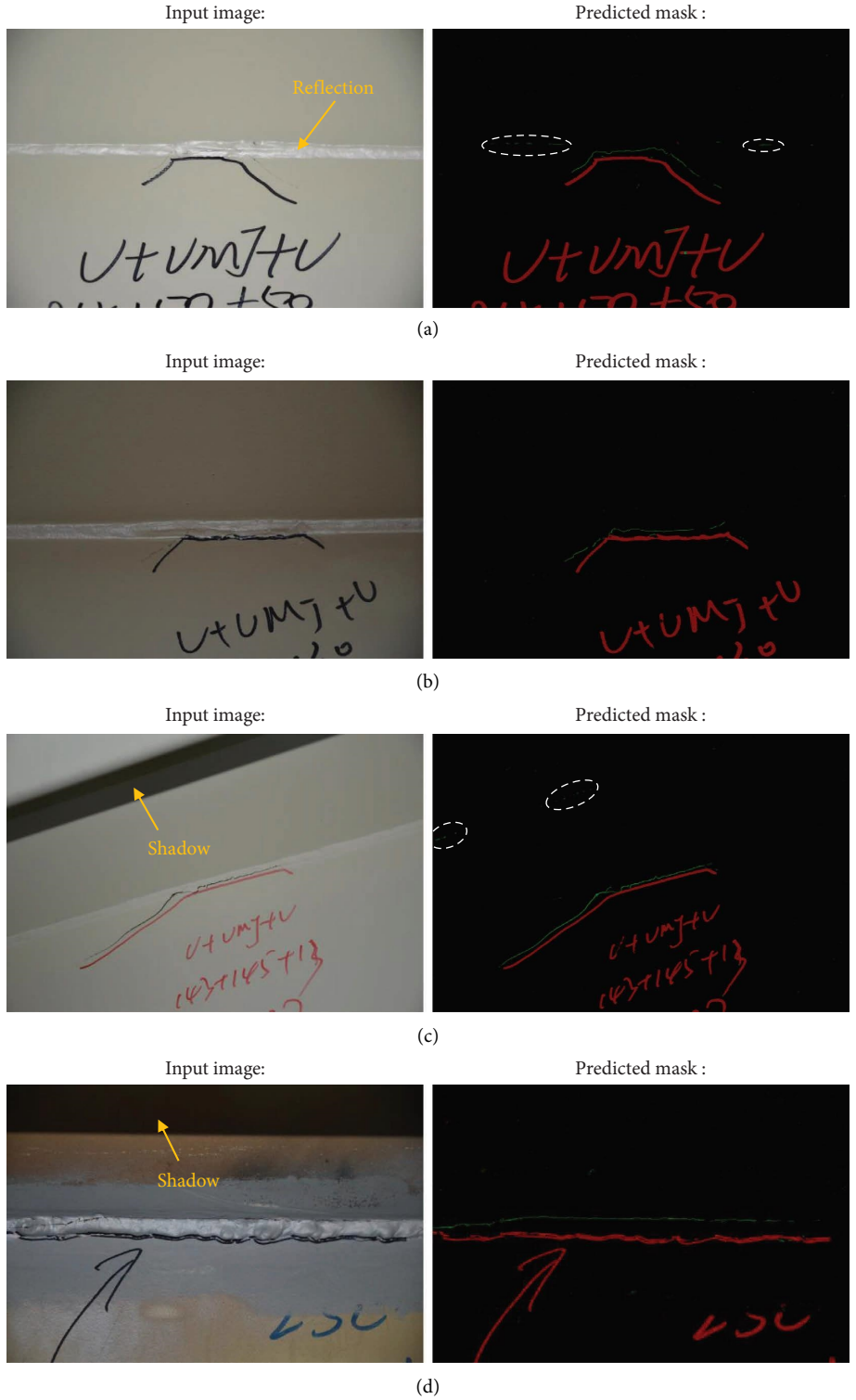

Figure 12: Original-image segmentation results under varying lighting conditions: (a) reflection scenario, (b) dim environment, (c) shadow under bright condition, and (d) shadow under dim condition.

demonstrates satisfactory overall performance under ideal or complex real-bridge conditions. The presence of minor misclassifications and errors, particularly in challenging scenarios involving cluttered backgrounds and obstacles or confusions of irrelevant objects, can be mitigated through the augmentation of training data and the inclusion of more diverse scenarios, which would enhance the model's ability to differentiate genuine cracks from confounding factors.

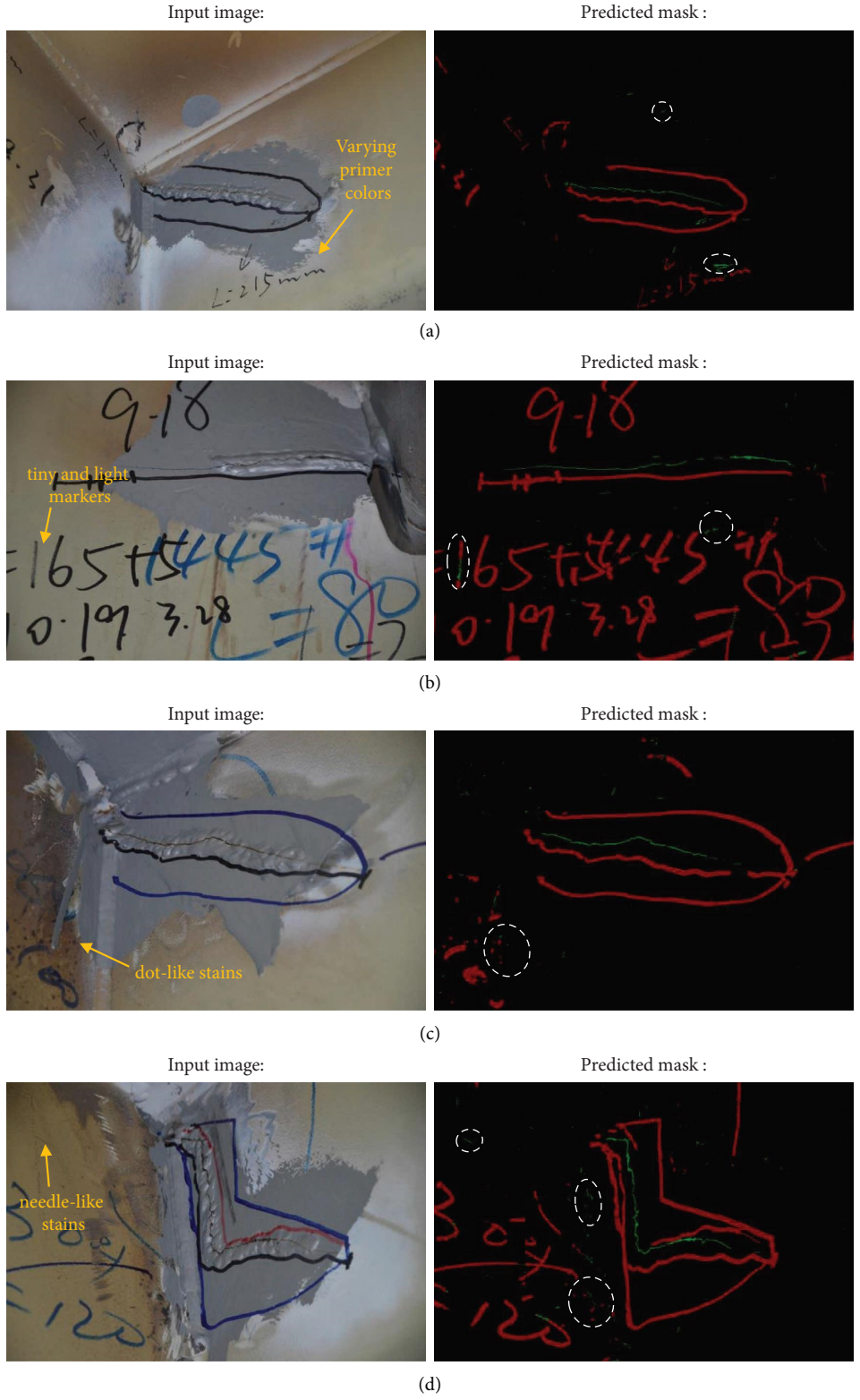

FIGURE 13: Original-image segmentation results under the condition of cluttered backgrounds: (a) varying primer colors, (b) confusing markings, (c) dot-like stains, and (d) needle-like stains.

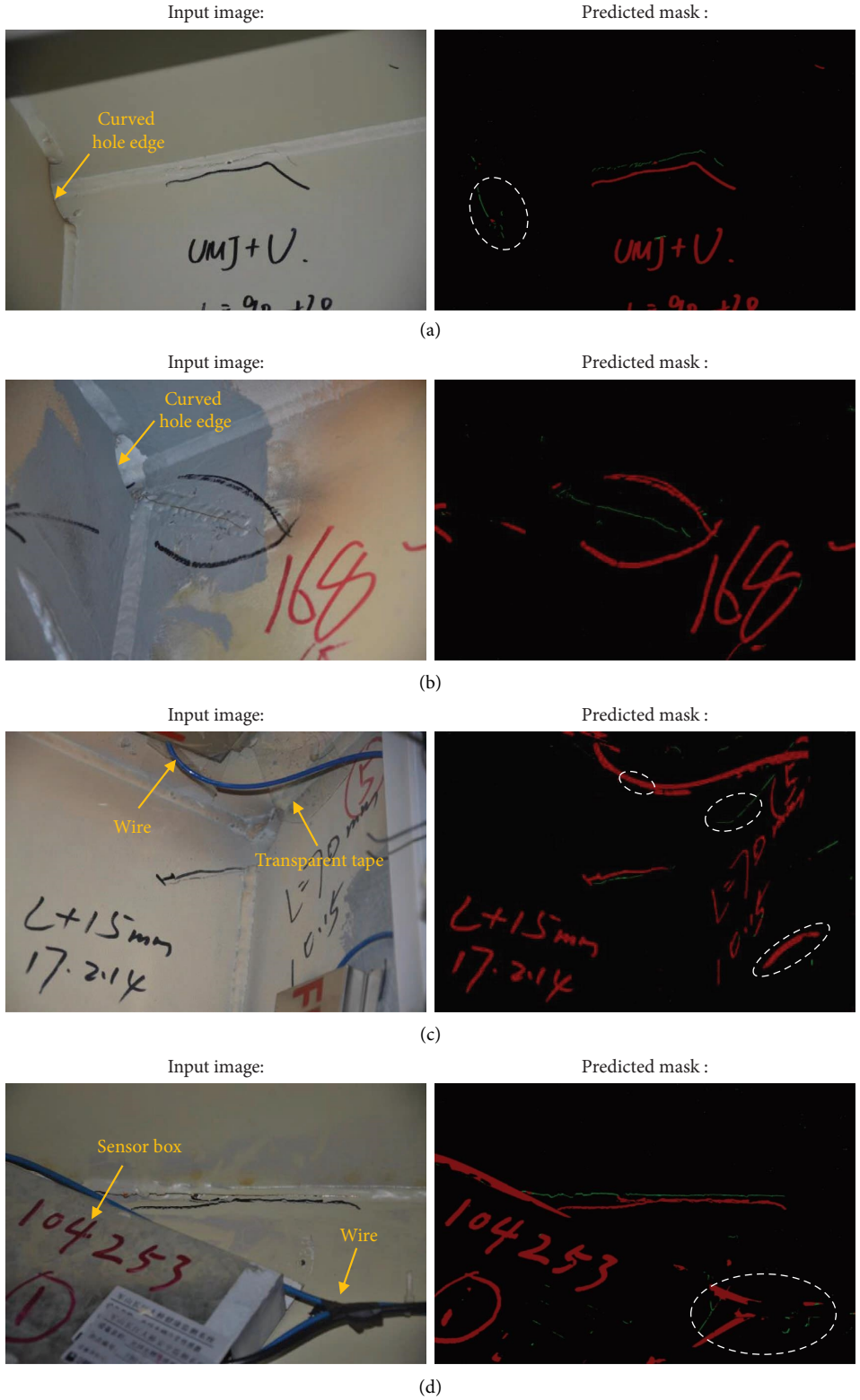

FIGURE 14: Original-image segmentation results when faced with irrelevant objects: (a, b) curved hole-edge, (c) transparent edge, and (d) sensor box and wire.

5. Conclusions

This paper introduced an improved WSL-based semantic segmentation method for accurate fatigue crack detection in steel bridge girders. The proposed method utilized the annotation map refinement (AMR) technique to generate high-quality initial pseudolabels, overcoming the limitation of highlighting only discriminative regions in conventional WSL-based methods. These pseudolabels were then used to train the segmentation model in a two-stage approach. First, the model learned essential semantic and localization information from the initial labels. Then, the model was further refined iteratively using a segmentation refinement module equipped with postprocessing algorithms.

Experimental evaluations compared the proposed method with different labeling tools and state-of-the-art techniques, demonstrating faster image-level annotation and the superiority of AMR in generating more accurate and complete object regions, leading to an improvement for pseudolabels in Intersection over Union (IoU) accuracy by approximately 2%. Ablation studies confirmed the effectiveness of the main components, and comparisons with traditional WSL-based and FSL-based methods revealed superior performance by the proposed method. The visualizations of real-bridge conditions showcased the model's ability to accurately detect genuine cracks and markers. However, further optimization, including data augmentation, is needed to enhance performance under challenging conditions. Overall, our method achieves comparable inference results to FSL-based approaches while significantly reducing annotation workload. Further validation is recommended to assess its effectiveness in more diverse scenarios, and future research should focus on studying the effect of segmentation network structures and integrating the proposed method with more advanced networks to enhance its performance.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors appreciate the support of the National Natural Science Foundation of China (grant no. 52378288), the Distinguished Young Scientists of Jiangsu Province (grant no. BK20190013), and the Jiangsu Natural Science Foundation (grant no. BK20211003 and BK20221400).

References

[1] Y. L. Zhou, X. Qian, A. Birnie, and X. L. Zhao, "A reference free ultrasonic phased array to identify surface cracks in welded steel pipes based on transmissibility," *International Journal of Pressure Vessels and Piping*, vol. 168, pp. 66–78, 2018.

[2] D. Wang, Y. Dong, Y. Pan, and R. Ma, "Machine vision-based monitoring methodology for the fatigue cracks in U-Rib-to-deck weld seams," *IEEE Access*, vol. 8, pp. 94204–94219, 2020.

[3] Y. J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[4] C. M. Yeum and S. J. Dyke, "Vision-based automated crack detection for bridge inspection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 10, pp. 759–770, 2015.

[5] J. Zhao, F. Hu, W. Qiao et al., "A modified U-net for crack segmentation by Self-Attention-Self-Adaption neuron and random elastic deformation," *Smart Structures and Systems*, vol. 29, no. 1, pp. 1–16, 2022.

[6] M. O'Byrne, B. Ghosh, F. Schoefs, and V. Pakrashi, "Regionally enhanced multiphase segmentation technique for damaged surfaces," *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 9, pp. 644–658, 2014.

[7] L. Wu, S. Mokhtari, A. Nazef, B. Nam, and H. B. Yun, "Improvement of crack-detection accuracy using a novel crack defragmentation technique in image-based road assessment," *Journal of Computing in Civil Engineering*, vol. 30, no. 1, Article ID 4014118, 2016.

[8] G. Mirzaei, A. Adeli, and H. Adeli, "Imaging and machine learning techniques for diagnosis of Alzheimer's disease," *Reviews in the Neurosciences*, vol. 27, no. 8, pp. 857–870, 2016.

[9] K. W. Liao and Y. T. Lee, "Detection of rust defects on steel bridge coatings via digital image recognition," *Automation in Construction*, vol. 71, pp. 294–306, 2016.

[10] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, 2018.

[11] C. V. Dung, H. Sekiya, S. Hirano, T. Okatani, and C. Miki, "A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks," *Automation in Construction*, vol. 102, pp. 217–229, 2019.

[12] W. Nash, T. Drummond, and N. Birbilis, "Quantity beats quality for semantic segmentation of corrosion in images," 2018, https://arxiv.org/abs/1807.03138.

[13] X. W. Ye, T. Jin, Z. X. Li, S. Y. Ma, Y. Ding, and Y. H. Ou, "Structural crack detection from benchmark data sets using pruned fully convolutional networks," *Journal of Structural Engineering*, vol. 147, no. 11, Article ID 4721008, 2021.

[14] Z. Dong, J. Wang, B. Cui, D. Wang, and X. Wang, "Patch-based weakly supervised semantic segmentation network for crack detection," *Construction and Building Materials*, vol. 258, Article ID 120291, 2020.

[15] D. Zhang, K. Song, J. Xu, H. Dong, and Y. Yan, "An image-level weakly supervised segmentation method for no-service rail surface defect with size prior," *Mechanical Systems and Signal Processing*, vol. 165, Article ID 108334, 2022.

[16] Q. Li and X. Liu, "Novel approach to pavement image segmentation based on neighboring difference histogram method," *Congress on Image and Signal Processing*, vol. 2, 2008.

[17] H. D. Cheng and M. Miyojim, "Automatic pavement distress detection system," *Information Sciences*, vol. 108, no. 1–4, pp. 219–240, 1998.

[18] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *Proceedings of the European Signal Processing Conference*, Glasgow, UK, August 2009.

[19] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, vol. 17, no. 4, pp. 255–263, 2003.

[20] 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.

[21] T. Yamaguchi, S. Nakamura, R. Saegusa, and S. Hashimoto, "Image-based crack detection for real concrete surfaces," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 3, no. 1, pp. 128–135, 2008.

[22] S. K. Sinha and P. W. Fieguth, "Automated detection of cracks in buried concrete pipe images," *Automation in Construction*, vol. 15, no. 1, pp. 58–72, 2006.

[23] M. Song and D. Civco, "Road extraction using SVM and image segmentation," *Photogrammetric Engineering and Remote Sensing*, vol. 70, no. 12, pp. 1365–1371, 2004.

[24] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.

[25] F. Liu, G. Xu, Y. Yang, X. Niu, and Y. Pan, "Novel approach to pavement cracking automatic detection based on segment extending," in *Proceedings of the 2008 International Symposium on Knowledge Acquisition and Modeling*, Wuhan, China, December 2008.

[26] W. Huang and N. Zhang, "A novel road crack detection and identification method using digital image processing techniques," in *Proceedings of the 2012 7th International Conference on Computing and Convergence Technology*, ICCCT, Seoul, Republic of Korea, December 2012.

[27] Q. Li, Q. Zou, D. Zhang, and Q. Mao, "FoSA: F∗ Seed-growing Approach for crack-line detection from pavement images," *Image and Vision Computing*, vol. 29, no. 12, pp. 861–872, 2011.

[28] H. B. Yun, S. Mokhtari, and L. Wu, "Crack recognition and segmentation using morphological image-processing techniques for flexible pavements," *Transportation Research Record*, vol. 2523, no. 1, pp. 115–124, 2015.

[29] Y. Hu and C. X. Zhao, "A novel LBP based methods for pavement crack detection," *Journal of Pattern Recognition Research*, vol. 5, no. 1, pp. 140–147, 2010.

[30] M. Petrou, J. Kittler, and K. Y. Song, "Automatic surface crack detection on textured materials," *Journal of Materials Processing Technology*, vol. 56, no. 1–4, pp. 158–167, 1996.

[31] K. Y. Song, M. Petrou, and J. Kittler, "Texture crack detection," *Machine Vision and Applications*, vol. 8, no. 1, pp. 63–75, 1995.

[32] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme, "An innovative methodology for detection and quantification of cracks through incorporation of depth perception," *Machine Vision and Applications*, vol. 24, no. 2, pp. 227–241, 2013.

[33] C. Dong, L. Li, J. Yan, Z. Zhang, H. Pan, and F. N. Catbas, "Pixel-level fatigue crack segmentation in large-scale images of steel structures using an encoder–decoder network," *Sensors*, vol. 21, no. 12, p. 4135, 2021.

[34] Y. Xu, Y. Bao, J. Chen, W. Zuo, and H. Li, "Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images," *Structural Health Monitoring*, vol. 18, no. 3, pp. 653–674, 2019.

[35] S. Quqa, P. Martakis, A. Movsessian, S. Pai, Y. Reuland, and E. Chatzi, "Two-step approach for fatigue crack detection in steel bridges using convolutional neural networks," *Journal of Civil Structural Health Monitoring*, vol. 12, no. 1, pp. 127–140, 2022.

[36] 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*, CVPR, June 2015.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, June 2016.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, June 2016.

[40] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction*, vol. 99, pp. 52–58, 2019.

[41] X. Li, T. Lai, S. Wang et al., "Weighted feature pyramid networks for object detection," in *Proceedings of the 2019 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking*, ISPA/BDCloud/SocialCom/SustainCom, December 2019.

[42] J. Li, Z. Jin, and J. Shu, "Deep learning-based fatigue cracks detection in bridge girders using feature pyramid networks," *Research Square*, 2021.

[43] 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.

[44] J. Shi, J. Dang, M. Cui et al., "Improvement of damage segmentation based on pixel-level data balance using vgg-unet," *Applied Sciences*, vol. 11, no. 2, p. 518, 2021.

[45] L. Zhang, J. Shen, and B. Zhu, "A research on an improved Unet-based concrete crack detection algorithm," *Structural Health Monitoring*, vol. 20, no. 4, pp. 1864–1879, 2021.

[46] X. Cui, Q. Wang, J. Dai, Y. Xue, and Y. Duan, "Intelligent crack detection based on attention mechanism in convolution neural network," *Advances in Structural Engineering*, vol. 24, no. 9, Article ID 1369433220986638, 2021.

[47] Z. M. P. H. Q. Li, "One-step deep learning-based method for pixel-level detection of fine cracks in steel girder images," *Smart Structures and Systems*, vol. 29, no. 1, pp. 153–166, 2022.

[48] Z. Li, H. Zhu, and M. Huang, "A deep learning-based fine crack segmentation network on full-scale steel bridge images with complicated backgrounds," *IEEE Access*, vol. 9, pp. 114989–114997, 2021.

[49] S. Kim, L. T. Nguyen, K. Shim, J. Kim, and B. Shim, "Pseudo-label-free weakly supervised semantic segmentation using image masking," *IEEE Access*, vol. 10, pp. 19401–19411, 2022.

[50] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, June 2016.

[51] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting Dilated Convolution: a simple approach for weakly- and semi-supervised semantic segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.

[52] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: three principles for weakly-supervised image segmentation," in *Proceedings of the Computer Vision– ECCV 2016*, pp. 695–711, Amsterdam, The Netherlands, October 2016.

[53] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4981–4990, Salt Lake City, UT, USA, June 2018.

[54] Y. T. Chang, Q. Wang, W. C. Hung, R. Piramuthu, Y. H. Tsai, and M. H. Yang, "Weakly-supervised semantic segmentation via sub-category exploration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8988–8997, Seattle, WA, USA, June 2020.

[55] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "FickleNet: weakly and semi-supervised semantic image segmentation using stochastic inference," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5262–5271, Long Beach, CA, USA, June 2019.

[56] J. Qin, J. Wu, X. Xiao, L. Li, and X. Wang, "Activation modulation and recalibration scheme for weakly supervised semantic segmentation," *AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, pp. 2117–2125, 2022.

[57] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does It: weakly supervised instance and semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[58] Y. Li, Y. Liu, G. Liu, and M. Guo, "Weakly supervised semantic segmentation by iterative superpixel-CRF refinement with initial clues guiding," *Neurocomputing*, vol. 391, pp. 25–41, 2020.

[59] H. Choi, G. Koo, B. J. Kim, and S. W. Kim, "Weakly supervised power line detection algorithm using a recursive noisy label update with refined broken line segments," *Expert Systems with Applications*, vol. 165, Article ID 113895, 2021.

[60] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with Gaussian edge potentials," *Advances in Neural Information Processing Systems*, vol. 24, pp. 1–9, 2011.

[61] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical image computing and computer-assisted intervention*, Munich, Germany, October 2015.

[62] Y. Bao, J. Li, T. Nagayama, Y. Xu, B. F. Spencer, and H. Li, "The 1st international Project competition for structural Health monitoring (IPC-shm, 2020): a summary and benchmark problem," *Structural Health Monitoring*, vol. 20, no. 4, pp. 2229–2239, 2021.

[63] 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.

[64] Baidu, "EasyDL," 2022, https://ai.baidu.com/easydl/.

[65] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[66] Q. Han, X. Liu, and J. Xu, "Detection and location of steel structure surface cracks based on unmanned aerial vehicle images," *Journal of Building Engineering*, vol. 50, no. January, Article ID 104098, 2022.