Probability-Guided Edge Enhancement Network for Remote Sensing Image Semantic Segmentation

Chunyan Yu, *Senior Member, IEEE*, Yakun Zuo, Qiang Zhang, *Member, IEEE*, and Yulei Wang, *Member, IEEE*

***Abstract*— Semantic segmentation in remote sensing images (RSIs) assigns unique semantic labels to each pixel and plays a crucial role in real-world applications such as environmental change monitoring, precision agriculture, and economic assessment. Although convolutional neural networks (CNNs) and Transformer-based models for semantic segmentation of RSIs have achieved remarkable success, existing approaches still struggle to accurately detect weak edges and occluded objects due to the complexity and fuzziness of edges in RSIs. To overcome this obstacle, we propose a novel probability-guided edge enhancement network (PEEN) for semantic segmentation of RSIs, which is the first attempt to leverage the probability function (PF) to guide the segmentation model in performing edge prediction for RSIs. Specifically, in the feature extraction stage of PEEN, we present a convolutional self-attention mechanism to enhance the global feature representation of the encoder–decoder network. In the edge enhancement stage of PEEN, we innovatively build an iterative probability-guided edge prediction module to refine edge prediction mathematically and iteratively. With the cooperation of the mentioned two stages, the proposed model yields precise segmentation of the objects and edge portions in RSIs. Experimental results and analysis demonstrate that the PEEN model outperforms the existing popular CNN-based and Transformer-based models in semantic segmentation with 85.54% and 88.35% of mean intersection over union (mIOU) on the Vaihingen and Potsdam test datasets, respectively. Our code is available at https://github.com/Zyk517/PEEN**

***Index Terms*— Edge enhancement, probability-guided, remote sensing, semantic segmentation.**

I. Introduction

IN RECENT years, remote sensing image (RSI) interpretation technology [1], [2] has made remarkable advancements and plays a pivotal role in various fields such as precision agriculture [3], environmental monitoring [4], urban planning [5], [6], and disaster management [7], [8]. Among the applications, semantic segmentation of RSIs (SSRSIs) [9], [10] aims to partition the RSI into semantically meaningful regions and assign each pixel with a specific class label, which is crucial for practical applications including vegetation analysis [11], urban cover investigation [12], and identification of natural disasters like floods and wildfires [13].

The development of semantic segmentation tasks has been extremely rapid in the past few decades. Initially, traditional semantic segmentation methods generally relied on handcrafted feature extraction that was implemented by segmentation algorithms such as region growing [14], graph cuts [15], and Markov random fields [16]. While effective in certain cases, traditional methods often struggled in complex scenes and could not learn hierarchical representations directly from RSIs. The emergence of convolutional neural networks (CNNs) [17] promotes the progress of RSI methods. Wang et al. [18] proposed the adaptive feature fusion UNet (AFF-UNet), which incorporated a channel attention convolution block and a spatial attention block based on CNNs. With the collaboration of the blocks, the AFF-UNet approach effectively addressed the challenges of varying object sizes and class confusion in RSIs. Zeng et al. [19] proposed a multiscale global context network (MSGCNet) for SSRSIs, which employed convolution kernels of different sizes to establish a multiscale perception fusion model, and effectively decreased the problems of target scale differences and class confusion in RSIs. Compared with traditional methods, the CNN performs well in capturing fine-grained features and improves the segmentation ability of RSIs to a large extent. Nevertheless, the CNN-based methods merely capture local context information and neglect global context information in RSIs. In recent years, inspired by the remarkable success of the Transformer models in natural language processing tasks, Transformer-based methods have gained traction as promising approaches in semantic segmentation. Unlike CNNs, Transformers employ self-attention [20] mechanisms to capture long-range information from fed samples and supply effective modeling of spatial context and global relationships. Li et al. [21] proposed a synergistic attention perception neural network (SAPNet) to relieve the attention bias problem of SSRSIs with the presented synergistic attention module and space–channel attention. Liu et al. [22] proposed a global–local Transformer segmentor (GLOTS) framework for RSIs, which designed a global–local attention module to solve the problems of inconsistent feature representation and insufficient utilization of context information. Nevertheless, self-attention mechanisms demand significant computational resources and tend to emphasize global information excessively [23]. As a result, transform-based semantic

Received 23 April 2025; revised 5 June 2025; accepted 7 July 2025. Date of publication 15 July 2025; date of current version 28 July 2025. This work was supported by the National Natural Science Foundation of China under Grant 62471079 and 62401095. *(Corresponding author: Qiang Zhang.)*

The authors are with the Center for Hyperspectral Imaging in Remote Sensing (CHIRS) at Information and Technology College, Dalian Maritime University, Dalian 116026, China (e-mail: yucy@dlmu.edu.cn; zyk19980517@dlmu.edu.cn; qzhang95@dlmu.edu.cn; wangyulei@dlmu.edu.cn).

Digital Object Identifier 10.1109/TGRS.2025.3589235

Fig. 1. Exhibit the case where the small objects are occluded. (a) Car is shadowed by the building. (b) Car is obscured by the building or a tree.

segmentation models are prone to ignoring the local details of RSIs and generating undesirable segmentation results at edges.

As is known, capturing fine-grained details and boundaries between different objects is crucial for SSRSIs. As illustrated in Fig. 1, small objects (e.g., cars) are occluded by larger objects (e.g., buildings or trees) or obscured by the shadows in RSIs. In such situations, traditional CNN-based or Transformer-based methods often encounter difficulties in precisely delineating object boundaries. To address this issue, researchers have made numerous efforts and developed various edge enhancement techniques [24], [25], [26] to strengthen the detection of edges or boundaries. The popular techniques consist of edge detection operators [27] and multiscale feature fusion [28], incorporating edge information [29] and constraining with edge-related loss functions [30]. Although edge enhancement methods enhance the detection of edge information to a certain extent, existing approaches still have limitations in accurate edge prediction, especially with occlusion and shading of RSIs.

In this article, we propose a novel probability-guided edge enhancement network (PEEN) for SSRSIs. The core idea of the presented network solves the problem of predicting edge pixels from a mathematical probability point of view for the first time. Specifically, in the feature extraction stage, the PEEN model incorporates the convolutional self-attention (Conv SA) mechanism to capture long-range dependencies through depth-separable convolution with large kernels for segmenting occluded objects. In the edge prediction enhancement module, we developed the iterative probability-guided edge prediction (IPEP) module, which employs the PF to predict edge pixels with the supervision of edge labels accurately. Furthermore, the PEEN network incorporates asymmetric convolutional blocks during the iterative process, thereby facilitating the segmentation precision of edge pixels gradually.

In summary, the main contributions are as follows.

1) Unlike the traditional methods that rely on designed networks or constrained loss functions for edge enhancement, we conceptually present a new PEEN for SSRSIs. Our model converts edge prediction to a distance probability computation task and mathematically achieves edge detection, which is the first attempt to formulate edge prediction as a probabilistic task in semantic segmentation.
2) A pixel probability-to-boundary distance mapping is established through a specially designed PF, which incorporates relative positional relationships between pixels rather than relying solely on isolated pixel features. Furthermore, an iterative scheme is designed in conjunction with different PFs to achieve progressive refinement of edge predictions.
3) The innovative Conv SA mechanism in the encoder is proposed to capture long-range dependencies with large kernel convolutions, which is beneficial to address the segmentation issues of occluded objects in RSIs by effectively extracting global contextual information. With this mechanism, the encoder acquires spatial relationships and intricate features of the surrounding objects, leading to enhanced embedding for subsequent recognition.

The rest of this article is organized as follows. Section II discusses the related work. Section III introduces the proposed method in detail. Section IV reports the experiments and provides a discussion of the experimental results. Finally, the conclusion is outlined in Section V.

II. Related Works

In this section, we comprehensively analyze deep learning-based approaches for SSRSIs, including CNN-based methods, Transformer-based methods, and edge enhance-based methods.

A. CNN-Based Semantic Segmentation Methods

The fully convolutional network (FCN) [31] is recognized as the pioneering CNN architecture that effectively tackles semantic segmentation tasks in an end-to-end manner. Subsequently, CNN-based approaches have emerged as the dominant methods in the field of SSRSIs [32], [33], [34], [35], [36]. Nevertheless, the oversimplified decoder architecture of FCNs restricts the ability to extract higher-level features and global contextual information, consequently impacting the accuracy of semantic segmentation. To accommodate input images of different sizes and preserve spatial information, Ronneberger et al. [37] proposed the classical UNet network based on the encoder–decoder structure. By incorporating skip connections between the encoder and the decoder, the UNet directly connects features from different layers of the encoder to corresponding layers of the decoder, which mitigates issues like information loss and gradient vanishing. Following the

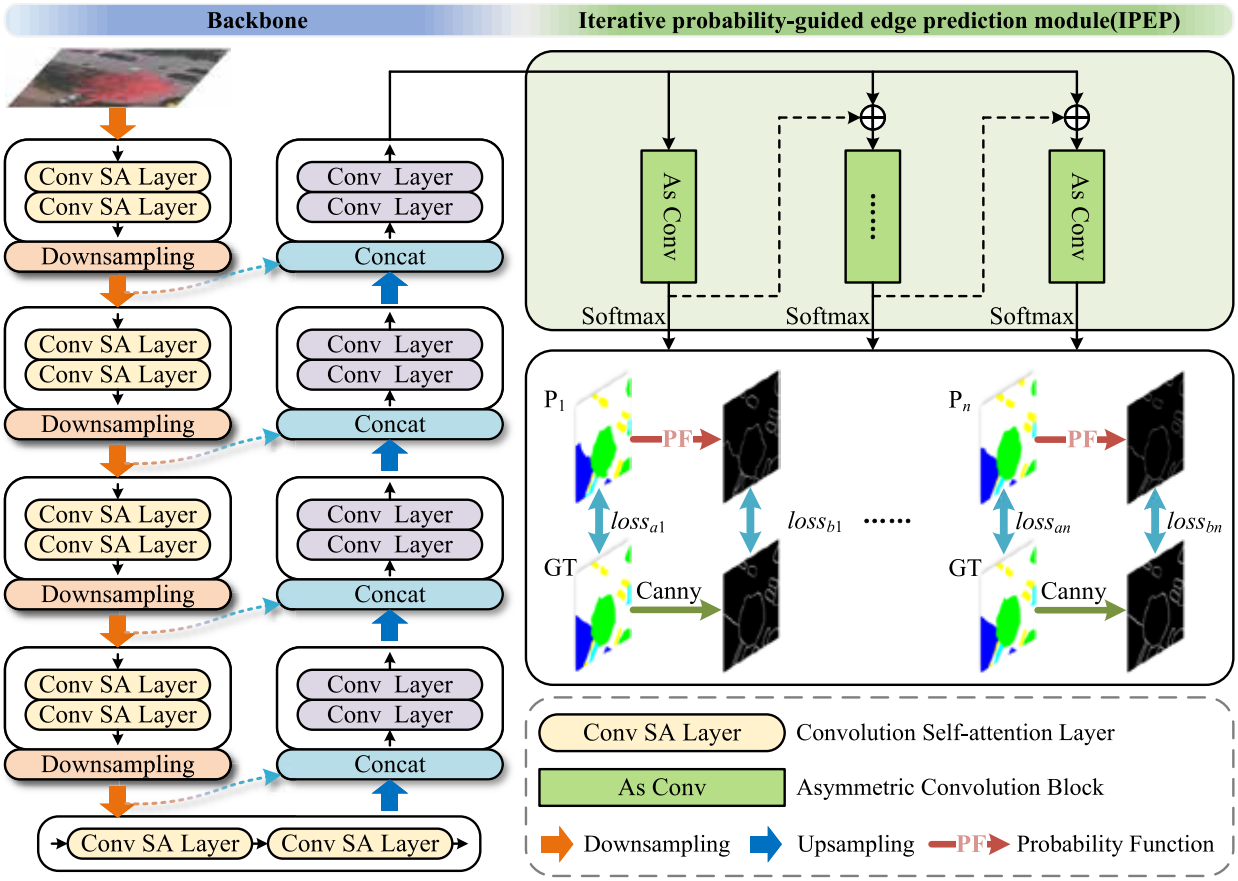

Fig. 2. Framework of the proposed PEEN model. The backbone is employed to extract fusion features with global context information. Subsequently, the fusion features are fed into the IPEP module to complete the edge enhancement gradually and iteratively through PFs.

UNet, the encoder–decoder framework has become the prevailing structure of RSI segmentation networks [38], [39]. Typically, Diakogiannis et al. [40], Yue et al. [41], and Zhou et al. [42] proposed distinct skip connections to capture more comprehensive contextual information. Meanwhile, Liu et al. [43], Zhao et al. [44], and Shen et al. [45] introduced diverse decoder architectures to preserve semantic information effectively.

Although the aforementioned CNN-based methods have achieved encouraging performance, they encounter bottlenecks in SSRSIs. Specifically, CNN-based segmentation networks with restricted receptive fields only extract local semantic features and cannot model global context information. In RSIs, the occlusion of small objects presents a challenge for precisely segmenting the edges based solely on local information.

B. Transformer-Based Semantic Segmentation Methods

With the emergence of the Vision Transformer (ViT) [46] model, researchers have attempted to apply Transformer-based methods with self-attention mechanisms in the semantic segmentation task. Most of the existing Transformers for semantic segmentation still employ the encoder–decoder framework, which is divided into two categories according to different encoder–decoder combination mechanisms. The first category is constructed by a Transformer-based encoder and decoder structure. Typical models include the Segmenter [47], SegFormer [48], and SwinUNet [49]. The second category adopts a hybrid structure, which is composed of a Transformer-based encoder and a CNN-based decoder. Generally speaking, Transformer-based SSRSI methods commonly employ the second structure. For example, Wang et al. [50] proposed a dual-branch hybrid CNN–Transformer network (DBCT-Net), which fully exploited the advantages of CNNs in local specific feature extraction and achieved the global dependencies through the Transformer part. Liu et al. [51] presented a Transformer-based multimodal fusion network (TMFNet) to significantly improve the segmentation accuracy of small targets in RSIs through the edge region attention module.

Although the Transformer-based models have demonstrated potential in semantic segmentation tasks, the complexity of the self-attention mechanism is much higher than the CNN and impacts the feasibility of the model in SSRSIs. Besides, Transformer-based models leverage a self-attention mechanism to capture global dependencies within the input sequence and struggle to effectively capture local fine-grained information, resulting in inaccurate segmentation of edge objects of RSIs.

C. Edge Enhance-Based Semantic Segmentation Methods

In image segmentation, boundary information is crucial to improve segmentation accuracy significantly. Recently,

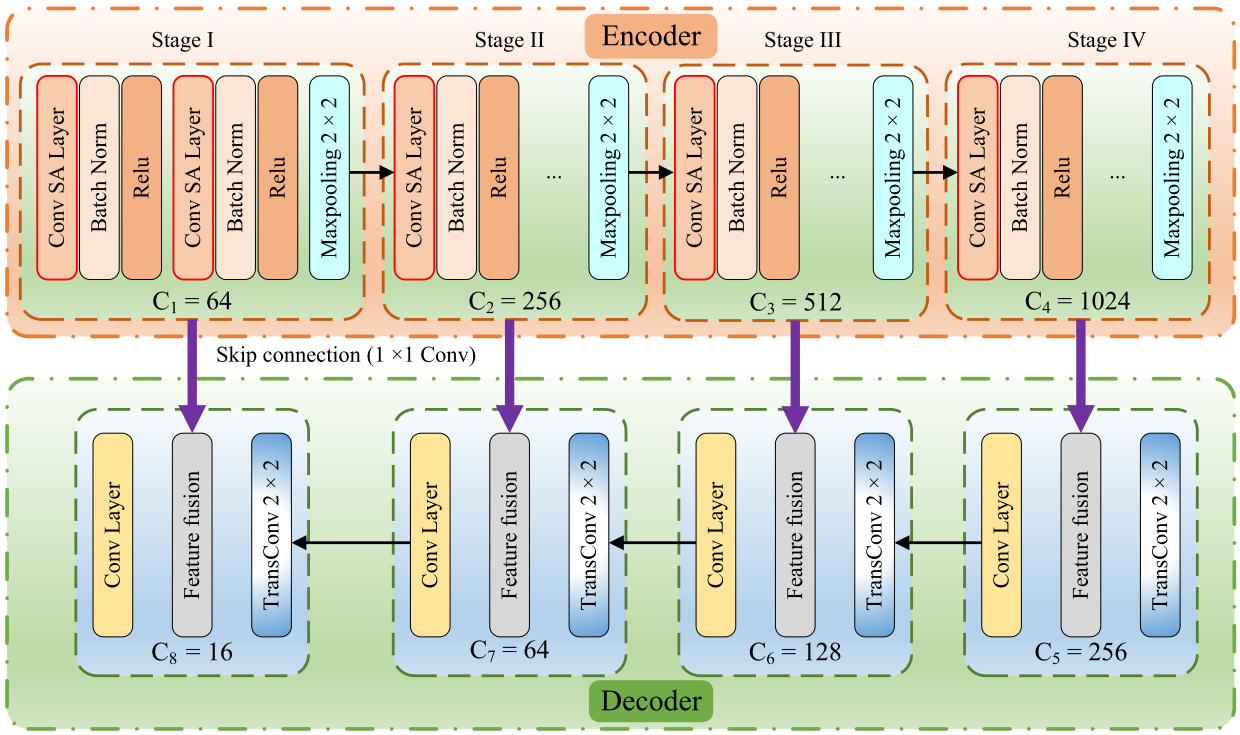

Fig. 3. Architecture of the backbone. We build the encoder with a Conv SA mechanism to capture multiscale global context features. Besides, the upsampling in the decoder is performed by transposed convolution. C_i represents the number of channels of each stage.

CNN-based models have been increasingly applied to effectively extract edges for image segmentation, for example, ECAE [52], BIBED-Seg [53], and EPFNet [54]. Nevertheless, effectively fusion edge information into semantic segmentation tasks is a significant challenge.

Currently, edge-enhanced semantic segmentation methods are broadly classified into two categories. The first category involves the utilization of specific edge loss functions to measure and minimize the discrepancy between the segmentation outcomes and the ground-truth edges. For instance, Sun et al. [55] proposed an SSRSI model that incorporates an adaptive edge loss function constraint, which aims to alleviate the challenges of recognizing small objects and address the issue of sample imbalance. Li et al. [56] proposed a semantic segmentation network with enhanced edge loss and preserved spatial boundary information through the supervision of multiple weighted edge losses. The second category fuses the edge information to enhance the edge information description and obtain continuous boundaries. For example, Sun et al. [57] proposed a boundary attention module to enhance the representation of edge information and alleviate the issue of blurred segmentation boundaries. He et al. [58] proposed the Edge–FCN network by introducing edge information as prior knowledge into FCNs to revise the segmentation results. Despite the aforementioned methods achieving relatively accurate segmentation results along the edges, the core mechanism relies on detecting edge information from the image for segmentation. Notably, the utilization of CNN-based models or edge operators for extracting edge information from images is subject to the complexity of the image. Since the intricate nature of object information in RSIs, the extracted edge information often appears fragmented and incomplete, which limits the applicability to SSRSIs.

III. Methodology

Fig. 2 illustrates the framework of the proposed PEEN model that primarily contains backbone and IPEP modules. The backbone network employs an encoder–decoder architecture, where the encoder captures long-range multiscale contextual features through the Conv SA mechanism. The decoder then utilizes transposed convolutions to gradually restore the feature maps to the same spatial dimensions as the original image, after which the processed feature maps are fed into the IPEP module. In the IPEP module, we present the iterative module (IM) with asymmetric convolutions to enhance feature extraction from the backbone and generate corresponding semantic predictions. Besides, we adopt PFs to iteratively optimize the edge prediction results in the IPEP module.

A. Encoder Based on Conv SA

The detailed structure of the encoder is illustrated in the top half of Fig. 3. Specifically, the encoder consists of 4 downsampling stages. Each stage utilizes a designed large kernel convolution-based Conv SA mechanism to extract nonlocal features, which is followed by maximum pooling to reduce the resolution of the embedding maps by a factor of 2. Moreover, we implement skip connections with the 1×1 convolution operation, which aids in feature refinement and mitigating the loss of edge information.

1) Convolutional Self-Attention: In the encoder, we propose the Conv SA mechanism to handle the contextual information. To exhibit the difference between Conv SA and self-attention mechanisms clearly, we illustrate the computational processes of Conv SA and traditional self-attention mechanisms in Fig. 4(a) and (b), respectively. As observed, the Conv SA has a structure similar to the self-attention mechanism. The distinction lies in the pattern adopted to generate the similarity score matrix **A**. Each element $A_{(i,j)}$ in the similarity score matrix **A** represents the correlation score between the ith element and the jth element in the sequence; the higher the score, the stronger the correlation between the two elements. Instead of utilizing query (**Q**) and key (**K**), Conv SA employs a $k \times k$ depth-separable convolution to generate **A**. Subsequently, Conv SA performs a Hadamard product [59] with the value (**V**) to obtain the final output. Besides, the Conv SA is exclusively composed of convolution operations and exhibits a linear growth in complexity. Consequently, Conv SA retains the long-range modeling capability similar to self-attention while reducing computational costs. Specifically, given the input tokens $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, we adopt a simple depth-wise convolution with kernel size $k \times k$ and the Hadamard product to calculate the output **Z** as follows:

$$\mathbf{A} = \mathrm{DConv}_{k\times k}(\mathbf{W_1 X}) \tag{1}$$

$$\mathbf{V} = \mathbf{W_2 X} \tag{2}$$

$$\mathbf{Z} = \mathbf{A} \odot \mathbf{V} \tag{3}$$

where $\odot$ is the Hadamard product,W_1 means the weight matrix of the $k \times k$ deep convolution, W_2 is the value matrix, and $\mathrm{Dconv}_{k\times k}$ denotes a depth-wise convolution with kernel size $k \times k$.

2) Large Kernel in Conv SA: Although a small-sized convolution kernel (e.g., 3 × 3) is popular in CNNs such as VGGNet [60] and ResNet [61], it is worth noting that the 3 × 3 convolutional kernel captures mainly smaller local information in each convolutional operation and is unable to acquire global contextual information. In the context of SSRSIs, the 3 × 3 convolution kernel has a relatively local receptive field, which does not effectively capture long-range dependencies in RSIs and results in effectively identifying occluded objects. In contrast, the large kernel convolutions in Conv SA have a larger receptive field, enabling the model to capture features over a wider range and assist in handling occluded objects more effectively. In our article, we set the kernel size as 15 × 15 in Conv SA and analyze the impact of varying kernel sizes on the model in the subsequent experimental section.

B. Decoder Based on Transposed Convolution

The structure of the decoder is demonstrated in the bottom half of Fig. 3. Likewise, the decoder consists of 4 upsampling stages. In each stage, the resolution of the feature map is augmented by a factor of 2 with transposed convolution. Subsequently, feature fusion occurs between the corresponding decoder and encoder via skip connections, and a sequence of convolutional layers is employed to further process the features to extract higher-level semantic information. Through 4 consecutive upsampling stages, the PEEN model generates a feature map with a dimension identical to that of the input RSI.

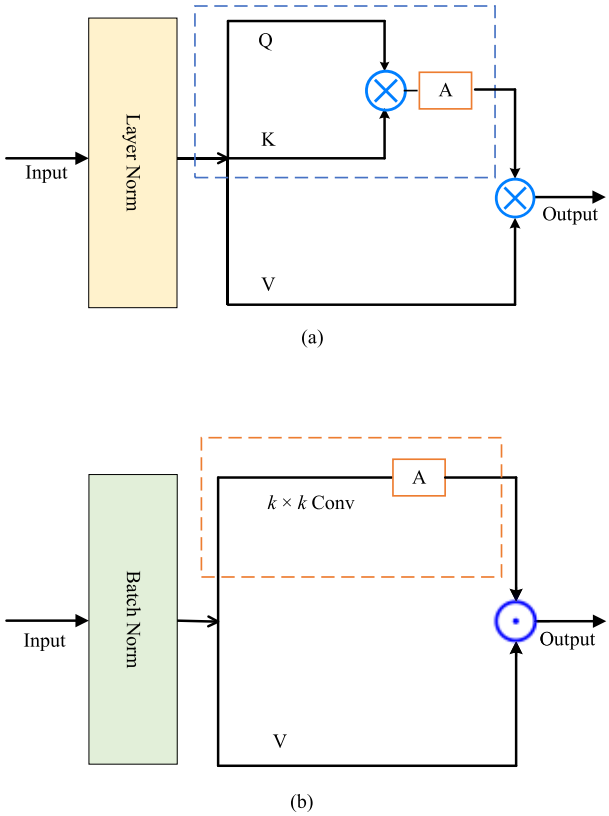

Fig. 4. Comparison of Conv SA with the self-attention mechanism. Instead of generating attention matrices via a matrix multiplication between Q and K, we directly produce weights with a $k \times k$ depth-wise convolution to reweigh the value via the Hadamard product in Conv SA (⊗: matrix multiplication, ⊙: Hadamard product). (a) Self-attention. (b) Conv SA.

C. IPEP Module

1) Probability-Based Edge Prediction Method:

a) Principle clarification: In this section, we introduce the principle clarification based on the probability-based edge prediction method and present the schematic. Specifically, as illustrated in Fig. 5, the green contour indicates the boundary of the building, D, E, F, and G represent four pixels on the building, and H_E, H_F, and H_G denote the shortest distances from E, F, and G to the boundary, respectively. We assume the probability of pixel G belonging to the boundary pixel is 0 and the probability of pixel D is 1; thus, the probabilities of pixels E and F belonging to the boundary pixel are determined as $1\text{-}H_E/H_G$ and $1\text{-}H_F/H_G$, respectively. Notably, the four points and green contour are merely examples, including both edge and nonedge pixels, which are not four boundaries actually. The calculation process of pixel distance is not limited to a specific boundary, while we compute the distances between the inner pixel of a remote sensing object and all pixels on the boundary, and select the minimum one. In this way, the distance is converted to the probability that it belongs to the boundary pixels with the PF.

b) Probability-guided edge prediction: Due to the ambiguity and uncertainty of RSI edge pixels, it is not possible

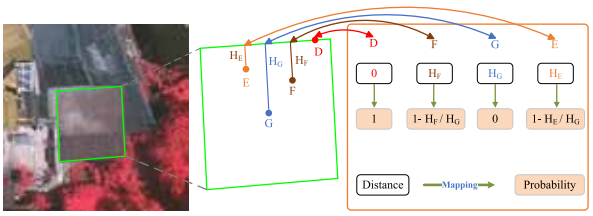

Fig. 5. Illustration of the relation between probability and the distance to the boundary. H_E, H_F, and H_G are the shortest distance from pixels E, F, and G to the boundary, respectively.

to accurately predict RSI edge pixels by employing a simple distance ratio. To accurately predict all possible edge pixels, we design the PF based on the Sigmoid function [62], which is defined as follows:

$$\mathrm{PF}_{(i,j)} = 1 - C * \left(\frac{2}{1 + e^{\frac{-\alpha H_{(i,j)}}{R}}} - 1 \right) \tag{4}$$

$$C = \frac{1 + e^{-\alpha}}{1 - e^{-\alpha}};\ \alpha \in (0, \infty) \tag{5}$$

where α is a parameter for generating different PFs and C is a constant for keeping the value range of the PF in [0, 1] and ensuring the probability of the farthest pixel belonging to the boundary pixel is 0. (i, j) denotes the coordinates of the pixel in RSIs and $H_{(i,j)}$ is the shortest distance from pixel (i, j) to the boundary. R represents the scale of the segmented object and is defined as

$$R = \max\left(H_{(i,j)}\right); i, j \in O \tag{6}$$

where O represents the segmented objects.

2) Iteration Module: The detailed structure of the IM is shown in Fig. 6, which employs the 1×3 and 3×1 asymmetric convolution block at each iteration to capture and enhance the features at the edges. Compared with the convolution operation with a kernel size of 3×3, the asymmetric convolution with a kernel size of 1×3 and 3×1 is beneficial to enable the perception and extraction of edge information. In other words, the asymmetric convolution operation is sensitive in specific directions and accurately captures local directional details of the edges. With the assistance of the IM, our network fully leverages the information from previous predictions to iteratively refine and improve the accuracy of the final predictions. Besides, the parameters in the iteration layers are optimized by a gradient back-propagation algorithm. Specifically, assuming f_0 is the output of the backbone, the obtained feature in the IM is formulated as

$$f_1 = \mathrm{ConvBlock}(f_0);\ P_1 = \sigma(f_1) \tag{7}$$

$$f_i = \mathrm{ConvBlock}(f_0 \oplus f_{i-1}) \tag{8}$$

$$P_i = \sigma(f_i);\quad i \in (1, n] \tag{9}$$

where ConvBlock refers to the asymmetric convolution block of 1×3 and 3×1, P_i represents the segmentation prediction result for each iteration,"$\oplus$" refers to the concatenation operation, $\sigma(\cdot)$ means the Softmax activation function, and n denotes the number of iterations.

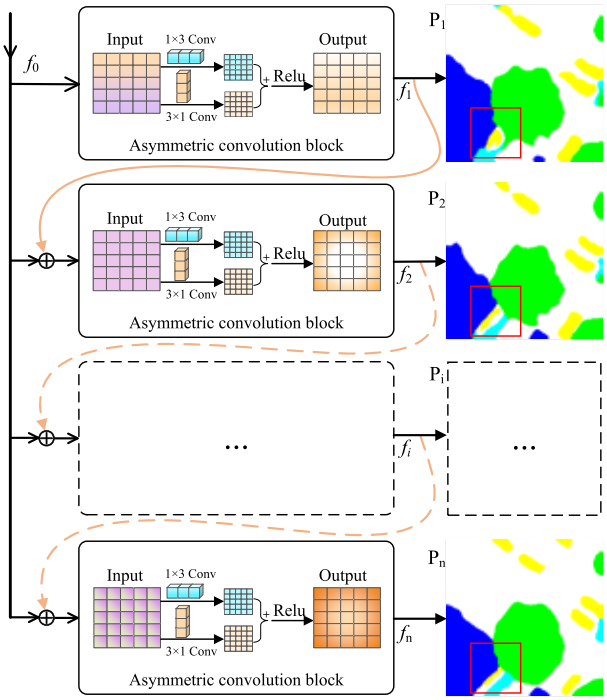

Fig. 6. Structure of the IM in the IPEP module, where f_0 represents the feature extracted by the backbone and the set $\{P_1, P_2, \ldots, P_i, \ldots, P_n\}$ is the prediction of segmentation results.

Algorithm 1 IPEP Module Algorithm

Require: The output of the backbone f_0
Ensure: The prediction results for each iteration P_n

1: **for** $n \in \{1, \ldots, i, n\}$ **do**
2: Feed f_0 into the IPEP module.
3: $f_1 \leftarrow f_0$ is enhanced via asymmetric convolution as defined in Equation (7), generating f_1 and the first iteration segmentation prediction P_1.
4: $f_i \leftarrow f_0$ and f_{i-1} are concatenated via Equation (8) to derive f_i.
5: $P_i \leftarrow f_i$ generates the ith iteration prediction P_i via Equation (9).
6: **end for**
7: **for** $Loss_{ai}$, $Loss_{bi}$ and $Loss_i$ in $\{1, \ldots, i, n\}$ **do**
8: $Loss_{ai} \leftarrow$ Calculate the mse loss for the ith iteration with Equation (11).
9: $Loss_{bi} \leftarrow$ Calculate the Dice coefficient loss for the ith iteration with Equation (12).
10: $Loss_i \leftarrow$ Calculate the total loss with Equation (13).
11: **end for**
12: $L_s \leftarrow$ Calculate loss of the IPEP module with Equation (14).
13: $\theta_i^{(t+1)} = \theta_i^{(t+1)} - \eta \frac{\partial L_s}{\partial \theta_i^{(t)}} \leftarrow$ Gradient update, η is learning rate, t is the number of training.
14: **End**

The algorithm flow of the IPEP module is shown in Algorithm 1. After obtaining the initial feature map f_0 from the backbone, we propagate f_0 into the first iterative layer of the IM and yield f_1. Then, f_1 is utilized to generate the prediction result P_1 through the Softmax function and fed into

TABLE I
QUANTITATIVE COMPARISON RESULTS ON THE VAIHINGEN TEST SET WITH OTHER NETWORKS

Method	Imp surf.	Building	Low veg.	Tree	Car	mF1	OA	mIoU
FCN [31]	88.56	92.43	80.66	87.92	70.35	83.98	86.56	72.92
BiseNet [31]	89.23	91.15	80.55	85.28	72.87	83.82	86.48	73.56
Deeplabv3+ [38]	92.13	94.36	83.57	89.43	86.88	89.29	90.49	81.33
PSPNet [63]	92.67	94.89	84.32	89.54	87.45	89.77	90.76	81.98
DANet [64]	91.09	94.22	83.28	88.78	86.79	88.83	90.14	80.93
BoTNet [65]	91.76	92.07	81.64	88.76	71.24	84.13	87.63	74.23
BANet [66]	92.26	94.88	83.65	90.01	85.71	89.30	90.23	81.46
Segmenter [47]	89.64	92.68	81.47	89.13	68.35	84.25	88.24	73.83
UNetFormer [67]	93.14	95.71	85.44	90.38	89.17	90.77	91.34	84.19
IDRNet [68]	91.64	94.93	83.61	90.14	87.46	89.56	90.79	82.04
SfNet [69]	93.16	95.75	84.38	90.29	88.31	90.38	91.08	83.34
SGFNet [70]	93.21	95.49	84.23	89.91	85.29	89.63	90.88	82.94
BDNet [71]	93.09	95.93	85.07	90.01	88.79	90.58	91.19	83.83
FBRNet [72]	92.81	96.29	85.28	89.72	86.39	90.10	90.67	83.03
PEEN	**94.09**	**96.77**	**85.91**	**90.74**	**90.63**	**91.63**	**92.83**	**85.54**

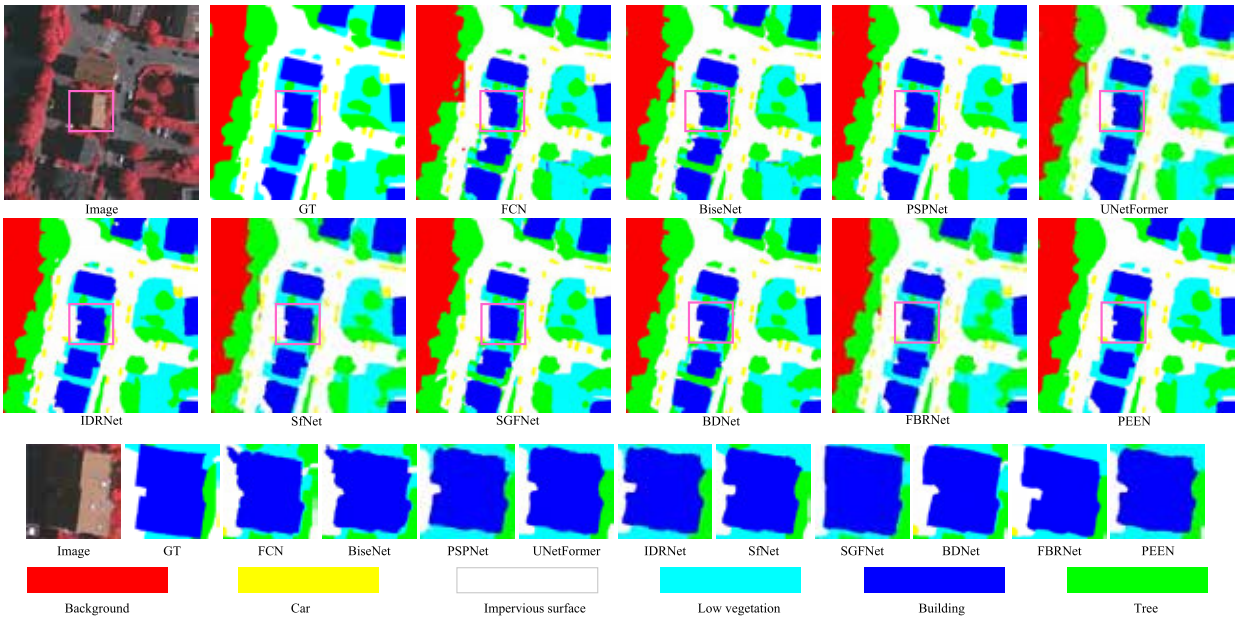

Fig. 7. Enlarged visualization of results on the Vaihingen dataset.

the next iteration layer to provide prior information for producing the next prediction result. Finally, after a certain number of iterations, the segmentation results are reinforced. Besides, due to the ambiguity and uncertainty of RSI edge pixels, it is impossible to accurately predict RSI edge pixels with only one PF. In the PEEN model, we employ a series of PFs, which rectify the problem of incorrect edge pixel coverage encountered by individual PFs. Specifically, we obtain a set of PFs by controlling the hyperparameter α according to 4. In the IM, the first iterative α_1 is equal to 1 ($\alpha_1 = 1$), and the ith iterative $\alpha_i = \alpha_{i-1} + k$, the value of k represents the spacing between the values of α. In detail, the value of α_i in the ith iteration is calculated by

$$\alpha_i = k * (i-1) + 1; i \in [1, n], k \in [2, \infty) \tag{10}$$

where n is the number of iterations, k is the step of the α value, and i indicates the ith iteration. In our experiments, the value of α always starts from 1.

D. Loss Function

In this work, we employ the mean-squared error (mse) [73] constraint on pixel classification across the image and adopt the Dice coefficient loss function [74] to enhance the classification of edge pixels. The loss for the ith iteration is the sum of the mse and the Dice coefficient loss. The loss function is defined as follows:

$$\text{Loss}_{ai} = \frac{1}{\Omega} \sum_{p \in \Omega} \|\text{GT}(x), P_i(x)\|_2 \tag{11}$$

TABLE II
QUANTITATIVE COMPARISON RESULTS ON THE POTSDAM TEST SET WITH OTHER NETWORKS

Method	Imp surf.	Building	Low veg.	Tree	Car	mF1	OA	mIOU
FCN [31]	89.57	95.13	84.23	83.88	81.26	86.81	87.68	78.58
BiseNet [31]	90.24	93.61	85.23	85.89	92.16	89.43	89.46	81.26
Deeplabv3+ [38]	92.47	95.19	87.35	87.89	96.25	91.83	90.36	83.57
PSPNet [63]	92.69	96.87	87.76	88.43	94.96	92.14	90.89	84.01
DANet [64]	91.62	95.96	86.01	87.57	85.99	89.43	89.57	81.24
BoTNet [65]	92.33	96.14	86.78	88.61	93.88	91.55	90.36	84.67
BANet [66]	93.02	96.33	87.35	87.86	95.67	92.05	90.89	85.14
Segmenter [47]	91.29	94.88	85.47	85.12	88.67	88.49	88.12	80.27
UNetFormer [67]	93.55	96.89	87.15	88.92	**96.47**	92.60	91.19	87.17
IDRNet [68]	92.46	95.68	86.16	84.29	95.36	90.79	90.14	84.58
SfNet [69]	92.87	96.48	87.79	89.03	96.45	92.52	91.03	86.39
SGFNet [70]	93.41	97.13	88.23	88.74	95.39	92.62	91.02	83.94
BDNet [71]	93.29	96.84	87.45	89.04	95.35	92.39	91.08	85.63
FBRNet [72]	92.47	95.13	87.46	88.61	95.19	91.77	90.87	83.58
PEEN	**93.72**	**97.31**	**88.33**	**89.22**	96.39	**92.99**	**92.14**	**88.35**

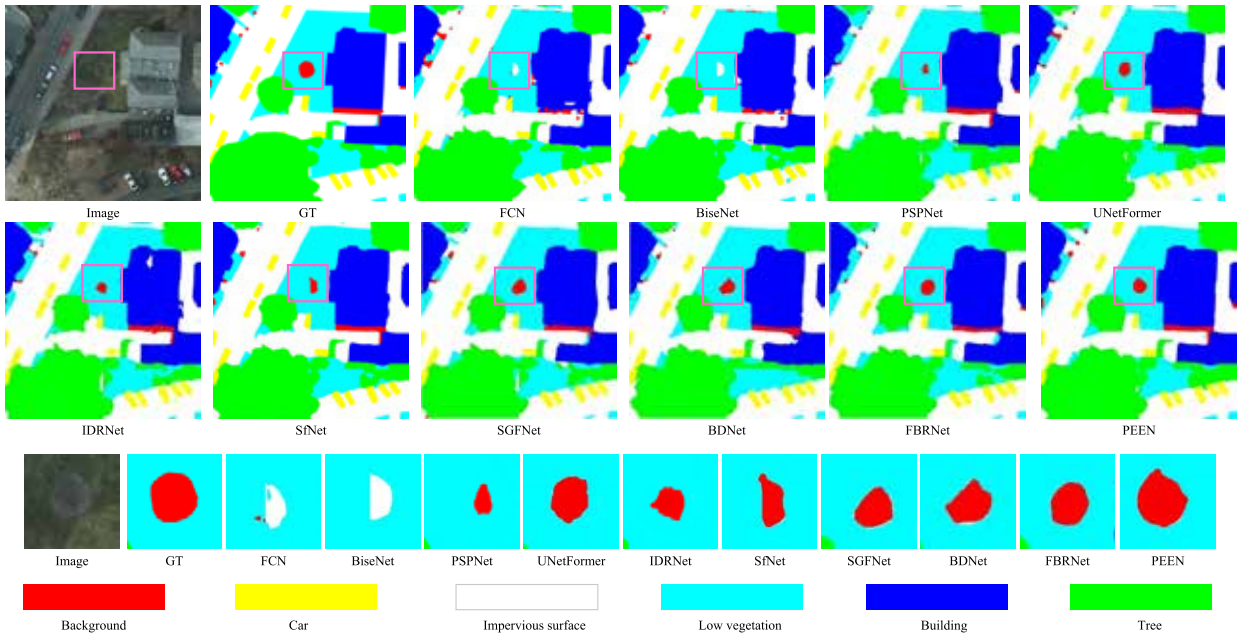

Fig. 8. Enlarged visualization of results on the Potsdam dataset.

$$\text{Loss}_{bi} = 1 - \frac{2 * \left(\text{pre}_{bi} \cap G_b\right)}{\left|\text{pre}_{bi}\right| + \left|G_b\right|} \tag{12}$$

$$\text{Loss}_i = \text{Loss}_{ai} + \text{Loss}_{bi} \tag{13}$$

where GT is the ground truth, P_i is the ith entire predicted result, x denotes pixel in the image domain Ω, pre_{bi} is the ith edge prediction output, and G_b is the edge label generated by the Canny operator. Loss_{ai} is the mse loss for the ith iteration, Loss_{bi} is the Dice coefficient loss for the ith iteration, and Loss_i is the total loss for the ith iteration.

Overall, the total loss L_s of the PEEN model is the sum of the total losses (loss_i) in each iteration, which can be calculated by the following formula:

$$L_s = \sum_{i=1}^{n} \text{loss}_i \tag{14}$$

where Loss_i is the loss of the ith iteration and n is the number of iterations.

IV. EXPERIMENTS

A. Datasets

1) Vaihingen: The Vaihingen dataset contains 33 images with an average size of 2494 × 2064 pixels and a ground sampling distance (GSD) of 9 cm. Each image contains three multispectral bands (near-infrared, red, and green) as well as a digital surface model (DSM) and a normalized digital surface model (NDSM). The dataset includes five foreground classes (impervious surfaces, buildings, low vegetation, trees, and cars) and one background class (clutter). Specifically, images numbered 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, and 33(12 images in total) are selected for testing, image number

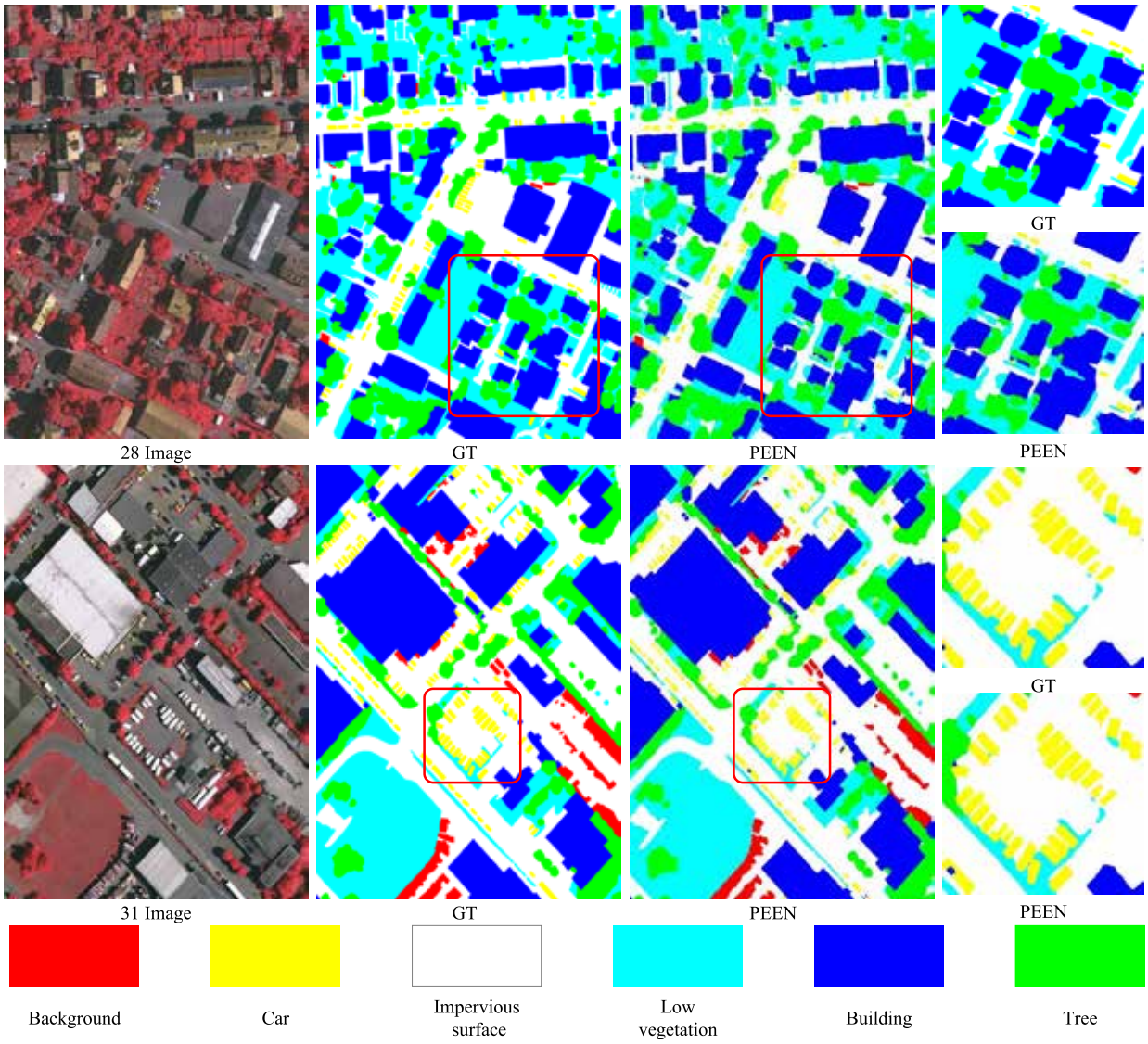

Fig. 9. Mapping results for the test images of Vaihingen numbered 28 and 31.

30 is applied for validation, and the remaining 20 images are utilized for training. The images are cropped into small patches of 256 × 256 pixels.

2) Potsdam: The Potsdam dataset contains 38 fine-resolution images of 6000 × 6000 pixels (GSD 5 cm) and the same category information as the Vaihingen dataset. The dataset provides three multispectral bands (red, green, blue, and near-infrared), as well as DSM and NDSM. In our experiments, images numbered 2_13, 2_14, 3_13, 3_14, 4_13, 4_14, 4_15, 5_13, 5_14, 5_15, 6_13, 6_14, 6_15, and 7_13 (14 images in total) are utilized for testing, image number 2_10 is employed for validation, and the remaining 22 images (excluding image 7_10 with incorrect annotations) are applied for training. Only the red, green, and blue bands are utilized in the experiment, and the original image blocks are cropped into small patches of 256 × 256 pixels.

B. Experimental Setting

The hardware conditions of the experiment were an AMD Ryzen Thread Ripper 3990X 64-Core Processor CPU, a main frequency of 2.90 GHz, a dynamic acceleration frequency of 4.3 GHz, 64GB of RAM, an NVIDIA Quadro RTX 8000 GPU, and 48 GB of graphics memory. The software environment adopts PyTorch 1.7 as the development framework and runs in Ubuntu 18.04 and Python 3.8 environments. For each method, the overall accuracy (OA), mean crossover ratio (mIoU), and $F1$-score ($F1$) are selected as evaluation indices.

C. Comparison With Other Methods

To verify the effectiveness of the proposed PEEN model, we compare our method with other classical approaches, including the FCN [31], BiseNet [75], DANet [64], Deeplabv3+ [38], PSPNet [63], BoTNet [65], BANet [66], Segmenter [47], UNetFormer [67], IDRNet [68], and SfNet [69]. Additionally, we evaluate our model with the edge-enhanced semantic segmentation networks, including the SGFNet [70], BDNet [71], and FBRNet [72].

1) Comparison With Other Methods on the Vaihingen Dataset: Table I reports the experimental quantitative comparison results on the Vaihingen dataset. The OA, mF1, and

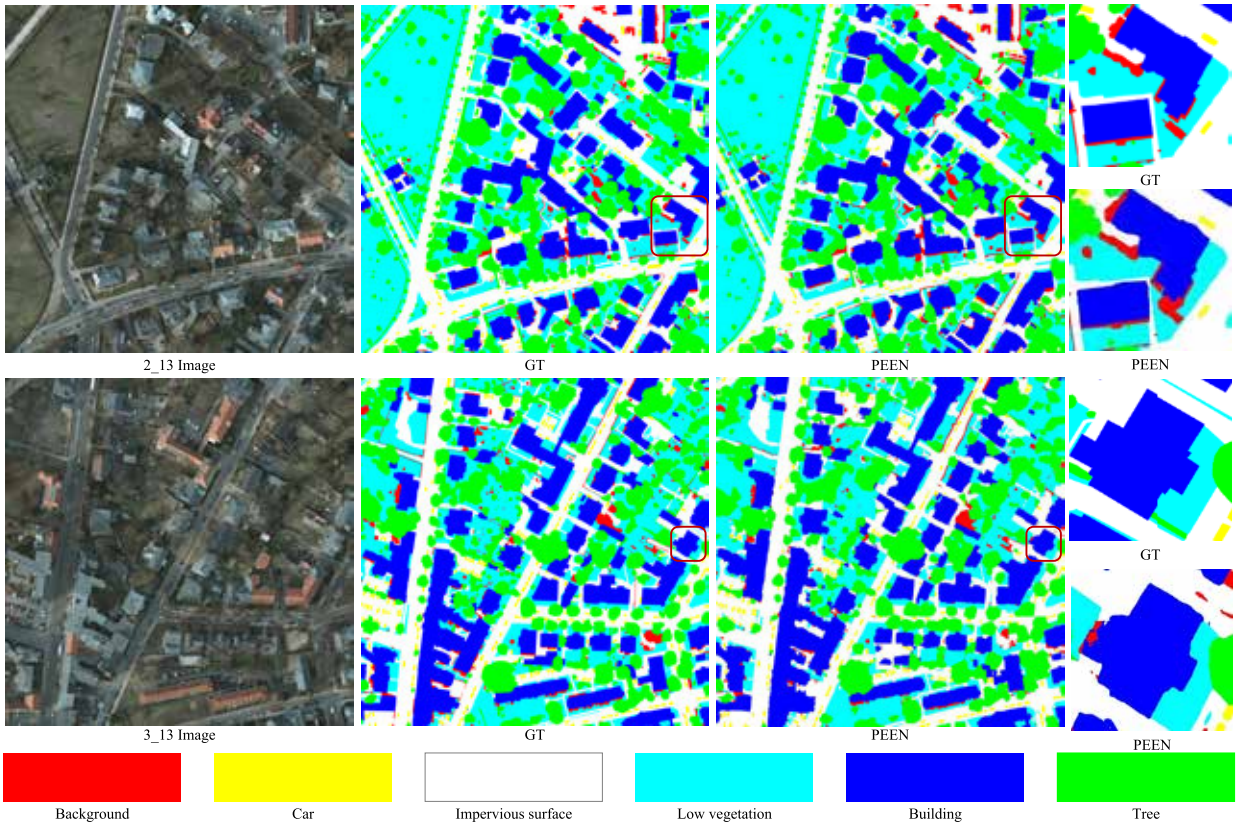

Fig. 10. Mapping results for the test images of Potsdam numbered 2_13 and 3_13.

mIoU of the proposed PEEN achieve the highest values of 91.63%, 92.83%, and 85.54%, respectively. Compared to the UNetFormer model, which has the highest $F1$-score for the car category among the comparison models, PEEN improves by 1.56%. Compared with the edge-enhancement-based models SGFNet, BDNet, and FBRNet, the mIoU of PEEN is higher than the three methods, which demonstrates the significant effectiveness of the PEEN model on small targets and edges.

Fig. 7 illustrates the visualization results on the Vaihingen dataset with the magnified segmentation details, which specifically highlights the building category in a red rectangle. As observed, the segmentation results of the PEEN model exhibit the closest alignment with GT, characterized by intact structural contours and sharply defined boundaries. The visualization results on the Potsdam dataset with the enlarged visualization of the segmentation results are shown in Fig. 8. The red box in the figure highlights partial regions of the background categories. As observed, the FCN and BiseNet classify the target as an impervious surface category. In contrast, the PEEN model accurately identifies the background class, which is the most complete and closest to the GT image.

2) Comparison With Other Methods on the Potsdam Dataset: For the Potsdam dataset, the compared results are shown in Table II. As can be observed, our PEEN model achieves the mean $F1$-score of 92.99% and an mIoU of 88.35%. Compared with the comparative models SGFNet, BDNet, and FBRNet based on edge enhancement, the PEEN model achieves improvements of at least 0.37%, 1.06%, and 2.72%, respectively. The results indicate that the proposed PEEN method demonstrates greater effectiveness than other edge-based approaches and achieves superior performance in blurred edges.

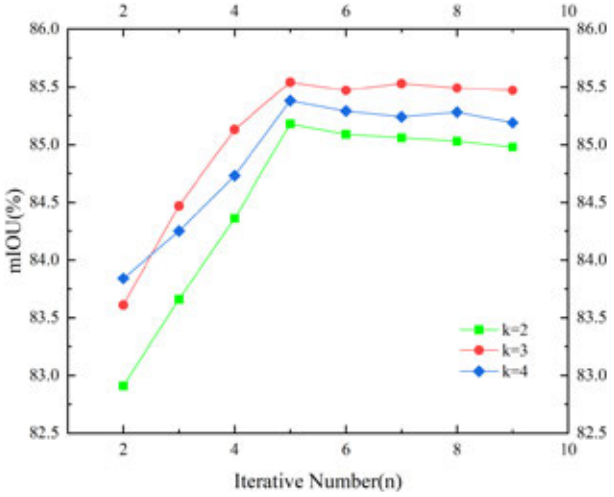

Fig. 11. Influence of different k and n values in the IM on mIoU.

Additionally, Fig. 9 presents full-image segmentation results for sample Nos. 28 and 31, which further illustrate the consistent performance across diverse scene contexts. All the segmentation result images validate the ability of PEEN to maintain effective segmentation on local edge details and global scene structure. Moreover, Fig. 10 presents full-image segmentation results for the samples of 2_13 and 3_13 of the

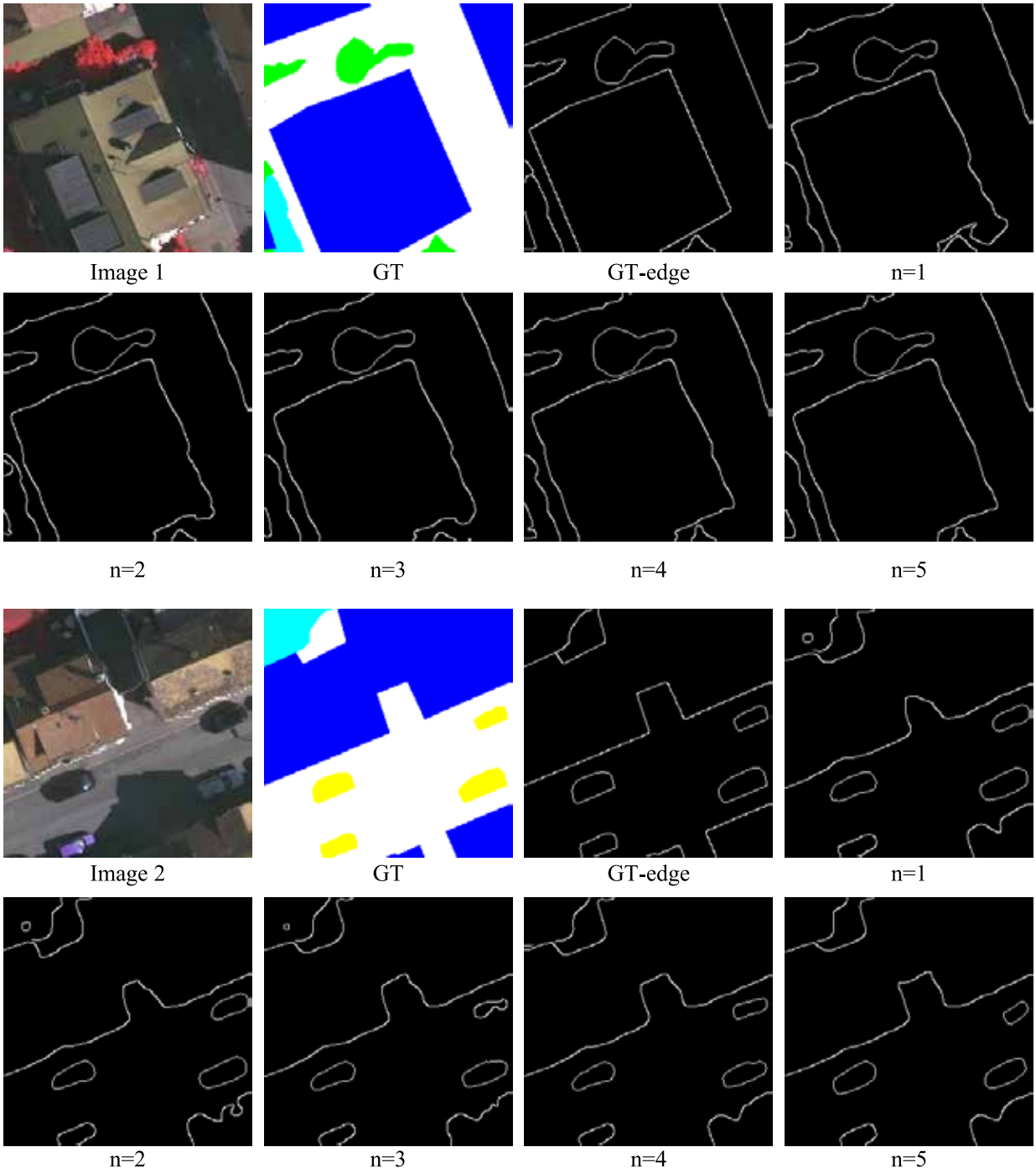

Fig. 12. Visualization of segmentation at edges during iteration.

Potsdam dataset. The qualitative results demonstrate that the PEEN model effectively segments both the local edge details and the complete structure of objects.

D. Ablation Study

We conducted ablation experiments to validate the effectiveness of the Conv SA mechanism and the IPEP module in our network. Notably, in the implementation without Conv SA, we replace it with traditional 3×3 convolutions. The experimental results are shown in Table III. From the data in the table, it can be observed that both proposed modules contribute to improvements in various evaluation metrics. Specifically, the incorporation of Conv SA leads to an improvement of 4.76% and 3.11% in mIoU values compared to utilizing standard 3×3 convolutions on the Vaihingen and Potsdam datasets, respectively. Furthermore, the combination of the IPEP module with the model enhanced mIOU values when either 3×3 convolution or Conv SA was employed. Consequently, both the long-range information extraction capability of Conv SA and the edge enhancement capability of the IPEP module benefit the PEEN model. Conv SA improves

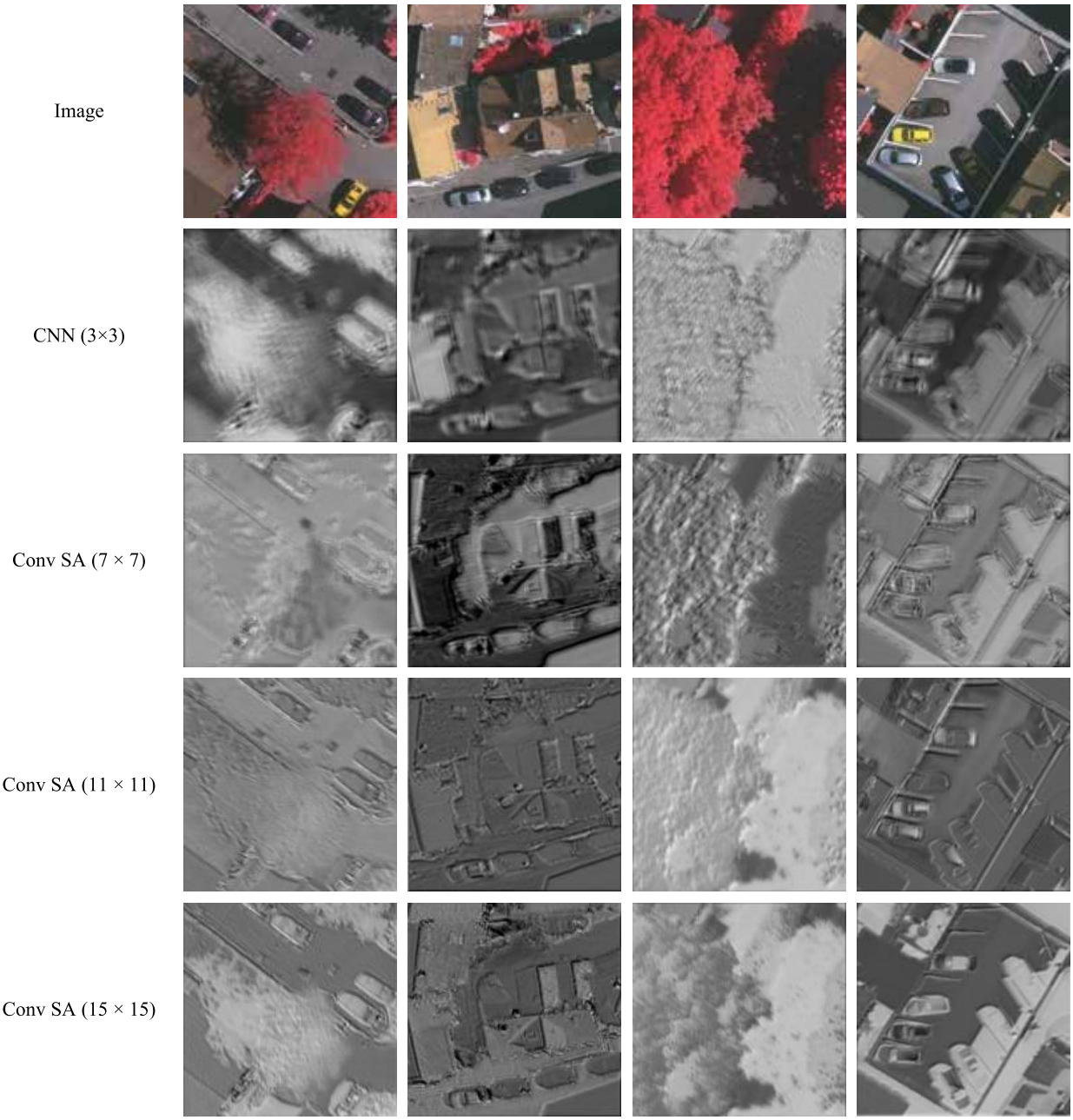

Fig. 13. Comparison of visualized feature maps extracted by a CNN with 3×3 convolutional kernels and Conv SA with different kernel sizes.

TABLE III
ABLATION STUDY OF EACH COMPONENT OF THE MODEL

Dataset	Method	mIoU
Vaihingen	CNN	78.01
	Conv SA	82.77
	CNN + IPEP	83.49
	Conv SA + IPEP	85.54
Potsdam	CNN	82.72
	Conv SA	85.83
	CNN + IPEP	86.29
	Conv SA + IPEP	88.35

the segmentation of occluded objects, while the IPEP module strengthens edge segmentation effectiveness.

E. Parameter Optimization and Experimental Analysis

1) Influence of Different α and n values in the IM: In this section, we conduct experiments to examine the impact of different values of α during iterations and explore the relationship between semantic segmentation performance and the number of iterations n. The experiments are conducted on the Vaihingen dataset without any additional pretraining. During training, we randomly cropped regions of the RSI and resized them to 256×256. The PEEN model is trained for 200 iterations with a batch size of 8. Fig. 11 presents some representative experimental results. It is evident that the mIoU value consistently increases with an increasing n. The mIOU value reaches the peak and starts to stabilize around $n = 5$. Additionally, we observe that the approach with the setting of $k = 3$ yielded the best performance after

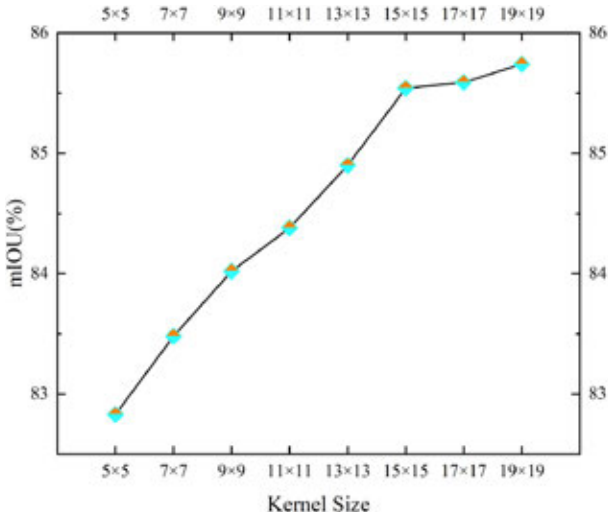

Fig. 14. Influence of convolution kernel size in Conv SA on mIoU.

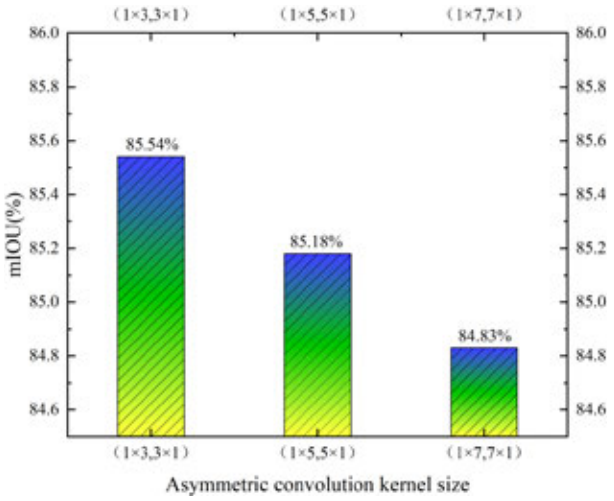

Fig. 15. Influence of the size of asymmetric convolution kernel in the IM on mIoU.

five iterations. Consequently, for all subsequent experiments, we set the number of iterations for the IM to 5 and the corresponding k value to 3. The α values were selected from the set $\{1, 4, 7, 10, 13\}$. Additionally, as depicted in Fig. 12, we visualize the details of local edges during the iterative process; it is obvious that there is a gradual improvement in edge accuracy with increasing iterations.

2) Influence of the Kernel Size in Conv SA: The Conv SA employs variable-sized convolutional kernels. In this section, we analyze the impact of different kernel sizes on the segmentation results. The experiments are conducted on the Vaihingen dataset without any additional pretraining. During training, we randomly crop the RSI regions and resize them to 256×256 and then train the model for 200 epochs with a batch size of 8. Fig. 13 illustrates a comparative visualization of local feature maps. Our Conv SA exhibits superior feature extraction capabilities compared to traditional 3×3 convolutions. Particularly, when utilizing a 15×15 kernel size, the extracted feature maps demonstrate the highest level of clarity and resolution, while also preserving a greater amount of fine-grained details. As illustrated in Fig. 14, the mIOU value of the PEEN model demonstrates an almost linear increase as the kernel size expands. Nevertheless, after reaching a kernel size of 15, further increasing the kernel size does not significantly improve the mIOU value. Therefore, for the sake of model efficiency, the kernel size is set to 15×15 for all other experiments in this article. Furthermore, we conduct visualization experiments to compare the feature maps extracted by the Conv SA mechanism with different kernel sizes (7×7, 11×11, and 15×15) with those extracted by a CNN containing 3×3 convolutional kernels.

3) Influence of Asymmetric Convolution Kernel Size: In this section, we compare the effect of different convolution kernel sizes for asymmetric convolution in IM on the performance of the PEEN model. The compared convolutional kernel combinations are $(1 \times 3, 3 \times 1)$, $(1 \times 5, 5 \times 1)$, and $(1 \times 7, 7 \times 1)$. The experiments are conducted on the Vaihingen dataset without any additional pretraining, and the results are shown in Fig. 15. As shown, the PEEN model achieves an mIoU value of 85.54%, which is higher than the set of $(1 \times 5, 5 \times 1)$ combination and the set of $(1 \times 7, 7 \times 1)$ combination. In the other experiments, the asymmetric convolutional kernels are configured by the default setting of $(1 \times 3, 3 \times 1)$.

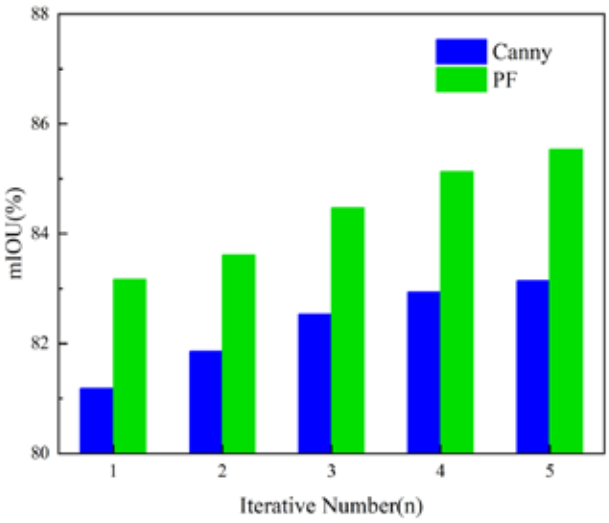

Fig. 16. Influence of utilizing PFs or Canny in the IPEP module on mIoU.

4) Effectiveness of PF: During the training stage in this article, we calculate the difference between the predicted edges and the actual edges in the GT images. Since the edges of the GT images are straightforward to compute, we can theoretically use any edge detection operator, such as the Canny operator, the Sobel operator, or PF, to achieve similar results. In our implementation, we opted for the simple, yet effective Canny operator in the IPEP module.

To verify the effectiveness of PF and the Canny operator in edge prediction, we replace PF with the Canny operator to conduct experiments in this section. The experiments are conducted on the Vaihingen dataset without any additional pretraining. The experimental results are shown in Fig. 16. We evaluate the performance utilizing mIoU values and compare the results obtained after applying a single iteration of

TABLE IV
COMPARISON RESULTS OF PEEN AND COMPARISON MODELS IN COMPLEXITY AND INFERENCE SPEED

Method	Parameters (M)	Speed (FPS)	Vaihingen (mIoU)	Potsdam (mIoU)
FCN [31]	40.35	74.15	72.92	78.58
BiseNet [75]	14.02	**132.45**	73.56	81.26
Deeplabv3+ [38]	45.38	60.49	81.33	83.57
PSPNet [63]	65.47	65.53	81.98	84.01
DANet [64]	38.47	78.91	80.93	81.24
BoTNet [65]	21.36	87.15	74.23	84.67
BANet [66]	12.47	114.56	81.46	85.14
Segmenter [47]	**7.54**	15.34	73.83	80.24
UNetFormer [67]	11.84	114.26	84.19	87.17
IDRNet [68]	24.15	125.78	82.04	84.58
SfNet [69]	54.78	98.31	83.34	86.39
SGFNet [70]	50.84	90.26	82.94	83.94
BDNet [71]	24.15	83.78	83.83	85.63
FBRNet [72]	34.22	86.31	83.03	83.58
PEEN	50.47	108.56	**85.54**	**88.35**

TABLE V
COMPARISON RESULTS BETWEEN CONV SA AND OTHER COMMON ATTENTION MECHANISMS

Attention	Imp surf.	Building	Low veg.	Tree	Car	mF1	OA	mIoU
SE [76]	91.89	94.21	83.68	87.45	87.93	89.03	90.63	82.72
CBAM [77]	90.83	94.39	82.35	89.33	89.01	91.49	89.48	83.56
CA [78]	90.69	93.27	81.89	88.48	88.96	90.89	89.33	83.01
GAM [79]	90.19	94.12	82.28	89.78	87.79	91.14	88.83	83.31
EMA [80]	91.14	93.73	83.44	88.31	87.67	91.34	89.01	83.89
Conv SA	**94.09**	**96.77**	**85.91**	**90.74**	**90.63**	**91.63**	**92.83**	**85.54**

reinforcement learning with PF and Canny operators. Overall, the mIoU values exhibit a positive correlation with the iteration count n in both methods. Nevertheless, the mIoU values obtained with the Canny operator are significantly lower than those achieved with a single application of PF. Even after performing five iterations employing the Canny operator, the mIoU value is not as high as that obtained with PF after a single iteration. Specifically, with one iteration, the mIoU value obtained with PF reaches 83.14%, while that obtained with Canny is 81.18%. After five iterations employing Canny, the mIoU value increases to 83.14%, which is still lower than the mIoU value obtained with PF after one iteration. Overall, our PF improves the accuracy of edge segmentation by converting pixel-edge distances into probabilities of pixels belonging to edges, using different α values in each iteration to reinforce this process. Compared to the Canny algorithm, which detects edges using pixel thresholds, our approach is clearly more effective.

5) Comparison of Network Efficiency: In this section, we compare the network efficiency of the PEEN and the comparison networks in terms of mIoU, parameters, and speed. The experiments are performed on the Vaihingen and Potsdam datasets, respectively. The results are shown in Table IV. As observed, with an input image size of 256×256, the PEEN model achieves a parameter count of 50.47 M and an inference speed of 108.56 FPS, which is generally comparable to other nonlightweight models. Additionally, our model demonstrates superior segmentation performance, achieving mIoU values of 85.54% on the Vaihingen dataset and 88.35% on the Potsdam dataset.

6) Comparison of Conv SA With Other Common Attention Mechanisms: To verify the effectiveness of Conv SA, a comprehensive comparison was conducted with other common attention modules in this section. The compared attention mechanisms included SE [76], CBAM [77], CA [78], GAM [79], and EMA [80]. The experiments were carried out based on the Vaihingen dataset, and the experimental setups were all the same. The experimental results are presented in Table V, and the bolded values indicate the optimal experimental results. As observed, our proposed mechanism achieved superior performance compared to other approaches with the mF1 of 91.63%, OA of 92.83%, and mIoU of 85.54%, respectively. The experimental results convincingly illustrate that Conv SA is valuable to incorporate spatial context information into RSIs during the information acquisition process. The mentioned integration effectively promotes the crucial feature representation and alleviates the challenges associated with edges and small target recognition.

V. CONCLUSION

In this article, we propose a novel idea that maps the distance between pixels and edges to probabilities and constructs an encoder–decoder structure PEEN model for efficient SSRSIs. Specifically, since global context information is crucial for occluded object segmentation, we designed the Conv

SA mechanism to extract long-distance-dependent information in RSIs. For enhancing the segmentation performance at the edges, we propose the IPEP module that utilizes PF to iteratively enhance the segmentation at the boundaries of the target, ultimately achieving precise results for SSRSIs. A comprehensive set of comparative studies, such as ablation and parameter optimization studies on the ISPRS Vaihingen and Potsdam datasets, demonstrated the effectiveness and efficiency of the proposed network for SSRSIs.

In the future, we plan to establish efficient deep learning architectures for the SSRSI to reduce computational complexity and memory requirements. With the lightweight implementation, the semantic segmentation models are ideally implemented in resource-constrained environments, that is, mobile devices or remote computing platforms. Additionally, by providing timely decision support, lightweight semantic segmentation networks are essential for real-time remote sensing monitoring, which has substantial applications in the fields of agricultural monitoring, environmental protection, and urban planning.

REFERENCES

[1] L. Zhang and L. Zhang, "Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 2, pp. 270–294, Jun. 2022.

[2] C. Yu, Y. Zhu, Y. Wang, E. Zhao, Q. Zhang, and X. Lu, "Concern with center-pixel labeling: Center-specific perception transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 5514614.

[3] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2229–2235.

[4] A. Alzu'bi and L. Alsmadi, "Monitoring deforestation in Jordan using deep semantic segmentation with satellite imagery," *Ecological Informat.*, vol. 70, Sep. 2022, Art. no. 101745.

[5] B. Neupane, T. Horanont, and J. Aryal, "Deep learning-based semantic segmentation of urban features in satellite images: A review and meta-analysis," *Remote Sens.*, vol. 13, no. 4, p. 808, Feb. 2021.

[6] G. Libessart, C. Franck-Néel, P. Branchu, and C. Schwartz, "The human factor of pedogenesis described by historical trajectories of land use: The case of Paris," *Landscape Urban Planning*, vol. 222, Jun. 2022, Art. no. 104393.

[7] Y. Pi, N. D. Nath, and A. H. Behzadan, "Detection and semantic segmentation of disaster damage in UAV footage," *J. Comput. Civil Eng.*, vol. 35, no. 2, Mar. 2021, Art. no. 04020063.

[8] T. Chowdhury, M. Rahnemoonfar, R. Murphy, and O. Fernandes, "Comprehensive semantic segmentation on high resolution UAV imagery for natural disaster damage assessment," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 3904–3913.

[9] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, Sep. 2020.

[10] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022.

[11] K. A. García-Pardo, D. Moreno-Rangel, S. Domínguez-Amarillo, and J. R. García-Chávez, "Remote sensing for the assessment of ecosystem services provided by urban vegetation: A review of the methods applied," *Urban Forestry Urban Greening*, vol. 74, Aug. 2022, Art. no. 127636.

[12] C. M. Viana, S. Oliveira, S. C. Oliveira, and J. Rocha, "Land use/land cover change detection and urban sprawl analysis," in *Spatial Modeling in GIS and R for Earth and Environmental Sciences*, H. R. Pourghasemi and C. Gokceoglu, Eds., Amsterdam, The Netherlands: Elsevier, 2019, pp. 621–651.

[13] M. Krichen, M. S. Abdalzaher, M. Elwekeil, and M. M. Fouda, "Managing natural disasters: An analysis of technological advancements, opportunities, and challenges," *Internet Things Cyber-Phys. Syst.*, vol. 4, pp. 99–109, Jan. 2024.

[14] S. A. Hojjatoleslami and J. Kittler, "Region growing: A new approach," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1079–1084, Jul. 1998.

[15] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N–D image segmentation," *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, Nov. 2006.

[16] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 1, pp. 25–39, Jan. 1983.

[17] J. Wu, "Introduction to convolutional neural networks," *Nat. Key Lab Novel Softw. Technol.*, vol. 5, no. 23, p. 495, 2017.

[18] X. Wang, Z. Hu, S. Shi, M. Hou, L. Xu, and X. Zhang, "A deep learning method for optimizing semantic segmentation accuracy of remote sensing images based on improved UNet," *Sci. Rep.*, vol. 13, no. 1, p. 7600, May 2023.

[19] Q. Zeng, J. Zhou, J. Tao, L. Chen, X. Niu, and Y. Zhang, "Multiscale global context network for semantic segmentation of high-resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5622913.

[20] Y. Su, L. Gao, A. Plaza, X. Sun, M. Jiang, and G. Yang, "SRViT: Self-supervised relation-aware vision transformer for hyperspectral unmixing," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 2025, doi: 10.1109/TNNLS.2025.3571798.

[21] X. Li et al., "A synergistical attention model for semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5400916.

[22] Y. Liu, Y. Zhang, Y. Wang, and S. Mei, "Rethinking transformers for semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5617515.

[23] P.-T. Jiang, Y. Yang, Q. Hou, and Y. Wei, "L2G: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16886–16896.

[24] X. Li et al., "Improving semantic segmentation via decoupled body and edge supervision," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K. Cham, Switzerland: Springer, 2020, pp. 435–452.

[25] 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 J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, Jan. 2018.

[26] H. Jung, H.-S. Choi, and M. Kang, "Boundary enhancement semantic segmentation for building extraction from remote sensed image," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5215512.

[27] W. Rong, Z. Li, W. Zhang, and L. Sun, "An improved Canny edge detection algorithm," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Aug. 2014, pp. 577–582.

[28] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved Sobel edge detection," in *Proc. 3rd Int. Conf. Comput. Sci. Inf. Technol.*, Jul. 2010, pp. 67–71.

[29] X. Ma, X. Zhang, and M.-O. Pun, "A crossmodal multiscale fusion network for semantic segmentation of remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 3463–3474, 2022.

[30] A. Abdollahi and B. Pradhan, "Integrating semantic edges and segmentation information for building extraction from aerial images using UNet," *Mach. Learn. Appl.*, vol. 6, Dec. 2021, Art. no. 100194.

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[32] R. Kemker, C. Salvaggio, and C. Kanan, "Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 60–77, Nov. 2018.

[33] I. Kotaridis and M. Lazaridou, "Remote sensing image segmentation advances: A meta-analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 309–322, Mar. 2021.

[34] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogramm. Remote Sens.*, vol. 152, pp. 166–177, Jun. 2019.

[35] X.-Y. Tong et al., "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111322.

[36] Y. Hua, D. Marcos, L. Mou, X. X. Zhu, and D. Tuia, "Semantic segmentation of remote sensing images with sparse annotations," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.

[38] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[39] Y. Sun, Y. Tian, and Y. Xu, "Problems of encoder–decoder frameworks for high-resolution remote sensing image segmentation: Structural stereotype and insufficient learning," *Neurocomputing*, vol. 330, pp. 297–304, Feb. 2019.

[40] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "ResUNet-A: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS J. Photogramm. Remote Sens.*, vol. 162, pp. 94–114, Apr. 2020.

[41] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "TreeUNet: Adaptive tree convolutional neural networks for subdecimeter aerial image segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 156, pp. 1–13, Oct. 2019.

[42] Z. Zhou, M. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. Int. Workshop Deep Learn. Med. Image Anal.*, vol. 11045. Cham, Switzerland: Springer, 2018, pp. 3–11.

[43] Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, and C. Pan, "Semantic labeling in very high resolution images via a self-cascaded convolutional neural network," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 78–95, Nov. 2018.

[44] W. Zhao, S. Du, Q. Wang, and W. J. Emery, "Contextually guided very-high-resolution imagery classification with semantic segments," *ISPRS J. Photogramm. Remote Sens.*, vol. 132, pp. 48–60, Oct. 2017.

[45] Y. Shen, J. Chen, L. Xiao, and D. Pan, "Optimizing multiscale segmentation with local spectral heterogeneity measure for high resolution remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 157, pp. 13–25, Nov. 2019.

[46] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[47] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7262–7272.

[48] E. Xie et al., "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Sys. (NIPS)*, vol. 34, Dec. 2021, pp. 12077–12090.

[49] H. Cao et al., "Swin-Unet: Unet-like pure transformer for medical image segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Jan. 2021, pp. 205–218.

[50] Q. Wang, X. Jin, Q. Jiang, L. Wu, Y. Zhang, and W. Zhou, "DBCT-Net: A dual branch hybrid CNN-transformer network for remote sensing image fusion," *Expert Syst. Appl.*, vol. 233, Dec. 2023, Art. no. 120829.

[51] Y. Liu et al., "A transformer-based multi-modal fusion network for semantic segmentation of high-resolution remote sensing imagery," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 133, Sep. 2024, Art. no. 104083.

[52] W. Miao, Z. Xu, J. Geng, and W. Jiang, "ECAE: Edge-aware class activation enhancement for semisupervised remote sensing image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5625014.

[53] B. Sui, Y. Cao, X. Bai, S. Zhang, and R. Wu, "BIBED-seg: Block-in-block edge detection network for guiding semantic segmentation task of high-resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1531–1549, 2023.

[54] J. Wu, C. Qin, Y. Ren, and G. Feng, "EPFNet: Edge-prototype fusion network toward few-shot semantic segmentation for aerial remote-sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[55] X. Sun, A. Shi, H. Huang, and H. Mayer, "BAS^{4}net: Boundary-aware semi-supervised semantic segmentation network for very high resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5398–5413, 2020.

[56] A. Li, L. Jiao, H. Zhu, L. Li, and F. Liu, "Multitask semantic boundary awareness network for remote sensing image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5400314.

[57] X. Sun, M. Xia, and T. Dai, "Controllable fused semantic segmentation with adaptive edge loss for remote sensing parsing," *Remote Sens.*, vol. 14, no. 1, p. 207, Jan. 2022.

[58] C. He, S. Li, D. Xiong, P. Fang, and M. Liao, "Remote sensing image semantic segmentation based on edge information guidance," *Remote Sens.*, vol. 12, no. 9, p. 1501, May 2020.

[59] G. P. H. Styan, "Hadamard products and multivariate statistical analysis," *Linear Algebra Appl.*, vol. 6, pp. 217–240, Jan. 1973.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[61] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jul. 2017, pp. 5987–5995.

[62] C.-H. Tsai, Y.-T. Chih, W. H. Wong, and C.-Y. Lee, "A hardware-efficient sigmoid function with adjustable precision for a neural network system," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 11, pp. 1073–1077, Nov. 2015.

[63] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jul. 2017, pp. 6230–6239.

[64] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149.

[65] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Los Alamitos, CA, USA, Jun. 2021, pp. 16514–16524.

[66] L. Wang, R. Li, D. Wang, C. Duan, T. Wang, and X. Meng, "Transformer meets convolution: A bilateral awareness network for semantic segmentation of very fine resolution urban scene images," *Remote Sens.*, vol. 13, no. 16, p. 3065, Aug. 2021.

[67] L. Wang et al., "UNetFormer: A UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 190, pp. 196–214, Aug. 2022.

[68] Z. Jin, X. Hu, L. Zhu, L. Song, Y. Li, and L. Yu, "IDRNet: Intervention-driven relation network for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2023, pp. 51606–51620.

[69] X. Li et al., "Sfnet: Faster and accurate semantic segmentation via semantic flow," *Int. J. Comput. Vis.*, vol. 132, no. 2, pp. 466–489, Feb. 2024.

[70] Y. Wang, G. Li, and Z. Liu, "SGFNet: Semantic-guided fusion network for RGB-thermal semantic segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 12, pp. 7737–7748, Dec. 2023.

[71] X. Li, L. Xie, C. Wang, J. Miao, H. Shen, and L. Zhang, "Boundary-enhanced dual-stream network for semantic segmentation of high-resolution remote sensing images," *GIScience Remote Sens.*, vol. 61, no. 1, Dec. 2024, Art. no. 2356355.

[72] S. Qu, Z. Wang, J. Wu, and Y. Feng, "FBRNet: A feature fusion and border refinement network for real-time semantic segmentation," *Pattern Anal. Appl.*, vol. 27, no. 1, p. 2, Mar. 2024.

[73] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not," *Geoscientific Model Develop.*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022.

[74] F. Milletari, N. Navab, and S. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

[75] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiseNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 325–341.

[76] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[77] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 3–19.

[78] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13713–13722.

[79] Y. Liu, Z. Shao, and N. Hoffmann, "Global attention mechanism: Retain information to enhance channel-spatial interactions," 2021, *arXiv:2112.05561*.

[80] D. Ouyang et al., "Efficient multi-scale attention module with cross-spatial learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

Chunyan Yu (Senior Member, IEEE) received the Ph.D. degree in environmental engineering from Dalian Maritime University, Dalian, China, in 2012.

She is currently an Associate Professor at the Information Science and Technology College, Dalian Maritime University. Her research interests include image segmentation, hyperspectral image classification, and pattern recognition.

Yakun Zuo received the bachelor's degree in electronic information science and technology from Henan Agricultural University, Zhengzhou, China, in 2020. He is currently pursuing the master's degree in software engineering at Dalian Maritime University, Dalian, China.

His research interests include remote sensing image processing and deep learning.

Qiang Zhang (Member, IEEE) received the B.E. degree in surveying and mapping engineering and the M.E. and Ph.D. degrees in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2017, 2019, and 2022, respectively.

He is currently an Associate Professor with the Center of Hyperspectral Imaging in Remote Sensing (CHIRS), Information Science and Technology College, Dalian Maritime University, Dalian, China. He has authored more than ten journal articles in IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), Earth System Science Data, and ISPRS Journal of Photogrammetry and Remote Sensing. His research interests include remote sensing information processing, computer vision, and machine learning. More details could be found at https://qzhang95.github.io

Yulei Wang (Member, IEEE) received the B.S. and Ph.D. degrees in signal and information processing from Harbin Engineering University, Harbin, China, in 2009 and 2015, respectively.

In 2011, she was awarded by China Scholarship Council to study at the Remote Sensing Signal and Image Processing Laboratory, University of Maryland, Baltimore, MD, USA, as a joint Ph.D. Student for two years. She is an Associate Professor with the Hyperspectral Imaging in Remote Sensing (CHIRS), Information Science and Technology College, Dalian Maritime University, Dalian, China. Her research interests include hyperspectral image processing and vital signs signal processing.