DHRNet: A Dual-Branch Hybrid Reinforcement Network for Semantic Segmentation of Remote Sensing Images

Qinyan Bai, Xiaobo Luo, Yaxu Wang, and Tengfei Wei

***Abstract*—In the field of remote sensing image processing, semantic segmentation has always been a hot research topic. Currently, deep convolutional neural networks (DCNNs) are the mainstream methods for the semantic segmentation of remote sensing image (RSI). There are two commonly used semantic segmentation methods based on DCNNs: multiscale feature extraction based on deep-level features, and global modeling. The former can better extract object features of different scales in complex scenes. However, this method lacks sufficient spatial information, resulting in poor edge segmentation ability. The latter can effectively solve the problem of limited receptive field in DCNNs obtaining more comprehensive feature extraction results. Unfortunately, this method is prone to misclassification, resulting in incorrect predictions of local pixels. To address these issues, we propose the dual-branch hybrid reinforcement network (DHRNet) for more precise semantic segmentation of RSI. This model is a dual-branch parallel structure with a multiscale feature extraction branch and a global context and detail enhancement branch. This structure decomposes the complex semantic segmentation task, allowing each branch to extract features with different emphases while retaining sufficient spatial information. The results of both branches are fused to obtain a more comprehensive segmentation result. After conducting extensive experiments on three publicly available RSI datasets, ISPRS Potsdam, ISPRS Vaihingen, and LoveDA, DHRNet demonstrates excellent results with the mean intersection over union of 86.97%, 83.53%, and 54.48% on the three datasets, respectively.**

***Index Terms*—Global context modeling, multiscale feature extraction, remote sensing, semantic segmentation.**

I. Introduction

REMOTE sensing technology [1] is an essential technique widely applied in various fields. With the rapid development of remote sensing imaging technology, very high-resolution remote sensing image (RSI) [2], [3] can be easily acquired. Semantic segmentation of RSI [4] is a research hotspot and finds practical applications in various tasks, such as urban planning [5], land cover mapping [6], change detection [7], [8], building and road extraction [9], [10], [11], vegetation extraction [12], and water body extraction [13], [14]. Semantic segmentation of RSI refers to the classification of each pixel in RSI, which is a dense classification task. When facing many very high-resolution RSI, fine semantic segmentation tasks still remain highly challenging. In the past, image segmentation was often based on features such as gray scale [15], color [16], spatial texture [17], [18], [19], geometry, and shape [20]. Subsequently, some works have proposed classical unsupervised learning algorithms such as FCM [21] and watershed [22]. Soon after, with the rapid growth in machine learning, more and better algorithms have been proposed, including support vector machines [23], Markov random fields [24], maximum likelihood [25], conditional random fields [26], and random forests [27]. However, most of these algorithms require manual preprocessing, postprocessing, and feature engineering, which can be complicated for nonprofessionals.

In recent years, deep convolutional neural networks (DCNNs) [28] have demonstrated significant superiority in image processing. They process images in an end-to-end [29] manner without the need for manual intervention, simplifying the operation for nonspecialists. DCNNs have shown superior performance across various fields. With the proposal of fully convolutional network (FCN) [30], a historic breakthrough was made in the semantic segmentation task. FCN presents a simple end-to-end approach for the semantic segmentation of images, with accuracy surpassing other concurrent methods, laying the foundation for the later development of DCNNs in the field of semantic segmentation. However, FCN simultaneously exposed two limitations of DCNNs: limited receptive field [31] and lack of spatial information. Subsequently, U-net [32], which was proposed, fuses the feature information by concatenating the low-level features and high-level features layer by layer through upsampling, restoring the lost spatial information during the downsampling layers. Unsatisfactorily, this approach weakens the contribution of deep-level features to the final segmentation result, and indiscriminately introduces low-level features which also introduces the noise information carried by them, affecting the final segmentation

Manuscript received 22 July 2023; revised 5 October 2023 and 3 December 2023; accepted 18 January 2024. Date of publication 23 January 2024; date of current version 8 February 2024. This work was supported in part by the Major Industrial Technology Research and Development Projects of high-tech industry in Chongqing under Grant D2018-82, in part by the Intergovernmental International Scientific and Technological Innovation Cooperation Project of the National key R & D Program under Grant 2021YFE0194700, in part by the key cooperation project of Chongqing Municipal Education Commission under Grant HZ2021008. *(Corresponding author: Xiaobo Luo.)*

The authors are with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, and also with the Chongqing Engineering Research Center of Spatial Big Data Intelligent Technology, Chongqing 400065, China (e-mail: s220231002@stu.cqupt.edu.cn; luoxb@cqupt.edu.cn; wyx960128@163.com; s210201105@stu.cqupt.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3357216

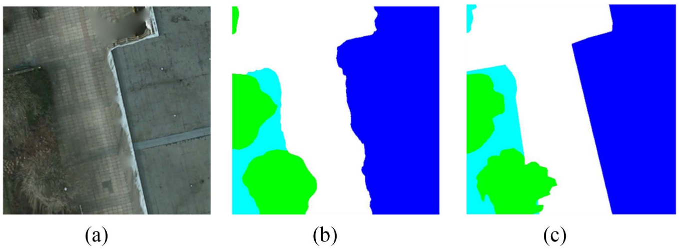

Fig. 1. Example of edge information loss caused by multiscale module. (a) Original image. (b) Segmentation result based on the multiscale model. (c) Ground truth.

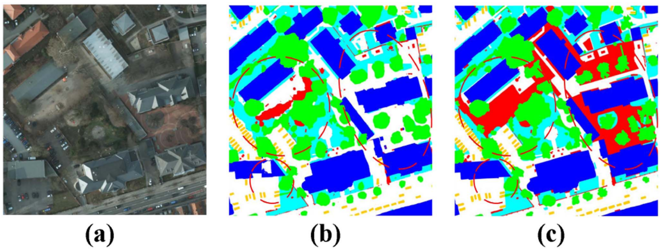

Fig. 2. Illustration of misclassification caused by the self-attention mechanism. (a) Original image. (b) Segmentation result from the self-attention mechanism based model. (c) Ground truth.

result. This is not conducive to segmentation tasks involving multiple categories and large-scale objects. FPN [33] extracts image pyramid feature maps and performs multiscale feature fusion through element-wise addition. Due to differences in feature scales between different layers, they may not align at the pixel level. Therefore, element-wise addition directly may result in negative impacts. Also, similar to U-net, this approach reduces the significance of deep-level features in contributing to the final segmentation result. As for DeepLabv3+ [34], it utilizes dilated convolutions with different dilation rates to obtain multiscale information. However, excessive use of dilated convolution may bring about gridding effect. Worse still, DeepLabv3+ connects and fuses deep-level features with only one low-level feature, leading to the loss of detailed spatial information (an illustration example of this phenomenon is given in Fig. 1), which is disadvantageous for tasks requiring precise edges such as building extraction.

Due to the constraints of convolution kernels, the contextual information that DCNNs can accept is limited. Contrastively, models based on attention mechanisms are capable of providing a global receptive field. As a remedy, to further address the issue of receptive field, many works have introduced attention mechanism into semantic segmentation tasks for global contextual modeling. DANet [35] embeds both spatial and channel attention mechanisms to enhance its ability to extract global context information. Nonlocal neural network [36] incorporates self-attention mechanism into semantic segmentation tasks to obtain global semantic information. The experimental results of these works have demonstrated that the attention mechanism can effectively alleviate the limited receptive field issue of DCNNs to some extent. However, modules based on self-attention mechanisms often demand significant computational resources, which is highly unfavorable for designing lightweight modules. Therefore, many studies have conducted research on simplifying the attention mechanism. For example, the COAT [37] reduces the complexity of the attention mechanism to linear through factorization, and the MANet [38] maps the self-attention mechanism using the method of kernel functions, also reducing its complexity to linear. These methods lighten the cost of using self-attention modules, allowing more lightweight networks to use self-attention mechanisms for global context modeling, thereby improving the final segmentation accuracy. Insufficiently, the use of attention mechanism also brings another problem. The self-attention mechanism possesses a strong global modeling capability, enabling each pixel in the feature map to acquire global context information. However, the self-attention module typically begins with randomly initialized weights and it does not have the strong inductive bias [39] of convolution. The weight distribution of the attention matrix obtained in the early training stages is not ideal, and the model requires longer training time to converge. Therefore, models based on self-attention mechanism require longer training time and larger training dataset compared to CNNs to achieve better segmentation results. For some smaller datasets or during the early stages of training, the self-attention module may generate incorrect attention matrix weight distribution due to insufficient training samples, leading to misclassification in the final results (Fig. 2 gives an illustration of this phenomenon).

To address the issues in current models based on multiscale and global modeling methods and obtain better segmentation results, we propose dual-branch hybrid reinforcement network (DHRNet), a dual-branch parallel network. The main purpose of the network design is to achieve higher segmentation accuracy with a minimal number of model parameters and computational complexity. Our main contributions are as follows.

1) Proposing a novel multiscale feature extraction branch (MFEB), which eliminates redundant channels through a channel selection module (CSM), providing the most suitable feature maps for each multiscale branch while reducing the model's parameter. Furthermore, three sets of strip convolutions with varying sizes are utilized to extract multiscale features of objects. Finally, deformable convolutions are employed to further fit the real shape of objects, enhancing the accuracy of feature extraction results.
2) Presenting a novel global context and detail enhancement branch (GEB), utilizing a cross-layer attention module (CAM) to enhance the model's global modeling capabilities, reduce lower level feature noise, and enhance the effectiveness of multilayer feature fusion. In addition, a class ratio extraction module (CREM) is employed to supervise this branch, accelerating the model's convergence speed and yielding smoother prediction results.
3) Introducing a novel lightweight end-to-end network called DHRNet to process segmentation tasks with different focus through a dual-branch parallel architecture. We compared our method with other widely used methods on

three public datasets and achieved excellent results with a parameter size of only 6.6 M.

The rest of this article is organized as follows. Section II presents the related works. Section III describes the proposed network. Section IV gives the experimental details. Section V concludes this article.

II. Related Works

In this section, we reviewed some works related to DHRNet and discussed some limitations of the existing work.

A. Encoder–Decoder Architecture

The encoder–decoder architecture [40] has been widely applied in the field of semantic segmentation since the advent of FCN. In the DCNNs based encoder–decoder architecture, the encoder serves as the backbone feature extraction network, using stacked convolution layers and downsampling layers to extract features of the image to be segmented. The decoder processes the features obtained from the encoder to obtain deep-level features rich in semantic information. The output result of the encoder is restored to the original image resolution in the decoder through relevant upsampling techniques. Finally, the pixel-level object classification task is completed through the dense classification layer. In U-net, a symmetrical "encoder–decoder" architecture design was proposed, which obtains more spatial information by successively upsampling deep-level features and merging with the features from each layer's output of the encoder. SegNet [41] adopts the pooling indices method to recover the spatial information lost in downsampling layers. In these structures, the ability of the encoder to extract features determines the quality of the features obtained by the decoder, and the degree of the decoder utilizing the features extracted by the encoder determines the quality of the final segmentation results. The advantage of the encoder–decoder structure is that the decoder can obtain more spatial information to optimize the final results by combining the features of different layers in the encoder. However, this structure has also introduced some issues. First, deep-level and low-level features may not be aligned, whereas using an upsampling method with element-wise addition may have a negative impact on spatial information. Second, this structure ignores global context information, which is not ideal for segmenting large-scale objects. Furthermore, low-level features often contain more noise. So, directly incorporating low-level features may introduce their noise and have a negative impact on the final prediction result. Finally, in this layer-by-layer upsampling process, the contribution of deep-level features to the final prediction result is actually being weakened. Considering that the rich semantic information is contained in deep-level features, this structure may not effectively harness this information. Consequently, it may not be well-suited for addressing complex scene segmentation problems.

B. Multiscale Architecture

In many images, the size of the objects to be segmented is not the same, especially in RSI, where the scenes are often more complex and the target size differences are huge. It is often difficult to extract or segment all targets through a single scale. Therefore, many works have proposed multiscale feature extraction methods to optimize the final segmentation results. PSPNet [42] adopts spatial pyramid pooling to obtain multiscale features, which not only merges the context information of different scales but also improves the expression of global feature information. However, introducing too much global average pooling also loses more spatial and edge information, and the effect is often not satisfactory when faced with tasks that require fine edge segmentation. DeepLabV3+ extracts multiscale features through dilated convolution with different dilation rates and has a strong segmentation ability for complex scenes and multicategory tasks. However, DeepLabv3+ connects and fuses deep-level features with only one low-level feature, lacking sufficient spatial information, and therefore lacks sufficient object edge segmentation ability. Moreover, deep-level features have huge differences from low-level features after going through its multiscale module, and direct fusion cannot effectively improve the final segmentation results. Although the multibranch dilated convolution used in its multiscale module atrous spatial pyramid pooling (ASPP) can obtain different-sized receptive fields while reducing the number of parameters, it may also bring about gridding effects, resulting in the loss of local information and a decrease in the correlation of distant information [43].

C. Global Context Information

The self-attention module is the core component of the transformer model [44]. Originally, the transformer was used to solve sequence-to-sequence machine translation tasks in the natural language processing field and is widely used and improved. Subsequent studies have shown that this model can be transferred to computer vision tasks [45]. The VIT model [46] only uses the encoder of the transformer to construct a network model for image classification, and it can be transferred to other downstream vision tasks. Afterward, a series of models based on the improvement of VIT (such as DeiT [47], PVT [48], Swin-transformer [49], MPVIT [50], etc.) achieved excellent results in a large number of tasks such as image classification [51], semantic segmentation, human pose estimation [52], and object detection [53]. Due to the self-attention mechanism's ability to aggregate global context information, which is lacking in DCNNs, many subsequent works have combined this mechanism with DCNNs to improve the final prediction performance of the model. For example, the nonlocal module [36] is an improved plug-and-play module based on the self-attention mechanism, designed to enhance the global receptive field. Although the attention mechanism can aggregate global context information, it also brings a significant increase in computational cost, with its computational complexity proportional to the square of the spatial resolution. For images with high spatial resolution, especially for RSI, directly using such a module would result in a significant additional computational cost. Therefore, many subsequent works have proposed improved attention formulas to reduce the enormous computational cost increase brought by the attention mechanism. In the COAT model, the authors

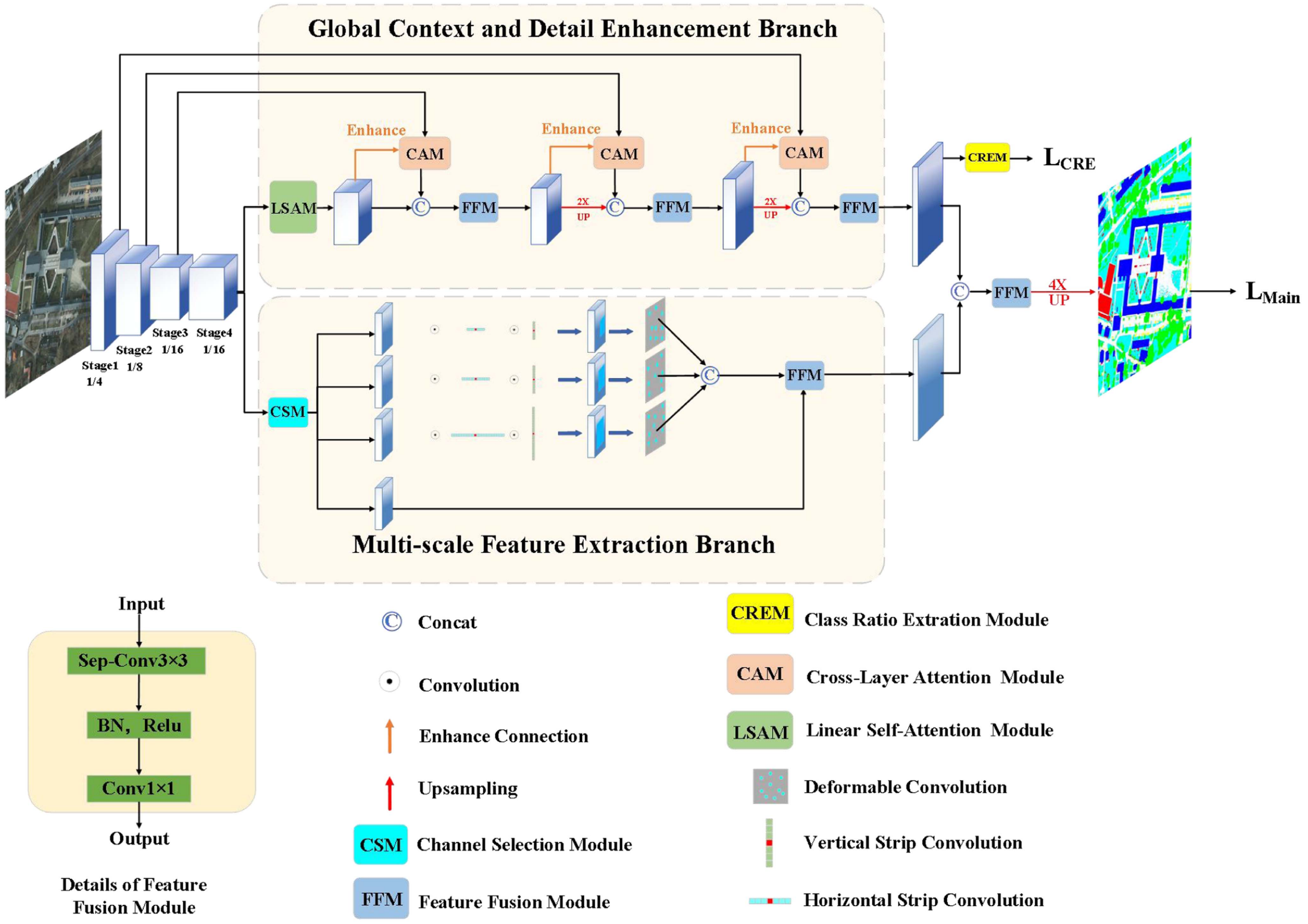
Global Context and Detail Enhancement Branch
Enhance
CAM
Enhance
CAM
Enhance
CAM
LSAM
FFM
2X UP
FFM
2X UP
FFM
CREM
L_CRE
FFM
4X UP
L_Main
Stage1 1/4
Stage2 1/8
Stage3 1/16
Stage4 1/16
CSM
FFM
Multi-scale Feature Extraction Branch
Input
Sep-Conv3×3
BN, Relu
Conv1×1
Output
Details of Feature Fusion Module
Concat
Convolution
Enhance Connection
Upsampling
CSM Channel Selection Module
FFM Feature Fusion Module
CREM Class Ratio Extration Module
CAM Cross-Layer Attention Module
LSAM Linear Self-Attention Module
Deformable Convolution
Vertical Strip Convolution
Horizontal Strip Convolution

Fig. 3. Overall architecture of the DHRNet.

used two mapping functions to factorize the original formula of attention mechanism, reducing the original $O(N^2)$ complexity to linear $O(N)$. SENet [54] constructed a simple and lightweight attention module by using max pooling downsampling and fully connected layers. CBAM [55] combined spatial attention and channel attention in a lightweight manner. These works have made it possible to apply the attention mechanism in lightweight models. Another problem is self-attention mechanism lacks the strong inductive biases of convolution, resulting in slow convergence speed and difficulty in obtaining satisfactory results with a scarcity of training data. Though approaches have been proposed to accelerate the convergence speed of self-attention-based models using model distillation techniques [47], those approaches also introduce computational overhead.

D. Design of Convolution Kernel

In prior works, the receptive field is often enlarged by stacking convolutional layers. As a remedy, many works split one large convolution kernel into several smaller ones to reduce the number of parameters and speed up the computation efficiency. However, recent work [56] has shown that such methods may have some problems. Some works have proposed the concept of “effective receptive field” [57], which represents the actual effect size of the receptive field of the model. According to the theory of effective receptive field, the size of the receptive field is proportional to the size of the convolution kernel and proportional to the square root of the number of convolutional layers. Increasing the receptive field by directly enlarging the convolution kernel is more effective than adding the depth of the convolutional layer. Through comparative experiments, it was found that although the theoretical receptive field size of the model remains consistent after decomposing the large convolution kernel into several small ones, the actual effective receptive field is reduced. This indicates that several small convolution kernels cannot completely replace the role of large ones.

III. Methodology

A. Overall Architecture of DHRNet

The overall architecture of our proposed DHRNet shown in Fig. 3 is a network composed of two branches: GEB and MFEB. This architecture separates the originally complex semantic segmentation task, allowing each branch to perform focused feature extraction tasks. MFEB emphasizes the scale differences of objects and has a stronger segmentation ability for multicategories and complex scenes. GEB contains rich spatial information and stronger global modeling ability, resulting in

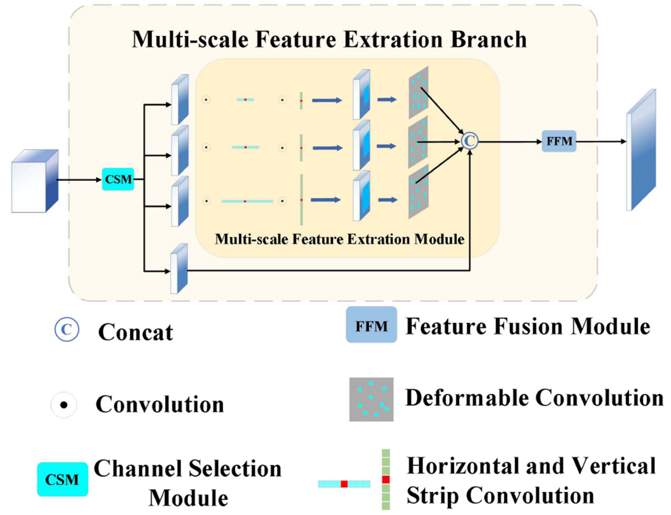
Multi-scale Feature Extration Branch
CSM
FFM
Multi-scale Feature Extration Module
© Concat
FFM Feature Fusion Module
• Convolution
Deformable Convolution
CSM Channel Selection Module
Horizontal and Vertical Strip Convolution

Fig. 4. Overall architecture of the MFEB.

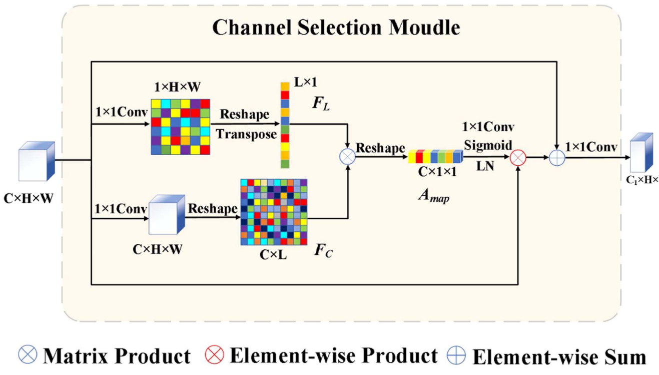

Fig. 5. Overall architecture of the CSM.

segmentation results with higher edge and global accuracy. The use of a parallel architecture also avoids the problem of a sequential architecture in which the contribution of a preceding module is weakened by subsequent modules. Finally, the results of the two parallel branches are fused at a higher resolution, ensuring that both branches contribute equally to the final result.

B. Multiscale Feature Extraction Branch

To enhance the multiscale feature extraction ability of the proposed model, we designed the MFEB. The overall architecture of this branch is illustrated in Fig. 4. This branch consists of two parts: a CSM and a multiscale feature extraction module (MFEM). Then, We will provide design rationales and detailed explanations for these two modules.

1) Channel Selection Module: To reduce the model's parameter count, we designed the CSM to choose the most suitable feature map channels for each branch in the MFEM, and eliminate redundant information. The inspiration for the design of this module comes from self-attention mechanism. The key distinction from SENet is that the former derives channel attention scores through pooling downsampling and fully connected layers. Our proposed CSM calculates the final channel attention scores based on the feature maps themselves, and optimizes by training the mapping matrix. This approach allows for better integration with the feature maps themselves, resulting in improved selection outcomes.

Fig. 5 illustrates the detailed design of the CSM. Given a feature map $F_{\text{in}} \in R^{C\times H\times W}$, this module first reconstructs it into a matrix $F_L \in R^{L\times 1}(L = H \times W)$ through 1×1 convolution, transpose and reshape operations. Similarly, 1×1 convolution and reshape operations are utilized to map the feature map F_{in} into matrix $F_C \in R^{C\times L}$. Then, matrix multiplication is performed between F_L and F_C, followed by a reshape operation to obtain channel attention score vector $A_{\text{map}} \in R^{C\times 1\times 1}$. After that, 1×1 convolution, layer normalization, sigmoid operation, and element-wise multiplication is performed between the input feature map F_{in} and the obtained feature score vector A_{map} along the channel dimension to enhance certain channels, resulting in the enhanced feature map $E \in R^{C\times H\times W}$. Finally, the enhanced feature map E and the input feature map F_{in} are connected with a residual connection, and then a 1×1 convolution is used to perform channel scaling, resulting in the final feature map $X \in R^{C1\times H\times W}$. The resulting feature map X is the most suitable feature map selected by this module for the current branch, which has more relevant feature channels compared to the original input feature map F_{in}. The feature selection process can be described as follows:

$$A_{\text{map}} = \delta(LN(f_r(F_L \odot F_C) * f)) \tag{1}$$

$$X = ((A_{\text{map}} \otimes F_{\text{in}}) \oplus F_{\text{in}}) * f \tag{2}$$

wherein, F_L and F_C are two feature mapping matrices obtained through 1×1 convolution and reshape; $\odot$ denotes matrix multiplication; $*$ denotes convolution operation; f_r denotes reshape operation; f denotes 1×1 convolution kernel; LN denotes layer normalization; δ denotes sigmoid function; A_{map} denotes the score vector; F_{in} denotes input feature map; X denotes the output feature map.

2) Multiscale Feature Extraction Module: In previous works, the ASPP module was widely used and improved. This module performs multiscale feature extraction using dilated convolutions of different dilation rates. However, this method may cause gridding effect, resulting in the loss of local information and a decrease in the correlation of distant information [43]. Subsequently, some works [56] used large kernel convolutions instead of dilated convolution to avoid the potential problems of gridding effect and maintain the module's ability to extract multiscale features, but this method also increased the number of parameters and computational complexity. In our work, we designed a novel MFEM for multiscale feature extraction, which replaces large kernel convolutions with three groups of strip convolutions of varied sizes. This approach reduces the number of parameters and computational complexity for each branch. In addition, more flexible deformable convolutions are used in this module after each group of strip convolutions to accurately fit the real shape of the object.

As shown in Fig. 4, we employ a multibranch approach for multiscale feature extraction. For each branch, the input feature

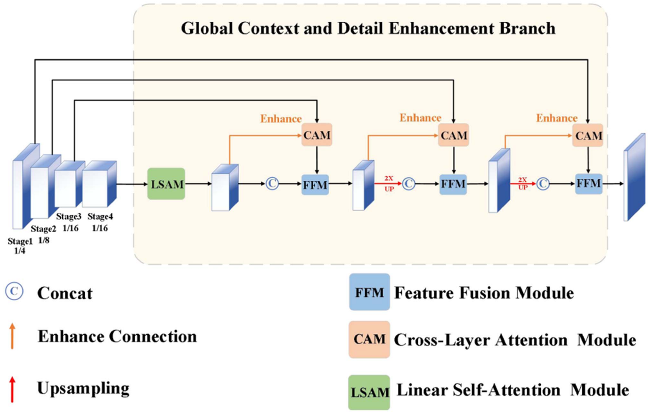
Global Context and Detail Enhancement Branch
Stage1 1/4
Stage2 1/8
Stage3 1/16
Stage4 1/16
LSAM
Enhance
CAM
FFM
2x UP
© Concat
Enhance Connection
Upsampling
FFM Feature Fusion Module
CAM Cross-Layer Attention Module
LSAM Linear Self-Attention Module

Fig. 6. Overall architecture of the GEB.

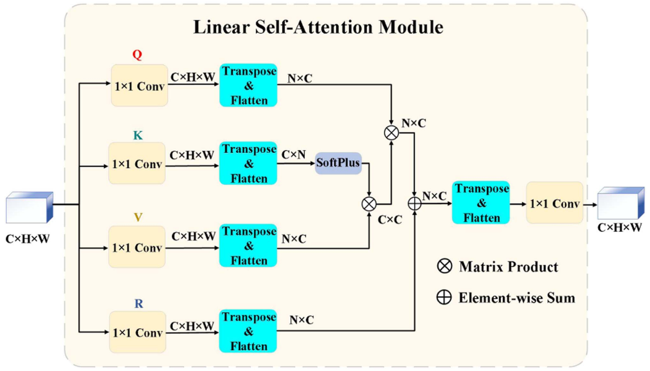

Fig. 7. Overall architecture of the LSM. Where Q, K, and V denote the query matrix, key matrix, and value matrix respectively, and R is the residual mapping matrix.

is $F_i(i = 1, 2, 3, 4) \in R^{C \times H \times W}$. MFEM uses three groups of strip convolution of different kernel sizes (7×1 and 1×7, 13×1 and $1 \times 13, 25 \times 1$ and 1×25) to extract features at different scales. Then, deformable convolution is used for more flexible shape fitting. Finally, the result feature maps of the three branches are concatenated and fused with the input feature map F_4 to obtain the final result. The entire process can be expressed as follows:

$$M_i = \sigma(\mathrm{BN}(F_i * f_v * f_h * f_d)) \tag{3}$$

$$F_{\text{out}} = \sigma(\mathrm{BN}(\mathrm{Concat}(M_1, M_2, M_3, F_4) * f)) \tag{4}$$

wherein, F_i ($i = 1, 2, 3, 4$) denotes the four output feature maps obtained through CSM; M_i ($i = 1, 2, 3$) denotes the feature map obtained after strip convolution and deformable convolution; $*$ denotes the convolution operation; f_v, f_h, and f_d denotes the vertical strip convolution, horizontal strip convolution, and deformable convolution, respectively; BN denotes batch normalization; σ denotes the ReLU activation function; f denotes the 1×1 convolution kernel; F_{out} denotes the final output feature map, and Concat denotes concatenating features along the channel dimension.

C. Global Context and Detail Enhancement Branch

The overall structure of the GEB is shown in Fig. 6. In this branch, we adopt the U-Net architecture as the basic structure, which can effectively combine low-level and deep-level features, enabling the model to obtain rich semantic information while retaining sufficient spatial information. In this branch, feature fusion module serves as a transitional module, used for feature fusion after concatenation of different layers. The module first employs a depth-wise separable convolution, which extracts spatial features on each channel. Subsequently, a point-wise convolution is performed to integrate features from different channels. This approach is more suitable for fusing features from multiple layers, and compared to regular convolutions, it has fewer parameters and computational complexity, making it more suitable for lightweight model. In addition, we introduce a linear self-attention module (LSM) to enhance the global modeling capability of the network with lower complexity compared to self-attention module. To reduce noise carried by low-level features and enhance the fusion effect of feature maps, we propose a novel CAM. This module utilizes deep-level feature map to guide the low-level feature map to remove noise and increase the similarity between the two feature maps, thereby enhancing the effect of subsequent feature fusion. Then, we will provide a more detailed explanation of the two modules.

1) Linear Self-Attention Module: The self-attention module requires high computational complexity, typically O(N^2), where N is the total number of pixels in a single channel of the input feature map. This high computational complexity overhead is evidently unfriendly for designing a lightweight and efficient network. Thus, many works have proposed methods for reducing the computational complexity of the self-attention module or proposed new lightweight attention modules. In COAT, the authors use two mapping functions to factorize the original formula of the attention mechanism and reduce the original O(N^2) complexity to linear O(N). The module's effectiveness is equivalent to the original self-attention module. The attention formula in COAT is as follows:

$$\mathrm{FactorAtt}(Q, K, V) = \Phi(Q)(\Psi(K)^T V). \tag{5}$$

Inspired by the attention formula in COAT, our proposed linear attention formula uses the identity function to map Φ, and a softplus function is implemented to map ψ, which eliminates the impact of negative values and results in numbers that are greater than zero. The linear attention formula can be expressed as follows:

$$LA(Q, K, V) = \frac{Q}{\sqrt{C}}(\mathrm{softplus}(K)^T V). \tag{6}$$

Based on the expression of the linear attention formula, we further proposed the LSM, as shown in Fig. 7. For feature map $F_{\text{in}} \in R^{C \times H \times W}$, four 1×1 convolutions are used to map it into query matrix Q, key matrix K, value matrix V, and residual matrix R. Attention computation is performed using the linear attention formula on Q, K, and V, and the result is added element-wisely with the residual matrix R, then through a 1×1 convolution operation to map it back to the original feature space. The entire computation process of the linear self-attention formula can be

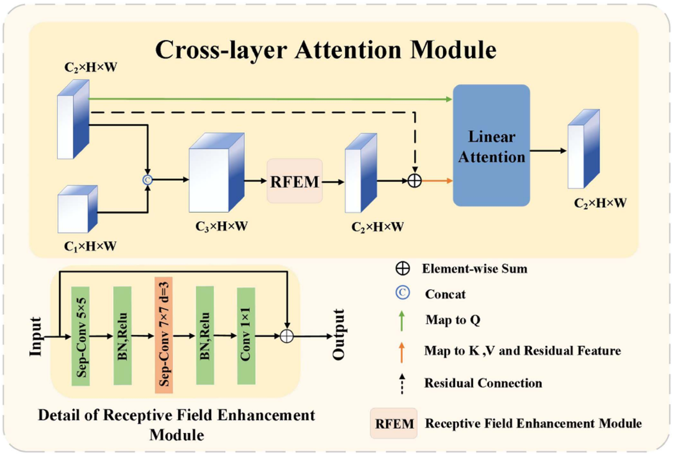

Fig. 8. Overall architecture of the CAM. Sep-Conv denotes the depth-wise separable convolution, and d denotes the dilation rate of the dilated convolution. Map to Q, K, and V denote mapping the feature as the query, key, and value matrices for the attention calculation.

described as follows:

$$\mathrm{LSA}(F_{\mathrm{in}}) = ((F_{\mathrm{in}} * f_1) \oplus LA(F_{\mathrm{in}} * f_2, F_{\mathrm{in}} * f_3, F_{\mathrm{in}} * f_4)) * f_5 \tag{7}$$

wherein, F_{in} denotes the input feature map; f_i (i = 1, 2, 3, 4, 5) is the convolution kernel with size 1 × 1; $*$ denotes the convolution operation; and $\oplus$ denotes the element-wise addition.

2) Cross-Layer Attention Module: Low-level features usually contain more noise, directly enhancing them with a self-attention module may not be the most appropriate method. In addition, in the process of layer-by-layer up-sampling similar to the U-net, the difference between high-level features and low-level features may be too large, and the enhancement brought by direct concatenation and fusion is limited. Therefore, we have designed a novel CAM to solve the above problems. This module eliminates noise from low-level features through the guidance of deep-level features, simultaneously enhancing the similarity among different layers of features, thus improving the effectiveness of feature fusion. The overall structure of CAM is shown in Fig. 8. The processes of this module are as follows: Initially, for the deep-level feature map $F_{\mathrm{high}} \in R^{C1 \times H \times W}$, its global context information is enhanced by the linear self-attention module. Then, the deep-level feature map F_{high} is concatenated with the low-level feature map $F_{\mathrm{low}} \in R^{C2 \times H \times W}$. For the concatenated feature map F_c, we employ a receptive field enhancement module (RFEM) for first feature fusion. This module utilizes two sets of large kernel convolutions for feature alignment, allowing the aligned features to better guide low-level features. Afterward, a residual connection is performed between the fused result and the low-level feature map F_{low} to obtain an enhanced feature map $F_{\mathrm{fusion}} \in R^{C2 \times H \times W}$ that contains more semantic information. Through the guidance of large kernel convolution in RFEM, the impact of some noise is also eliminated. Subsequently, the fused feature map F_{fusion} and the original low-level feature map F_{low} are used as inputs of the linear attention formula. In this module, the fused feature map F_{fusion} is mapped through a 1 × 1 convolution to form a query matrix Q, while the original low-level feature map F_{low} is mapped through two 1 × 1 convolutions to form the key matrix K and value matrix V. Through the linear attention formula, the final output is a feature map $F_m \in R^{C2 \times H \times W}$ that is more semantically similar to the deep-level feature map F_{high} and eliminates noise to some extent. Finally, the feature map F_m is concatenated and fused with the deep-level feature map F_{high}. Because of the high semantic similarity between the two feature maps, the effect of fusion has been further improved compared with before. The entire process can be expressed as follows:

$$F_{\mathrm{fusion}} = \mathrm{RFE}(\mathrm{Concat}(F_{\mathrm{high}}, F_{\mathrm{low}})) \oplus F_{\mathrm{low}} \tag{8}$$

$$\begin{aligned} &\mathrm{CA}(F_{\mathrm{fusion}}, F_{\mathrm{low}}) = ((F_{\mathrm{low}} * f_1) \oplus \\ &\mathrm{LA}(F_{\mathrm{fusion}} * f_2, F_{\mathrm{low}} * f_3, F_{\mathrm{low}} * f_4)) * f_5 \end{aligned} \tag{9}$$

$$F_{\mathrm{out}} = \sigma(\mathrm{BN}(\mathrm{Concat}(\mathrm{Up}(F_{\mathrm{high}}), \mathrm{CA}(F_{\mathrm{fusion}}, F_{\mathrm{low}})) * f)) \tag{10}$$

wherein, RFE denotes the receptive field enhancement module; $F_{\mathrm{high},}$ and F_{low} denote the deep-level and low-level feature maps, respectively; f_i (i = 1, 2, 3, 4, 5) and f denote 1 × 1 convolution kernel; $*$ denotes the convolution operation; $\oplus$ denotes element-wise addition; LA denotes the linear attention module; BN denotes batch normalization; σ denotes the ReLU activation function and Up denotes upsampling.

D. Class Ratio Extraction Module

Using attention mechanism can effectively improve the global modeling capability of the model, avoid negative impacts on the model's final performance caused by limited receptive fields, and enhance the generalization ability. What is more, many works have pointed out that another reason why attention mechanism can perform so well is that attention mechanism has fewer inductive biases [39] compared to convolution operation. Although these inductive biases can speed up the convergence speed of the model in the early training stages and achieve good results in small sample scenarios, in the later training stages, overly strong inductive biases will in turn limit its further optimization. Therefore, although attention-based models demonstrate excellent performance with sufficient training data and training time, they also suffer from slow training convergence speed, require larger training datasets and longer training times to fit properly. In scenarios where training data or time is limited, the effectiveness of attention mechanisms may not be ideal. To speed up the convergence and fitting speed of attention mechanism, we propose a novel CREM to supervise GEB. The details of the module are illustrated in Fig. 9. This module can make the output of the branch more similar to the ground truth label at the macro level, reducing the occurrence of misclassification in the early training stages.

In this module, a 1 × 1 convolution is employed to map feature map channels to the number of classification categories. Then, layer normalization and the ReLU activation function are used to suppress the influence of noise and negative number. Finally, we use another 1 × 1 convolution for the fusion of category information and global average pooling is applied to obtain the output vector, which stands for the proportion of each category in the

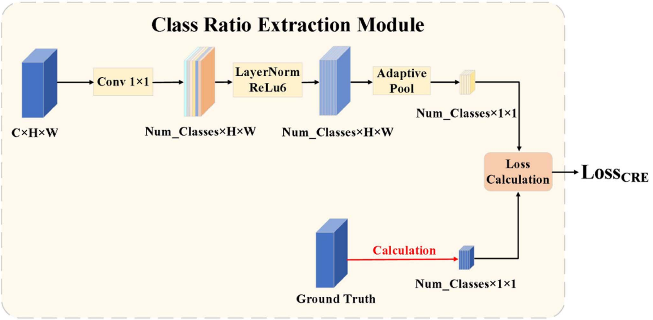

Fig. 9. Overall architecture of the CREM. Conv1 × 1-Down denotes the channel reduction performed through a 1 × 1 convolution operation. NUM_Classes refers to the final segmentation categories in the segmentation task.

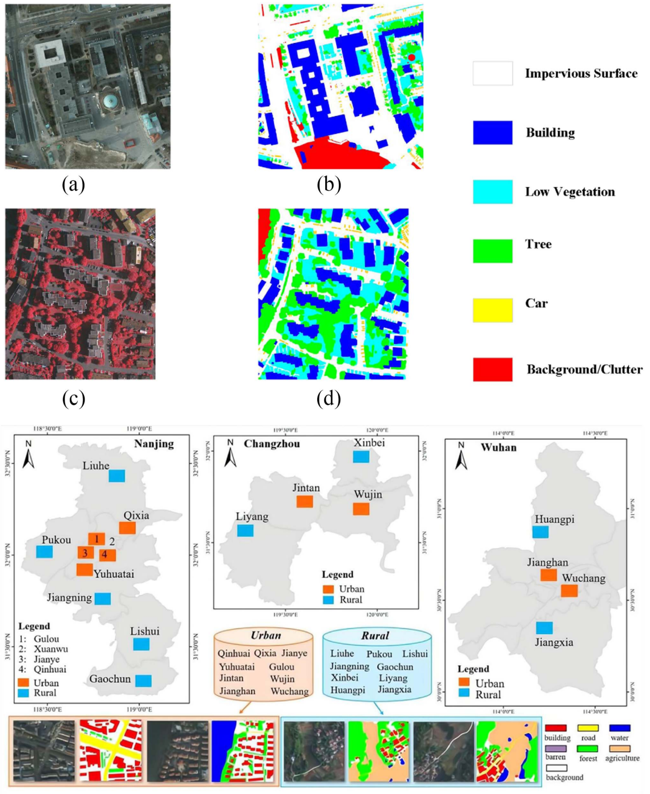

Fig. 10. Visualization of the original images and corresponding labels of Potsdam, Vaihingen, and LoveDA. (a) Postdam image. (b) Ground truth. (c) Vaihingen image. (d) Ground truth. (e) LoveDA dataset.

original image. Then, the binary cross-entropy loss is calculated between the module's output and the true category proportion to optimize GEB. It is worth noting that this module is only used in the training phase and will not affect the running speed during the prediction phase. Therefore, this module does not incur any additional computational overhead during inference phase. When coupled with a rational weight-loading strategy, it would not introduce any upsurge in the model's parameter count. The specific formula for the loss function can be expressed as follows:

$$M = \mathrm{Pool}_{\mathrm{avg}}(\sigma(\mathrm{LN}(F_{\mathrm{in}} * f_1)) * f_2) \qquad (11)$$

$$\mathrm{CRELoss} = -\sum_{i=1}^{N}(p(X_i)\log(\delta(M_i)) + (1-p(X_i))\log(1-\delta(M_i))) \qquad (12)$$

wherein, F_{in} denotes the input feature map; LN denotes layer normalization; σ denotes the ReLU activation function; $\mathrm{Pool}_{\mathrm{avg}}$ denotes global average pooling; f_1 and f_2 denote the 1 × 1 convolution kernel; $*$ denotes the convolution operation; M denotes the output proportion vector, and M_i denotes the value of the ith channel of M; δ denotes sigmoid function; N denotes the number of the classification categories, and $p(X)$ represents the proportion value of the ith category in the ground truth label.

E. Loss Function

Like most prior works, we employ the multiclass cross-entropy loss as the main loss function. It calculates the similarity between each pixel value of the network's output and the ground truth label, where a smaller loss indicates a closer similarity between the two distributions. Multiclass cross-entropy loss has been widely used in various dense prediction tasks, and can be expressed as follows:

$$\mathrm{CELoss} = -\sum_{i=1}^{H\times W}\sum_{j=1}^{N} p(X_{ij})\log(q(X_{ij})) \qquad (13)$$

wherein, H and W denote the spatial resolution of the input image, N denotes the number of categories, $q(X_{ij})$ denotes the probability that pixel I in the image is predicted to be of category j, and $p(X_{ij})$ denotes the true probability that pixel i in the image belongs to category j. To address the issue of imbalanced samples (i.e., when the frequency of occurrence for each category is not balanced), we use the dice loss as the auxiliary loss. The dice loss is a metric for evaluating the similarity between two samples, with a value ranging from zero to one. The larger the value is, the more similar the samples are. The dice loss can be expressed as follows:

$$\mathrm{DiceLoss} = 1 - \frac{2\,|X \cap Y| + \mathrm{Smooth}}{|X| + |Y| + \mathrm{Smooth}} \qquad (14)$$

wherein, Y denotes the pixel label of the ground truth image, and X denotes the pixel category of the model's predicted segmentation image. Smooth is the smoothing factor, and we set it to 1×10^{-5} in our experiments. Therefore, when using all modules, the final loss function of the proposed network is as follows:

$$\mathrm{Loss}_{\mathrm{total}} = \mathrm{CELoss} + \lambda\mathrm{DiceLoss} + \mu\mathrm{CRELoss}. \qquad (15)$$

λ, μ are the proportional coefficients, we take 0.7 and 0.3 in our experiments, respectively.

IV. Experiments

A. Datasets

We select ISPRS Potsdam, ISPRS Vaihingen and LoveDA [58] as our experimental datasets.

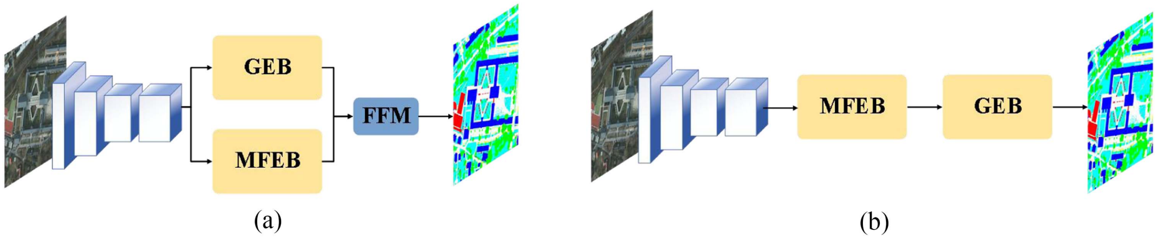

Fig. 11. Diagram of sequential structure and parallel structure. (a) Parallel structure and (b) sequential structure.

TABLE I
ABLATION EXPERIMENTS ON OUR PROPOSED MODULES AND ARCHITECTURES ON THE POTSDAM DATASET

Method	MultiScale	Global Context	Sequential	Parallel	OA	mF1	mIoU	Params
DHRNet					90.5	92.0	85.4	**4.2M**
	√				91.14	92.51	86.26	6.8M
		√			91.12	92.50	86.31	5.4M
	√	√	√		91.20	92.55	86.47	8.2M
	√	√		√	**91.51**	**92.94**	**86.97**	6.6M

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

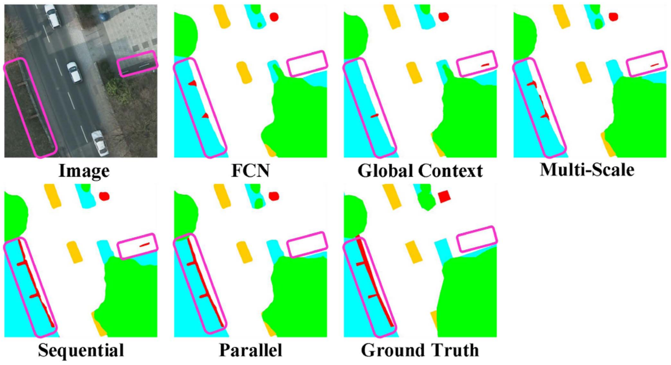

Fig. 12. Results of the ablation experiments for several of our proposed modules using the Potsdam dataset. Where, without using any of our proposed modules, the architecture is FCN with a van-tiny backbone.

TABLE II
ABLATION EXPERIMENTS FOR CLASS RATIO EXTRACTION MODULES

Method	Class Ratio Extraction	OA	mF1	mIoU	Params
DHRNet		91.51	92.94	**86.97**	**6.6 M**
	√	**91.62**	**92.96**	86.97	6.8 M

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

ISPRS Potsdam dataset provides 38 high-resolution aerial images with a resolution of 6000 × 6000. The data covers an area of 3.42 km^2, and each image contains four channels, near-infrared, red, green, and blue. In addition, each image has data from DSM and six labeling categories including impervious surfaces, buildings, low vegetation, trees, cars, and background/clutter. All the categories are manually labeled at the pixel level. To facilitate comparison with other works, we follow the same dataset split method as many previous works, using IDs: 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 for testing, using ID: 2_10 for validation, and the remaining 22 images, except for the one with erroneous annotation named 7_10, for training. In our experiments, only the red, green, and blue channels are used. Sample images and labels are shown in Fig. 10.

ISPRS Vaihingen dataset provides 33 high-resolution aerial images, each with different specific resolutions and an average size of 2494 × 2064. Similar to the Potsdam dataset, the Vaihingen images include near-infrared, red, green, and blue channels, as well as corresponding DSM data, with the same six labeling categories including impervious surfaces, buildings, low vegetation, trees, cars, and background/clutter. We used area IDs 2, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 27, 29,31, 33, 35, and 38 for testing, area ID 30 for validation, and the remaining 15 images for training. We only used the red, green, and blue channels in our experiments. Sample images and labels are shown in Fig. 10.

LoveDA dataset is a remote sensing image semantic segmentation dataset provided by the RSIDEA team at Wuhan University. The dataset contains 5987 images with a spatial resolution of 0.3 m and 166 768 annotated semantic objects. LoveDA dataset covers urban and rural areas with significant differences in scene characteristics, making it a challenging dataset. Each image has seven labeling categories including building, road, water, barren, forest, agriculture, and background. The dataset has been divided into training, validation, and test sets by the authors. It is worth noting that the test set of the dataset uses online validation to ensure fairness and authenticity. Sample images and labels are shown in Fig. 10.

B. Training Setting

We trained our model using the PyTorch framework, and all experiments were conducted on an NVIDIA RTX A5000 GPU with 24 GB of memory. We used the AdamW optimizer for network optimization, with an initial learning rate of 0.0003, and

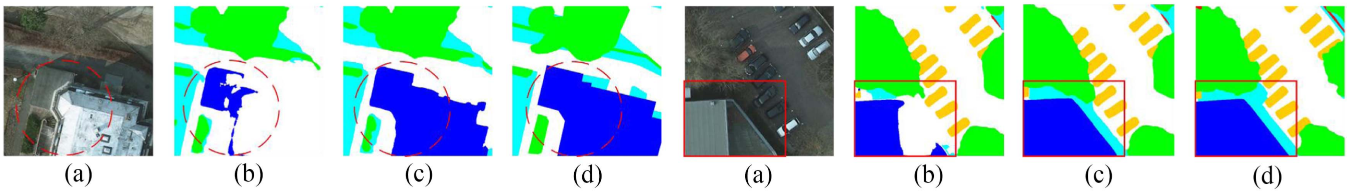

Fig. 13. Comparison of prediction results at an early stage of training (fifth epoch) for a model constructed solely using GEB. (a) Origin image. (b) Original prediction results. (c) Prediction results after using CREM. (d) Ground truth.

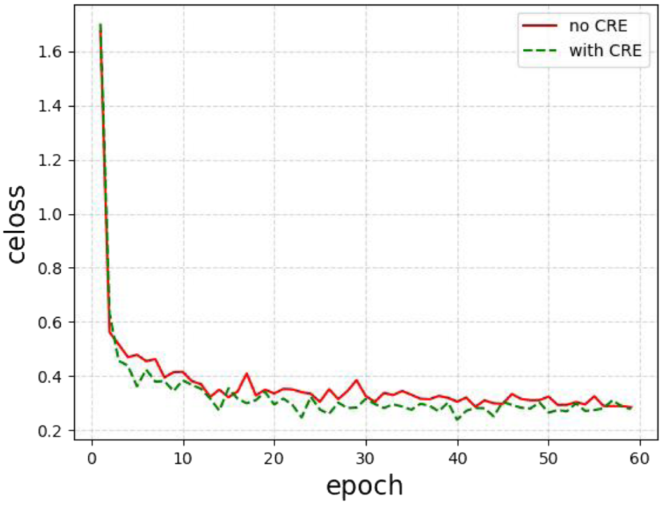

Fig. 14. Visualization of the main training loss (cross-entropy loss) reduction speed before and after adding the class ratio extraction module.

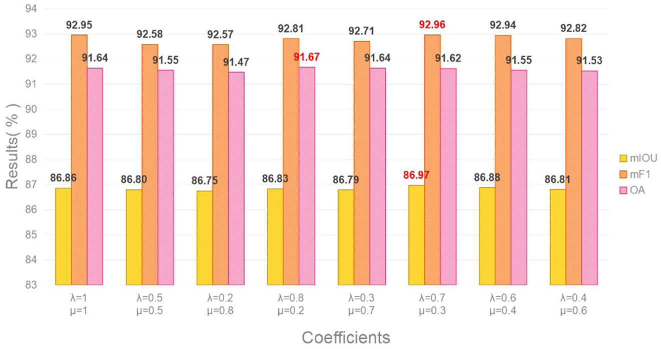

Fig. 15. Visualization of the accuracy of various metrics for eight different loss coefficient sets.

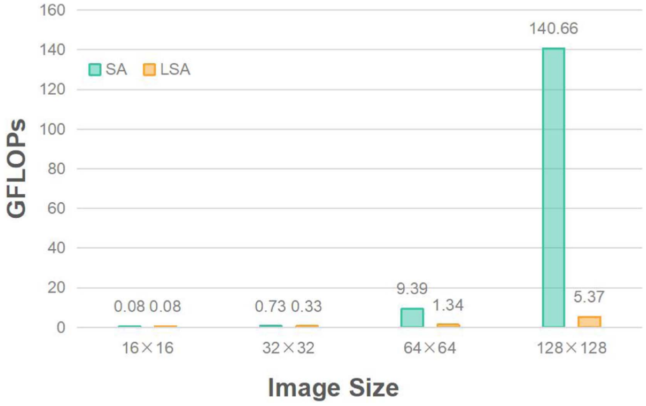

Fig. 16. Comparison of computational complexity for two attention modules at different feature map sizes (with a constant channel count of 256).

the yolox_warm_cos_lr formula [59] for learning rate updates. We set the batch size to 16 for the Potsdam and LoveDA dataset, and 8 for the Vaihingen dataset. In order to speed up model training, like other works, we cropped the datasets and their corresponding labels into patch size of 512 × 512. During the training process, we employed random horizontal and vertical flipping, random scaling, and random cropping for data augmentation. Since the introduction of CREM aims to address the issues associated with self-attention mechanisms, it may not be suitable for all models. Therefore, to ensure a fair comparison, we only utilized cross-entropy loss and dice loss when comparing with other networks. Instead, we demonstrated the effectiveness of this module and loss through ablation experiments. The maximum number of epochs for Potsdam, LoveDA and Vaihingen were 60, 80, and 100, respectively.

C. Evaluation Metrics

We evaluate the performance of these models using overall accuracy (OA), mean intersection over union (mIoU, usually the mean of IoU of all categories), and mean F1 score (usually the mean of F1 score of all categories). OA is used to measure the proportion of correctly classified results in the total samples, with a maximum value of one and a minimum value of zero, defined by the following equation:

$$\mathrm{OA} = \frac{\sum_{i=1}^{K} \mathrm{TP}_i}{\sum_{i=1}^{K} \mathrm{TP}_i + \mathrm{FP}_i + \mathrm{TN}_i + \mathrm{FN}_i}. \tag{16}$$

Intersection over Union (IoU) is a standard performance metric for object category segmentation problems. Given a set of images, IoU measures the similarity between predicted regions of the objects and the ground truth regions in the image set. It is defined by the following equation:

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{17}$$

$$\mathrm{mIoU} = \frac{1}{K}\sum_{i=1}^{K} \mathrm{IoU} = \frac{1}{K}\sum_{i=1}^{K} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i + \mathrm{FP}_i}. \tag{18}$$

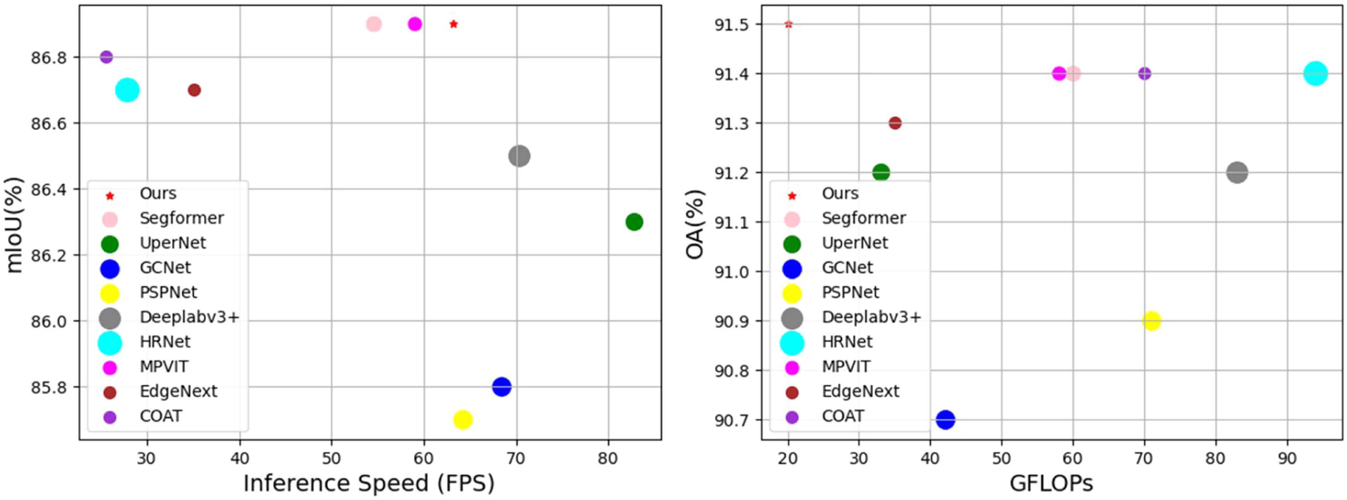

Fig. 17. Depiction of the inference speed-mIoU tradeoff and the FLOPs-OA tradeoff of different semantic segmentation methods generated on the 512 × 512 pixel-sized ISPRS Potsdam dataset using the RTX A5000 GPU. The red star-shaped dot represents our proposed method, while other colored dots correspond to different networks. A higher FPS (frames per second) indicates faster prediction speed of the model. FLOPs (floating point operations) here are different from FLOPS (floating point operations per second). It is worth noting that the size of the corresponding dots is positively correlated with their parameters, where larger dots indicate a larger number of parameters for the network.

TABLE III
Comparison of Two Self-Attention Modules

Method	**SAM**	**LSM**	**mIoU**	**OA**	**GFLOPs**	**FPS**
GEB	√		86.39	91.17	14.06	67
		√	86.31	91.12	11.55	74

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

The F1 score is a metric in statistics used to measure the accuracy of a model. It takes into account both the precision and recall of a classification model. It can be seen as a harmonic mean of a model's precision and recall, with a maximum value of one and a minimum value of zero, and is defined by the following equation:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20}$$

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{21}$$

$$mF1 = \frac{1}{K}\sum_{i=1}^{K} F1 = \frac{1}{K}\sum_{i=1}^{K} 2 \times \frac{\text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i}. \tag{22}$$

It is worth noting that in the above equations, TP, TN, FP, and FN represent the counts of true positive, true negative, false positive, and false negative, respectively, and K represents the number of classification categories. In our experiments, the calculation scope of OA includes all categories, including the background/clutter. We follow the same test method as many previous works, for the Potsdam and Vaihingen datasets, we conducted all experiments using eroded labels during testing and employed multiscale augmentation.

D. *Experimental Results*

1) Ablation Experiments: In order to verify the effectiveness of the individual modules of our proposed model and the rationality of the parallel architecture we ultimately chose, we conducted ablation experiments. In Fig. 11, a brief overview of the connection process between sequential and parallel structures is presented. As shown in Table I, we selected the FCN architecture with Van-tiny [68] backbone as the baseline for comparison. We tested the effectiveness of the proposed MFEB, GEB and compared the evaluation metrics of sequential and parallel architectures. It is evident that our proposed MFEB and GEB can effectively enhance the accuracy of segmentation results even when used individually. But, when the two branches are combined in a sequential way, in the process of layer-by-layer upsampling, concatenation, and fusion, this sequential structure introduces a large number of low-level features, resulting in a reduction in the contribution of deep-level features to the final prediction, which reduces the effectiveness of MFEB. Therefore, the improvement of the sequential structure is limited. In contrast, using the parallel structure, compared with the sequential structure, the OA was increased by 0.31%, the mF1 score was increased by 0.39%, and the mIoU was increased by 0.5%. It is worth noting that the parallel architecture separates the semantic segmentation tasks that each branch needs to process, allowing us to reduce the number of feature map channels while maintaining high accuracy. Therefore, compared with the sequential architecture, the parallel architecture also reduces the parameter

TABLE IV
COMPARISON WITH THE STATE-OF-THE-ART NETWORKS ON THE POTSDAM DATASET

Method	**Backbone**	**Imp. Surf.**	**Building**	**Low Veg.**	**Tree**	**Car**	**mIoU**	**mF1**	**OA**	**Params**
BiSeNetV2[60]	/	91.30	94.77	87.21	88.50	95.51	84.44	91.46	89.75	**5.2 M**
FCN-8s [30]	ResNet101[61]	91.81	95.66	87.23	88.51	95.96	84.94	91.86	90.02	49.83 M
GCNet [62]	ResNet101 [61]	92.79	96.10	87.37	88.82	96.06	85.79	92.23	90.74	43.0 M
DANet [35]	ResNet101	92.55	96.43	87.88	89.26	94.97	85.73	92.22	90.83	68.5 M
UperNet [63]	ResNet101	93.38	96.66	87.89	89.02	96.01	86.41	92.59	91.21	48.4 M
DeeplabV3+ [34]	Xception [64]	92.98	96.65	88.17	89.73	95.91	86.56	92.69	91.22	54.7 M
HRNetV2 [65]	HRNetV2-W48	93.33	96.89	88.03	89.50	96.09	86.71	92.76	91.39	65.9 M
EdgeNext[66]	EdgeNext-Base	93.27	96.75	88.28	89.52	96.08	86.73	92.78	91.33	18.51 M
Segformer [67]	MiT-B2	**93.45**	96.83	88.48	**89.98**	95.75	86.91	92.90	91.48	27.4 M
COAT [37]	COAT-Small	**93.45**	97.07	88.01	89.91	96.33	86.85	92.91	91.40	22.02 M
MPVIT [50]	MPVIT-Small	93.37	**97.09**	87.98	89.33	**96.49**	86.88	92.86	91.38	23.07 M
Ours	Van-tiny	93.36	97.01	**88.53**	89.48	96.38	**86.97**	**92.94**	**91.51**	6.6 M

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

TABLE V
COMPARISON WITH THE STATE-OF-THE-ART NETWORKS ON THE VAIHINGEN DATASET

Method	**Backbone**	**Imp. Surf.**	**Building**	**Low Veg.**	**Tree**	**Car**	**mIoU**	**mF1**	**OA**	**Params**
BiSeNetV2	/	92.55	95.27	83.73	88.65	88.79	82.05	90.01	90.54	**5.5 M**
FCN-8s	ResNet101	92.37	95.41	83.44	88.79	89.35	82.30	89.99	90.18	49.83 M
GCNet	ResNet101	93.12	95.84	85.06	90.13	88.20	82.52	90.47	90.90	43.0 M
DANet	ResNet101	92.83	95.82	84.69	90.03	88.25	82.58	90.32	91.00	68.5 M
UperNet	ResNet101	93.30	95.74	84.67	90.01	88.96	82.93	90.54	91.15	48.4 M
DeeplabV3+	Xception	**93.41**	96.01	84.43	90.17	88.91	83.04	90.59	91.25	54.7 M
HRNetV2	HRNetV2-W48	93.01	95.92	83.82	89.92	89.69	82.85	90.47	90.94	65.9 M
EdgeNext	EdgeNext-Base	93.05	95.77	84.50	89.92	88.73	82.70	90.39	91.02	18.51 M
Segformer	MiT-B2	93.32	95.99	84.58	90.16	89.27	83.16	90.67	91.23	27.4 M
COAT	COAT-Small	93..25	96.08	84.62	90.11	**89.77**	83.32	90.77	91.25	22.02 M
MPVIT	MPVIT-Small	93.30	**96.19**	84.46	89.87	89.71	83.27	90.70	91.21	23.07 M
Ours	Van-tiny	93.34	96.17	**85.27**	**90.25**	89.20	**83.53**	**90.91**	**91.36**	6.6 M

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

count by 22%. Fig. 12 shows the visualization results of the five different network architectures listed in Table I. We can observe a significant improvement in the segmentation results of elongated objects after adding MFEB (refer to the left pink box in Fig. 12), which demonstrates that the proposed MFEB possesses strong feature capturing capabilities.

With the use of our proposed parallel architecture, the final segmentation results are significantly better than the other four architectures. According to Table II, the model's overall accuracy and mean F1 score were both improved after incorporating CREM. During the training process, as the module simultaneously serves as a form of deep supervision, it converges faster. It is worth noting that our proposed CREM is designed to provide an additional inductive bias to GEB, accelerating its convergence, and it can be used in the early stages of training. Therefore, while the accuracy improvement from CREM may not be as impressive as other modules, this module can be disabled during the inference phase without affecting the model's prediction speed and parameter count.

To further demonstrate that the CREM can accelerate the convergence speed of the model, we conducted experiments to analyze the contrast in model predictions during the early stages of training before and after adding this module, as well as to observe the changes in the main training loss (cross-entropy loss). Fig. 13 illustrates the impact of using CREM on the early-stage predictions of a model constructed solely using GEB. It is evident from the figure that after employing CREM, the model can fit more rapidly and yield a more refined segmentation result. From Fig. 14, we can observe that, even though CREM is applied only to GEB, adding this module still results in a faster decrease in the training cross-entropy loss of the entire network.

To select the appropriate loss coefficient, we conducted comparative test on eight sets of different loss coefficients, and the results are shown in Fig. 15. In the case of $\lambda = 0.8$ and $\mu = 0.2$, the model achieved the highest OA of 91.67%. In the case of $\lambda = 0.7$ and $\mu = 0.3$, the model achieved the highest mIOU of 86.97% and mF1 of 92.96% simultaneously. Therefore, we finally selected this set of coefficients as the loss coefficients.

2) Comparison Between the Improved Linear Self-Attention Module and the Original Self-Attention Module: Fig. 16 provides a computational complexity comparison between the original self-attention module (SAM) and the linear self-attention module (LSM) at different image sizes. As the image size increases, the disparity in complexity between the two modules

TABLE VI
COMPARISON WITH THE STATE-OF-THE-ART NETWORKS ON THE LOVEDA DATASET

Method	Backbone	Background	Building	Road	Water	Barren	Forest	Agriculture	mIOU	Params
FCN-8s	ResNet101	44.68	53.76	52.03	76.92	16.96	45.35	57.67	49.63	49.83 M
GCNet	ResNet101	46.35	58.58	58.94	81.5	19.15	46.69	62.70	53.42	43.0 M
DANet	ResNet101	45.75	56.96	56.91	81.11	18.60	47.88	63.19	52.91	68.5 M
UperNet	ResNet101	**48.02**	59.06	**60.34**	80.23	16.86	47.07	64.99	53.80	48.4 M
Deeplabv3+	Xception	46.54	57.45	57.31	79.87	18.26	**47.91**	64.36	53.10	68.5 M
HRNetV2	HRNetV2-W48	46.09	59.61	59.39	80.35	18.06	47.23	62.55	53.33	65.9 M
EdgeNext	EdgeNext-Base	45.71	58.07	56.45	81.00	16.17	47.26	61.62	52.43	18.51 M
Segformer	MiT-B2	47.09	58.66	57.39	81.30	18.84	45.74	65.10	53.45	27.4 M
COAT	COAT-Small	47.45	60.57	58.63	80.97	18.95	46.60	64.54	54.26	22.02 M
MPVIT	MPVIT-Small	47.93	**61.21**	59.59	81.03	18.10	46.92	**65.12**	54.40	23.07 M
Ours	Van-tiny	47.18	60.21	59.92	**81.60**	**20.38**	46.69	64.14	**54.48**	6.6 M

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

TABLE VII
COMPARISON OF OUR PROPOSED NETWORK WITH OTHER COMMONLY USED SEGMENTATION NETWORKS USING THE SAME BACKBONE ON THE POTSDAM DATASET

Method	Backbone	Imp. Surf.	Building	Low Veg.	Tree	Car	mIoU	mF1	OA	Params
FCN-8s	Van-tiny[68]	92.89	96.29	87.50	89.14	95.67	85.98	92.30	90.83	4.5 M
PSPNet [42]	Van-tiny	92.70	96.33	87.73	89.33	95.28	85.77	92.27	90.91	4.3 M
GCNet	Van-tiny	92.03	96.19	86.96	88.01	94.50	85.03	91.57	90.27	**3.9 M**
DANet	Van-tiny	92.95	96.57	87.92	89.25	95.34	86.07	92.41	91.04	4.2 M
FPNet[33]	Van-tiny	92.90	96.46	87.80	89.40	95.44	86.06	92.40	91.03	5.6 M
UperNet	Van-tiny	93.33	96.60	87.59	89.23	95.29	86.09	92.41	91.11	7.0 M
DeeplabV3+	Van-tiny	92.55	96.43	87.88	89.26	94.97	85.73	92.22	90.83	7.4 M
Ours	Van-tiny	**93.36**	**97.01**	**88.53**	**89.48**	**96.38**	**86.97**	**92.94**	**91.51**	6.6 M

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

TABLE VIII
COMPARISON OF OUR PROPOSED NETWORK WITH OTHER COMMONLY USED SEGMENTATION NETWORKS USING THE SAME BACKBONE ON THE VAIHINGEN DATASET

Method	Backbone	Imp. Surf.	Building	Low Veg.	Tree	Car	mIoU	mF1	OA	Params
FCN-8s	Van-tiny	93.00	95.46	84.23	89.85	87.51	82.02	90.16	90.79	4.5 M
PSPNet	Van-tiny	92.96	95.69	85.03	90.00	86.68	82.17	90.07	91.05	4.3 M
GCNet	Van-tiny	93.17	95.57	84.17	89.69	87.49	82.09	90.02	90.85	**3.9 M**
DANet	Van-tiny	92.99	95.68	84.44	89.89	86.73	81.98	89.94	90.94	4.2 M
FPNet	Van-tiny	93.19	95.65	84.28	89.80	88.56	82.54	90.30	90.96	5.6 M
UperNet	Van-tiny	93.33	95.64	84.50	89.90	88.03	82.52	90.28	91.06	7.0 M
DeeplabV3+	Van-tiny	93.06	95.36	83.95	89.96	88.51	82.33	90.17	90.82	7.4 M
Ours	Van-tiny	**93.34**	**96.17**	**85.27**	**90.25**	**89.20**	**83.53**	**90.91**	**91.36**	6.6 M

The bold values represent the highest accuracy or best performance under a specific evaluation metric in the current comparative experiment.

becomes more pronounced. Particularly, when the input size reaches 128 × 128, the computational complexity gap between them reaches several times, and this gap continues to widen as the output size increases. In Table III, we present a quantitative comparison of the two self-attention modules. LSM is capable of achieving higher FPS while maintaining nearly the same level of accuracy as SAM.

3) Quantitative Comparison With State-of-the-Art Methods: To further verify the effectiveness of our proposed network, we compared it with several widely used semantic segmentation networks, including Segformer, FCN-8S, HRNet, Upernet, GCNet, DANet, Deeplabv3+, BiSeNetV2, EdgeNext, COAT, and MPVIT (COAT, MPVIT, and EdgeNext utilized UperNet for decoding). It is worth noting that we compared the networks using the most suitable backbone among all the backbones used in our experiment. Tables IV–VI report the final prediction results on three datasets. The results show that the proposed DHRNet achieved the best performance in multiple metrics

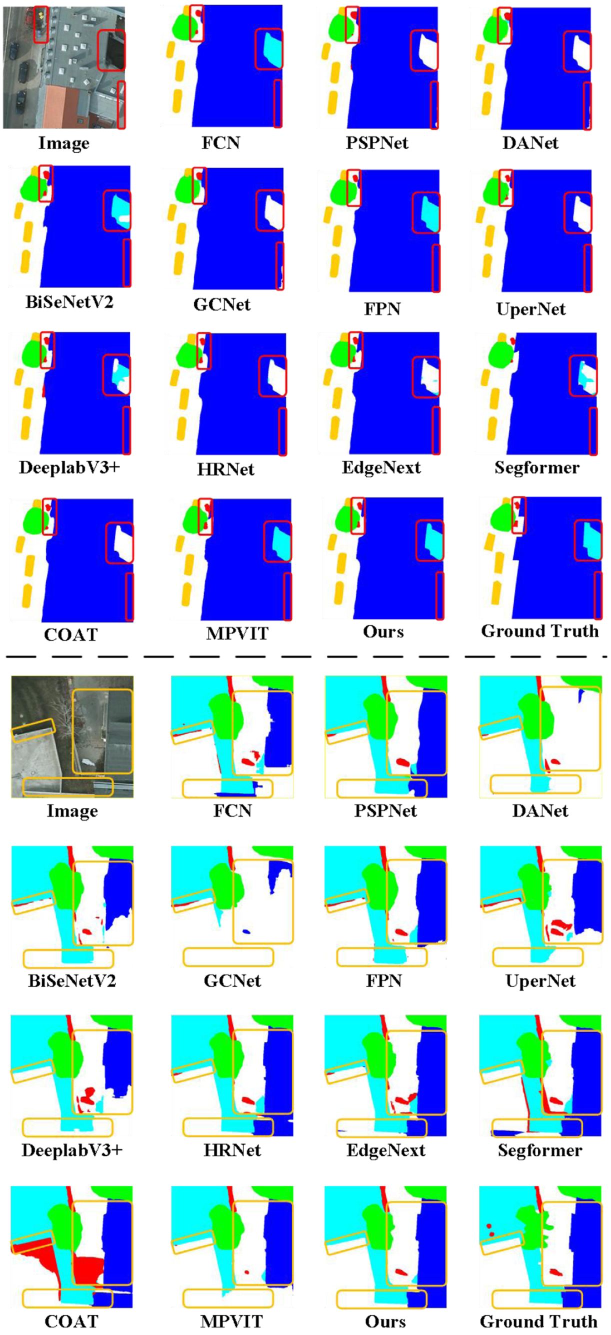

Fig. 18. Predicted segmentation maps of the most commonly used or best-performing networks at the original patch size (512 × 512) on the Potsdam dataset.

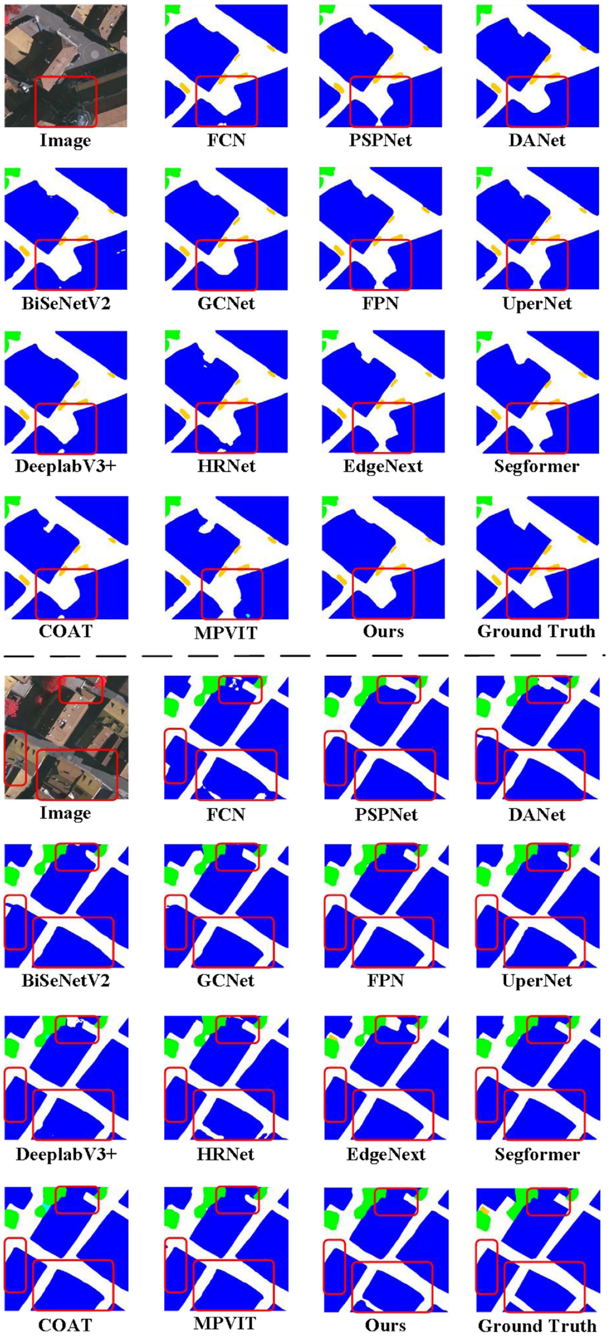

Fig. 19. Predicted segmentation maps of the most commonly used or best-performing networks at the original patch size (512 × 512) on the Vaihingen dataset.

compared with the other networks. The average metrics (mIoU, mF1, and OA) reached the highest while achieving the highest F1 score among multiple categories on both Potsdam and Vaihingen datasets. In the LoveDA dataset, the proposed DHRNet achieved the highest IOU in both water and barren, and it also had the highest mIOU. The number of parameters in our proposed network DHRNet (6.6 M) is significantly lower than other state-of-the-art networks.

4) Compared With Other Networks Using the Same Backbone: To verify the superiority of our proposed network over other networks with the same backbone, we compared DHRNet with several commonly used segmentation networks on the same backbone (Van-tiny), and our model achieved the best results. Tables VII and VIII report the results of various networks on the Potsdam and Vaihingen datasets with the same backbone (Van-tiny). It can be seen that with the same Van-tiny backbone, our proposed network outperformed any other network in all metrics on both datasets. Compared to other networks, the mean F1 score, mean Intersection over Union, overall accuracy, and the F1 score of each category have been significantly improved.

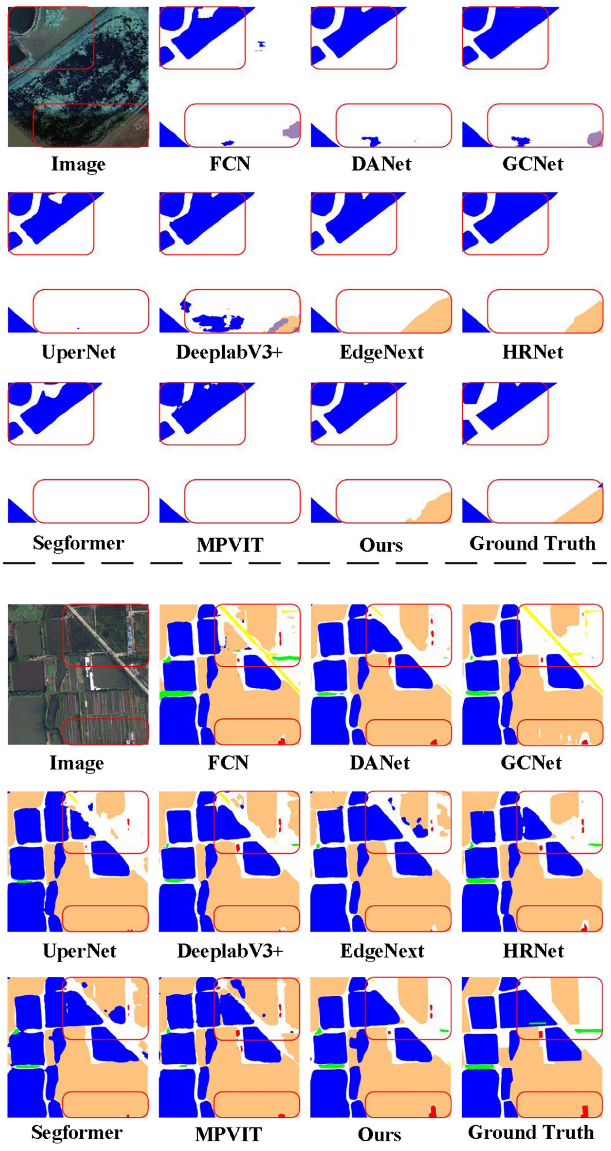

Fig. 20. Predicted segmentation maps of most commonly used or best-performing networks at the original patch size (1024 × 1024) on the LoveDA dataset.

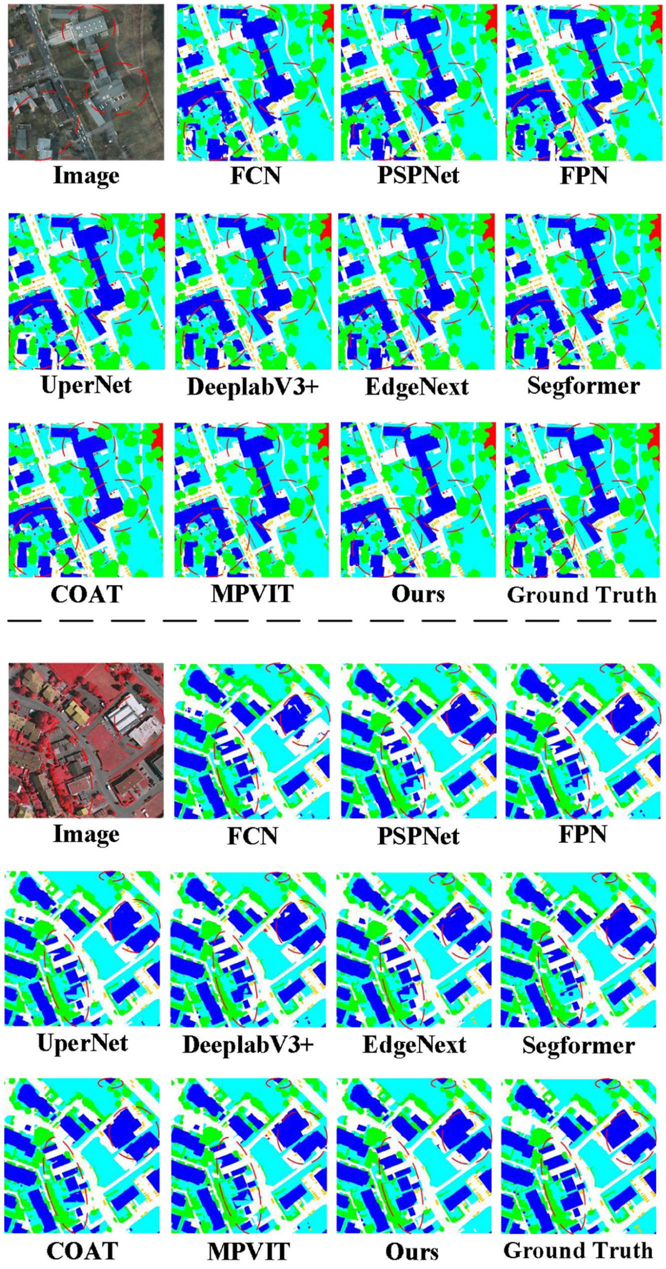

Fig. 21. Larger size prediction images of the most widely used or best-performing networks on the Potsdam and Vaihingen dataset.

These superior results suggest that the superior segmentation performance of DHRNet is not solely attributed to the choice of backbone.

5) Visualization of Experimental Results: We selected ten networks and compared them with our proposed DHRNet in terms of the speed-mIoU tradeoff and the complexity-OA tradeoff. The visualization results are shown in Fig. 17. It can be seen that our proposed DHRNet has the smallest number of parameters (6.6 M) and the lowest computational complexity (20 GFLOPs), while achieving the highest OA of 91.51% and mIoU of 86.97%.

To better illustrate the differences between our proposed network and other networks, we also visualized the final experimental results. We selected widely used networks (FCN-8s, PSPNet, BiseNetV2, and FPN) as well as networks that performed well in the experiments (DANet, GCN, UperNet, DeeplabV3+, HRNet, Segformer, EdgeNext, COAT, and MPVIT) and our proposed network DHRNet for visualizing partial test results on the Potsdam, Vaihingen and LoveDA datasets, as shown in Figs. 18–20. The displayed image size for Potsdam and Vaihingen datasets is 512 × 512, which is the size we used when splitting the dataset, while the displayed image size for LoveDA dataset is 1024 × 1024 which corresponds to the size in the provided datasets. To better compare the segmentation results of these models, we outlined some key regions. It is worth noting that all visualized results of the compared networks are predicted using the best-performing backbone in our experiments. LoveDA employs online verification to ensure the authenticity and fairness of the results. As a result, the test set does not provide

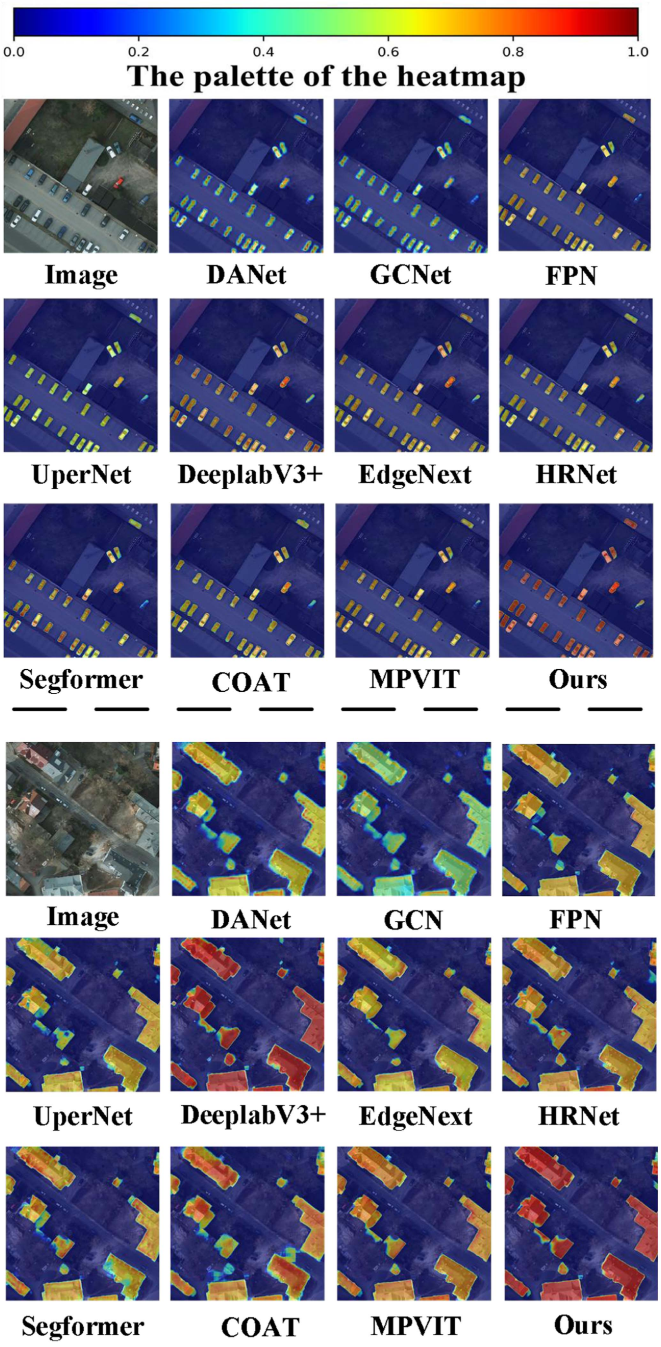

Fig. 22. Visualization result of the grad-CAM by using the final classification layer of some networks. We present the visual attention results for the categories of building (bottom) and car (top) on the Potsdam dataset. Areas with higher attention values, indicated by warmer colors, represent greater confidence in network classification.

ground truth labels. So, the visualizations we present are based on the validation set. From the visualization results, we can see that our proposed network exhibits stronger edge and small object segmentation capabilities. The misclassification rate is significantly reduced compared with other networks and the overall predicted image is smoother with fewer noise points, which is closer to the ground truth label.

RSI typically have high pixel resolutions, which often contain a diverse range of complex scenes with numerous categories. Directly performing pixel-level classification on such high-resolution RSI is considered a challenging task, which places higher demands on the segmentation capabilities of the network. Therefore, we visualized the results of some networks at larger sizes for comparison. Fig. 21 shows the higher resolution visual results of some networks on the Potsdam and Vaihingen datasets. It can be seen that our proposed network has higher classification accuracy and edge segmentation capability compared with other networks at the macro level, and the overall details are closer to the ground truth label which demonstrates that our proposed network exhibits strong classification ability even in complex scenes.

To demonstrate the superior classification and edge extraction capabilities of our proposed network for various objects, we employed Grad-CAM [69] visualization to examine the attention maps of the final classification layer of some networks on the Potsdam dataset. The visual results of Grad-CAM in Fig. 22 demonstrate our network's high confidence in target classification and strong edge segmentation capabilities.

V. Conclusion

In this article, we propose a DHRNet for semantic segmentation of large-scale RSI. We strike a balance between inference speed and final accuracy. The proposed model can perform predictions at a faster speed under the premise of high-precision results. In this proposed network, the multiscale feature extraction branch can extract more comprehensive feature information from different scales of features in RSI, while the global context detail enhancement branch can overcome the limitation of receptive fields, perform more comprehensive feature extraction, and reduce the impact of low-level features while obtaining richer spatial information. To reduce the misclassification caused by attention mechanisms, a class ratio extraction module is designed to supervise the network, which reduces misclassification errors and speeds up the convergence speed of network training. The results of a series of experiments conducted on the Potsdam, Vaihingen, and LoveDA datasets demonstrate that DHRNet is capable of achieving high-precision segmentation tasks while maintaining a small number of parameters (6.6 M) and computational complexity (20 GFLOPs). In the field of RSI semantic segmentation, DHRNet can effectively handle complex scenes, as well as small and elongated objects, and edge information. These findings highlight the robustness and generalizability of our approach, underscoring its potential as a valuable tool in the field of remote sensing. The series of remarkable experimental results demonstrate the effectiveness of our employed dual-branch parallel network architecture in handling RSI semantic segmentation tasks.

References

[1] C. Toth and G. Jozkow, "Remote sensing platforms and sensors: A survey," *ISPRS J. Photogrammetry Remote Sens.*, vol. 115, pp. 22–36, May 2016.

[2] T. Adao et al., "Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry," *Remote Sens.*, vol. 9, no. 11, Nov. 2017, Art. no. 1110.

[3] J. A. Benediktsson, J. Chanussot, and W. Moon, "Very high-resolution remote sensing: Challenges and opportunities," *Proc. IEEE*, vol. 100, no. 6, pp. 1907–1910, Jun. 2012.

[4] V. Dey, Y. Zhang, and M. Zhong, "A review on image segmentation techniques with remote sensing perspective," in *Proc. ISPRS Tech. Commission VII Symp. – 100 Years ISPRS*, vol. 38, Jan. 2010.

[5] D. S. Lu and Q. H. Weng, "Use of impervious surface in urban land-use classification," *Remote Sens. Environ.*, vol. 102, no. 1/2, pp. 146–160, May 2006.

[6] V. Maus, G. Câmara, R. Cartaxo, A. Sanchez, F. M. Ramos, and G. R. de Queiroz, "A time-weighted dynamic time warping method for land-use and land-cover mapping," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 8, pp. 3729–3739, Aug. 2016.

[7] R. Liu, D. Jiang, L. Zhang, and Z. Zhang, "Deep depthwise separable convolutional network for change detection in optical aerial images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1109–1118, 2020.

[8] P. Du, S. Liu, P. Gamba, K. Tan, and J. Xia, "Fusion of difference images for change detection over urban areas," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 4, pp. 1076–1086, Aug. 2012.

[9] Q. Bi, K. Qin, H. Zhang, Y. Zhang, Z. L. Li, and K. Xu, "A multi-scale filtering building index for building extraction in very high-resolution satellite imagery," *Remote Sens.*, vol. 11, no. 5, p. 482, Mar. 2019.

[10] Z. Chen, C. Wang, J. Li, N. Xie, Y. Han, and J. Du, "Reconstruction bias U-net for road extraction from optical remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2284–2294, 2021.

[11] Y. Xu, L. Wu, Z. Xie, and Z. Chen, "Building extraction in very high resolution remote sensing imagery using deep learning and guided filters," *Remote Sens.*, vol. 10, no. 1, p. 144, 2018.

[12] L. Zeng, B. D. Wardlow, D. Xiang, S. Hu, and D. Li, "A review of vegetation phenological metrics extraction using time-series, multispectral satellite data," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111511.

[13] S. Toure, O. Diop, K. Kpalma, and A. S. Maiga, "Shoreline detection using optical remote sensing: A review," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 2, Feb. 2019, Art. no. 75.

[14] Y. Zhou, J. Luo, Z. Shen, X. Hu, and H. Yang, "Multiscale water body extraction in urban environments from satellite images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 10, pp. 4301–4312, Oct. 2014.

[15] K. Parvati, P. Rao, and M. M. Das, "Image segmentation using gray-scale morphology and marker-controlled watershed transformation," *Discrete Dyn. Nature Soc.*, vol. 2008, 2008.

[16] T. Horiuchi, "Grayscale image segmentation using color space," *Inst. Elect., Inf. Commun. Engineers Trans. Inf. Syst.*, vol. 89, no. 3, pp. 1231–1237, 2006.

[17] A. Lucieer, A. Stein, and P. J. Fisher, "Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty," *Int. J. Remote Sens.*, vol. 26, no. 14, pp. 2917–2936, 2005.

[18] R. Trias-Sanz, G. Stamon, and J. Louchet, "Using colour, texture, and hierarchial segmentation for high-resolution remote sensing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 63, no. 2, pp. 156–168, 2008.

[19] J. Yuan, D. Wang, and R. Li, "Remote sensing image segmentation by combining spectral and texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 16–24, Jan. 2014.

[20] A. Shamir, "A survey on mesh segmentation techniques," in *Computer Graphics Forum*, vol. 27. Hoboken, NJ, USA: Wiley, 2008, pp. 1539–1556.

[21] K. H. Memon, S. Memon, M. A. Qureshi, M. B. Alvi, D. Kumar, and R. A. Shah, "Kernel possibilistic fuzzy c-means clustering with local information for image segmentation," *Int. J. Fuzzy Syst.*, vol. 21, no. 1, pp. 321–332, Feb. 2019.

[22] S. Haag, D. Schwartz, B. Shakibajahromi, M. Campagna, and A. Shokoufandeh, "A fast algorithm to delineate watershed boundaries for simple geometries," *Environ. Model. Softw.*, vol. 134, Dec. 2020, Art. no. 104842.

[23] H. Shi, Y. Yu, and Y. Wang, "Early warning method for sea typhoons using remote-sensing imagery based on improved support vector machines (SVMs)," *J. Coastal Res.*, vol. 82, pp. 180–185, 2018.

[24] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 809–823, Mar. 2012.

[25] J. Peng, L. Li, and Y. Y. Tang, "Maximum likelihood estimation-based joint sparse representation for the classification of hyperspectral remote sensing images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1790–1802, Jun. 2019.

[26] G. Cao, X. Li, and L. Zhou, "Unsupervised change detection in high spatial resolution remote sensing images based on a conditional random field model," *Eur. J. Remote Sens.*, vol. 49, pp. 225–237, 2016.

[27] L. Dong et al., "Very high resolution remote sensing imagery classification using a fusion of random forest and deep learning technique-subtropical area for example," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 113–128, 2020.

[28] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, 2017.

[29] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Appl. Sci.*, vol. 9, no. 19, 2019, Art. no. 4050.

[30] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[31] A. Coates and A. Ng, "Selecting receptive fields in deep networks," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 2528–2536.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv.: 18th Int. Conf.*, 2015, pp. 234–241.

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis Pattern Recognit.*, 2017, pp. 2117–2125.

[34] 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.*, 2018, pp. 801–818.

[35] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.

[36] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[37] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9981–9990.

[38] R. Li et al., "Multiattention network for semantic segmentation of fine-resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5607713.

[39] J. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.*, vol. 12, pp. 149–198, 2000.

[40] J. C. Ye and W. K. Sung, "Understanding geometry of encoder-decoder CNNs," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7064–7073.

[41] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[42] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[43] P. Wang et al., "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1451–1460.

[44] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[45] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[46] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representation*, 2021, pp. 1–11.

[47] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[48] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 568–578.

[49] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[50] Y. Lee, J. Kim, J. Willette, and S. J. Hwang, "Mpvit: Multi-path vision transformer for dense prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2022, pp. 7287–7296.

[51] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[52] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2d human pose estimation: A survey," *Tsinghua Sci. Technol.*, vol. 24, no. 6, pp. 663–676, 2019.

[53] L. Liu et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, pp. 261–318, 2020.

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

[55] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[56] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31x31: Revisiting large kernel design in cnns," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11963–11975.

[57] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4905–4913.

[58] J. Wang, Z. Zheng, A. Ma, X. Lu, and Y. J. Zhong, "LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation," in *Proc. Neural Inf. Process. Syst.*, vol. 1, 2021, pp. 1–16.

[59] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," 2021, *arXiv:2107.08430*.

[60] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, pp. 3051–3068, 2021.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[62] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4353–4361.

[63] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 418–434.

[64] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[65] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[66] M. Maaz et al., "Edgenext: Efficiently amalgamated cnn-transformer architecture for mobile vision applications," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 3–20.

[67] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 12077–12090, 2021.

[68] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *Comput. Vis. Media*, vol. 9, no. 4, pp. 733–752, 2023.

[69] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.

Qinyan Bai received the B.E. degree in computer science and technology from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2022. He is currently working toward the master's degree in computer science and technology with the Chongqing Engineering Research Center of Spatial Big Data Intelligence Technology, Chongqing, China.

His research interests include the processing of remote sensing images and deep learning.

Xiaobo Luo received the B.S. degree in GIS from the Institute of Geophysics and Geomatics, China University of Geosciences, Wuhan, China, in 1999, the M.S. degree in GIS from the School of Information Engineering, Chinese Academy of Sciences, Wuhan, in 2004, and the Ph.D. degree in cartography and GIS from the Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2010.

He is a Professor with the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include urban thermal infrared remote sensing, remote sensing image processing, and ecological environment monitoring and evaluating

Yaxu Wang received the B.E. degree in computer science from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2018, and the M.S. degree in GIS from the University of Queensland, Brisbane, QLD, Australia, in 2020. She is currently working toward the doctoral degree in computer science and technology with the Chongqing University of Posts and Telecommunications.

Tengfei Wei received the B.E. degree in software engineering from the Guilin University of Electronic Technology, Guilin, China, in 2021. He is currently working toward the master's degree in computer science and technology with the Chongqing Engineering Research Center of Spatial Big Data Intelligence Technology, Chongqing, China.

His research interests include remote sensing image field extraction and instance segmentation.