IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 30, 2020, accepted January 25, 2021, date of publication February 10, 2021, date of current version February 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058571

Real-Time Semantic Segmentation of Remote Sensing Images Based on Bilateral Attention Refined Network

JIALI CAI[1], CHUNJUAN LIU[1], HAOWEN YAN[1], XIAOSUO WU[1,2,3], WANZHEN LU[1], XIAOYU WANG[1], AND CHANGLIN SANG[1]

[1]School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China
[2]Institute of Sensor Technology, Gansu Academy of Science, Lanzhou 730070, China
[3]Key Laboratory of Opt-Technology and Intelligent Control, Ministry of Education, Lanzhou Jiaotong University, Lanzhou 730070, China

Corresponding authors: Xiaosuo Wu (wuxs_laser@lzjtu.cn) and Chunjuan Liu (liuchj@mail.lzjtu.cn)

This work was supported in part by the National Key Research and Development Program under Contract 2017YFB0504203, Planned project of Gansu science and Technology Department under Contract 18JR3RA123, Department of education of Gansu Province under Contract 2018B-029, Department of housing and urban rural development of Gansu Province under Contract JK2017-24, Youth Science Fund Project of Lanzhou Jiaotong University under Contract 2016003, Science and Technology Bureau of Chengguan District, Lanzhou City 2018-4-5, and Innovation fund of Gansu Academy of science and technology under Contract 2018QN-05.

ABSTRACT The trade-off between feature representation capability and spatial positioning accuracy is crucial to dense classification or semantic segmentation of remote sensing images. In order to better balance the low-level spatial details in the shallow network and the high-level abstract semantics in the deep network, the bilateral attention refinement lightweight network BARNet is introduced. In this way, we can use the fine-grained features in the shallow layer to further supplement and capture the deeper information of the high-level semantic features. The network employs an asymmetric encoder decoder architecture for the task of real-time semantic segmentation. Encoder part proposes a lightweight network residual unit with the split, concatenate and split bottleneck structure to achieve more light weighted, effificient and powerful feature extraction. In the decoding section, we propose an adaptive method to enhance feature representation in local attention enhancement module. In addition, the global context embedding module is introduced to divide the high-level features into two branches. One branch gets the weight vector to guide the low-level learning, and the other branch will get a semantic vector, which is used to calculate the multi-label category loss and further introduce into the overall loss function to regulate the training process better. The effectiveness and efficiency of the network are verified on ISPRS Potsdam data set and CCF data set, respectively. The results show that the models using these strategies outperform the baseline network on MIoU, PA and F1, which increase by 18.86%, 16.21% and 15.64% on the Potsdam dataset; 10.51%, 6.53% and 8.19% on the CCF dataset.

INDEX TERMS Remote sensing image, real-time semantic segmentation, local attention enhancement module, global context embedding module, multi-label category loss.

I. INTRODUCTION

The dense classification or semantic segmentation of remote sensing images (RSIs) is a critical step in the automatic analysis of remote sensing data, which is widely used in railway track risk assessment, land planning, environmental monitoring, and urban planning, etc. In recent years, with the development of convolutional neural networks, the accuracy of semantic segmentation on RSIs has been dramatically improved. In 2014, Fully Convolutional Network (FCN) [1] is proposed. FCN replaced the fully connected layer of the network with convolution. But the pooling operation in the network would reduce the resolution and thus weaken the location information. In order to solve this problem, the encoder-decoder framework is proposed [2]. In the encoder part, pooling gradually reduces the spatial dimension, while the decoder part gradually recovers the spatial dimension and detailed information. In order to effectively

The associate editor coordinating the review of this manuscript and approving it for publication was Weipeng Jing.

restore spatial information, U-net [3] adds a cross-layer connection. Later, DeeplabsV3 [4] and PSPNet [5] respectively extend the global average pool to the Atrous Spatial Pyramid Pooling and Spatial Pyramid Pooling.

However, the establishment of deeper and larger convolutional neural networks has also increased the running time of the network. Some existing semantic segmentation models have improved inference speed to a certain extent, at the expense of low-level details and channel capacity, resulting in a sharp decline in accuracy. Therefore, designing a network with high efficiency, small parameter capacity and high accuracy has become a challenging problem at present.

In order to overcome the problem that the platform's energy overhead and memory capacity limit the execution efficiency of semantic segmentation tasks, a variety of lightweight networks are built. For example, to solve a large number of floating-point operations in the network, a new and effective deep lightweight neural network ENet [6] is proposed. This network reduces the parameters by compressing the channel, but the spatial information will be destroyed, and the accuracy can't be improved. ICNet [7] introduces the cascaded feature fusion module based on PSPNet to realize a fast and high-quality segmentation model. ICNet accelerates the capture of semantics through low resolution, acquires details through high resolution, and merges features through cascaded networks. To accelerate the speed, the existing methods often adopt the method of losing the spatial resolution, which results in a severe decrease in precision. To maintain the accuracy, a new bidirectional segmentation network BiSeNet [8] appeared, which designs spatial paths and semantic paths to obtain spatial location information and semantic information. Finally, a new feature fusion module is introduced to combine the two feature maps to achieve a balance of speed and precision.

We believe that all levels of features contribute to semantic segmentation. High-level semantic features can better identify region categories, while low-level features can capture clearer and more detailed boundary textures. Therefore, we propose a bilateral attention refinement network BARNet, which extracts spatial details and classification semantics separately to enhance the receiving domain and capture rich contextual information.

In a nutshell, the main contributions of this article are as follows:

(1) A local attention enhancement module (LAEM) is proposed, which extracts scene information from local patches. This module can capture long-range dependencies and further reconstruct the feature maps for better representation.
(2) A global context embedding module (GCEM) is proposed to enhance the semantic representation of low-level features by introducing the attention of high-level features, and merge refined low-level features with high-level features to improve the rough representation of high-level features. Moreover, the introduction of semantic multi-label category loss can better standardize the training process.
(3) On the basis of the above two modules, a novel bilateral attention refinement network (BARNet) is proposed, which can better balance feature representation ability and spatial positioning accuracy from the perspective of space and channel.

The remainder of this article is organized as follows. Section 2 introduces the related works on semantic segmentation tasks. The proposed method is discussed in Section 3. Experimental data sets and evaluations are described in Section 4. Experimental results and discussions are presented in Section 5. As a final point, our work and future work are discussed in Section 6.

II. RELATED WORK

A. REAL-TIME SEMANTIC SEGMENTATION

Real-time semantic segmentation algorithms generate high-quality predictions with limited computations, which are usually executed under resource constraints or mobile applications [9]. Current real-time semantic segmentation models can be generally divided into two types [10]–[16]. The first one uses the existing lightweight backbone to extract features efficiently. For example, based on Xception [17] and MobileNet [18], [19] backbone networks, some effective feature aggregation or multi-branch modules are used to merge the low-level feature with the high-level feature. ICNet takes multi-scale images as input and introduces a cascade network. DFANet aggregates discriminative functions through sub-network and sub-stage cascade, respectively. BiSeNet designs two branches to deal with spatial details and categorical semantics separately. These models can achieve high accuracy.

The second is designed with valid modules that use methods such as convolutional solution and dilated convolution to reduce the computation and expand the acceptance field, and these modules are reused throughout the network to extract features. ENet [6] is the first real-time lightweight network proposed, which reduces the amount of calculation by decreasing the number of downsampling or the number of filters. ESPNet [20], [21] adopts an efficient spatial pyramid module to improve performance. Simultaneously, ERFNet [22] designs a non-bottleneck 1D module with residual connection and decomposition convolution to obtain excellent accuracy while maintaining high efficiency. In this article, we design a feature extraction unit using convolutional solution and depth separable convolution, by downsampling and repeatedly superimposing these units to construct a lightweight and efficient encoder.

B. ATTENTION MECHANISM

In vision tasks, attention mechanism first computes the attention weights that represent the degree of importance of features, and then the weight value is used to capture more informative features from the input feature maps [23].

In SENet [24], the squeeze-and-excitation (SE) block is put forward, which uses global-pooling to generate channel attention. In SCAttNet [25] and CBAM [26], based on an effective architecture, both spatial attention and channel attention are used, and the average pool and maximum pool functions are used in the two modules to increase the network's presentation ability. CCNet [27] harvests the contextual information of all the positions by stacking two serial crisis-cross attention modules. DANet [28] employs similar two modules to learn the information of all pixels and channel dependence. EncNet [29] introduces a context encoding module at the end of the network to encode global contextual information and re-weight the extracted features for discriminative representations. ACFNet [30] proposes a module based on attention category features to improve classification efficiency. HMANet [31] proposes a category-enhanced attention module, which can better distinguish the corresponding category through the category correlation between pixels. The category channel attention module is also embedded to re-weight category level and channel correlation.

In this article, the attention mechanism will be applied to the spatial dimension without increasing the amount of calculation. Inspired by the global attention upsampling module [32], we introduce a global context embedding module (GCEM) to embed the semantic information in high-level features into low-level features.

C. DEPTHWISE SEPARABLE CONVOLUTION

Depthwise separable convolution operation can reduce the computational cost and the number of parameters while maintaining similar (or slightly better) performance. MobileNet [19] convolves each input channel with the corresponding kernel channel. Then, a 1×1 kernel is used to perform point-wise convolution to project the output of the deep convolution into a new channel space. Inspired by it, this paper proposes a Local Attention Enhancement Module (LAEM) that reduces parameters through deep separable convolution, which can greatly reduce network parameters and contribute to the scalability of the network.

D. AUXILIARY LOSS

In recent years, some related studies have tried to use the auxiliary loss to optimize the segmentation model. PSPNet [5] proposes an additional loss to generate the results of each module. In the main network ResNet101 [33] model, in addition to training the final classifier on the main branch by using softmax loss, another loss is applied after the fourth stage to optimize the training process of the network. So this article also introduces auxiliary loss to standardize the network training process.

III. PROPOSED METHODS

A. OVERVIEW OF THE PROPOSED BARNet

As shown in Figure 1, for the input remote sensing image, firstly, the backbone network is used for feature extraction. Then the feature maps extracted from the shallow layer and deep layer are fed into the attention enhancement module to refine the features in channels, and the refined channel feature maps are fed into the global embedded module. Secondly, the extended loss function with additional consideration of global significance can better normalize the network to perceive context information and further maintain the accuracy of the model. The final results are produced by fusing the features from both branches.

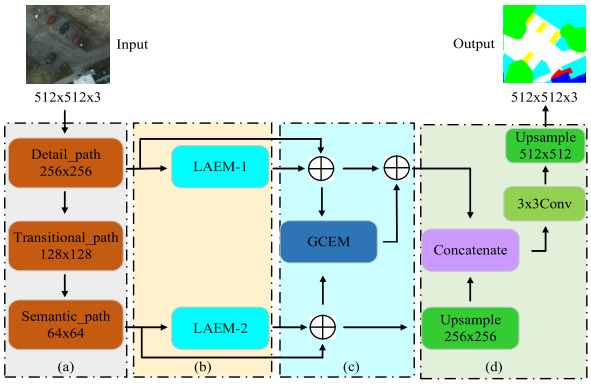

FIGURE 1. The architecture of the bilateral attention refinement network BARNet: (a) Lightweight encoder, which is constructed by superposition of scs-bt modules. (B) Local Attention Enhancement Module (LAEM), Which uses channel attention mechanism to learn the weight of important information to further enhance its feature representation. (C) The Global Context Embedding Module (GCEM), which embeds high-level information into the low-level and guides the low-level learning. (D) After GCEM processing, low-level features are semantically enriched, which is conducive to the prediction of pixel categories. The upper layer is upsampled and then merged with the lower layer to generate the final result.

B. MODULE WITH SPLIT, CONCATENATE AND SPLIT OPERATIONS

Taking advantage of the channel separation and shuffling module, we propose a lightweight network residual unit, which is called the split, concatenate and split bottleneck structure (SCS-bt). The recent years have witnessed multiple successful instances of a lightweight residual layer, such as bottleneck unit (Figure 2(a)) [6], ShuffleNet unit (Figure 2(b)) [34] and SS-nbt module (Figure 2(c)) [35]. The point-by-point convolution in Figure 2(a) reduces many parameters, which is disadvantageous for the network model. The Shuffle Net unit in Figure 2(b) uses dense 1×1 point-by-point group convolution, which will affect the communication between channels. Figure 2(c) uses the classic channel split and shuffling operations to reduce computational complexity while also improving efficiency. Nevertheless, too many branches in this module will cause network discomfort.

To balance performance and efficiency under a limited budget, we introduce the asymmetric residual unit of the upper and lower streams(as shown in Figure 2(d)). At the beginning of each unit, the input is split into two lower-dimensional branches, where each one has half the channels of the input. First, the input channel of the left branch starts to convolve. The first step is to replace the point-by-point group convolution [36] with 1×1 convolution, because the channel split is equivalent to the grouping operation in disguised form. In the second step, a 3×3 depth

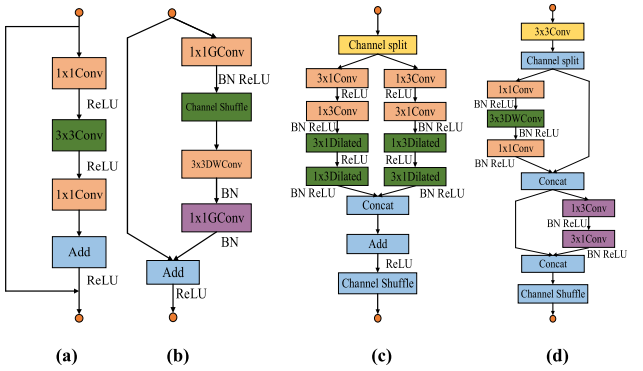

FIGURE 2. Comparison of different residual layer modules. (a) bottleneck, (b) ShuffleNet block, (c) SS-nbt block, (d) SCS-bt block.

separable convolution is used, instead of 3×3 ordinary convolution, which can save 8 to 9 times the amount of calculation; In the third step, we still adopt 1×1 convolution to recover the number of channels to cascade the information of the right branch. The module employs a channel cascade instead of element-by-element addition, which expands the channel and reduces the calculation cost. Then the result of the cascade is subjected to a second channel separation, and the input channel of the right branch starts to convolve. Because one-dimensional convolution can better extract features with less spatial information, the two-dimensional convolution (3×3) is converted into two one-dimensional convolutions (1×3 and 3×1) [37], and the convolution outputs of two branches are combined by concatenation, while keeping the same number of input and output channels. Finally, the shuffling operation is used to communicate information between channels, which can efficiently calculate all the features in all channels.

The designed SCS-bt adopts an asymmetric structure to make the network lighter and more efficient. In each SCS-bt unit, the merged feature channels are randomly shuffled and then join into the next unit. This can be regarded as a kind of feature reuse, to some extent, which enlarges network capacity without significantly increasing complexity.

As shown in Figure 3, inspired by Wang Yu et al [35], a $512 \times 512 \times 3$ picture is input into the network. Firstly, perform downsampling by stacking parallel outputs of a single 3×3 convolution (for batch specification and Relu nonlinear processing) and max-pooling. After downsampling, two residual blocks SCS-bt are superimposed to extract 1/2 feature map, and the feature map is a low-level detail branch containing more spatial details. Secondly, after the same convolution and down-sampling processing, three SCS-bt blocks are superimposed to extract a 1/4 feature map, which is called a transition branch. Finally, 1/8 of the feature map is extracted by superimposing eight SCS-bts, which is a high-level semantic branch with more abstract semantics. In these eight SCS-bt layers, we adopt dilated convolution [20], [38], [39] to collect more contexts. Using different dilation rates can increase the receptive field and collect different long-distance features.

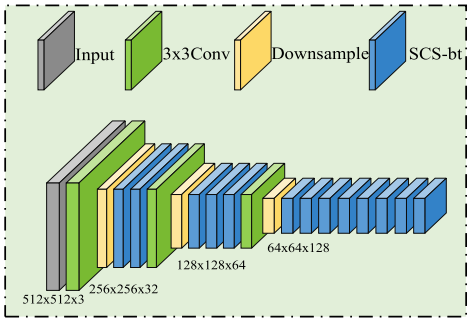

FIGURE 3. Lightweight encoder, which is constructed by superposition of scs-bt modules.

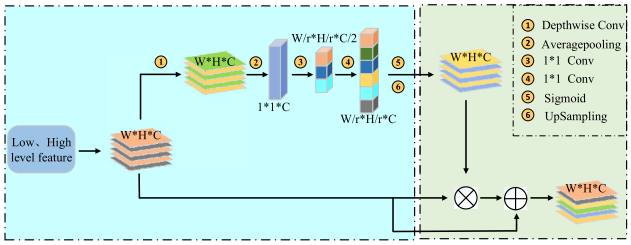

FIGURE 4. Detailed design of LAEM. From averaging pools to assigning attention weights to aggregate contextual information.

C. LOCAL ATTENTION ENHANCEMENT MODULE

Semantic segmentation of RSIs suffers greatly from the problem of intra-class inconsistency, since the distinction of object categories and the relationship between semantics cannot be obtained only from the appearance of the object, and needs to be obtained from nearby image data, image tags and other contextual information [40]. The reasonable use of contextual information can help us better accomplish tasks, but it also brings a big problem. As the amount of data increases or the correlation between images increases, it makes the scalability of the model relatively poor. The combined use of multiple contextual information can indeed get more accurate results. But the parameters for combination different levels of information are also very large. Therefore, how to design a model that can better combine multiple contextual information and develop more efficient algorithms is a research focus. In order to solve the problems of inconsistency within the class and large amount of contextual information parameters, we propose a Local Attention Enhancement Module (LAEM) to enhance the aggregation of contextual information in the extracted features.

The structure of the proposed LAEM module is shown in Figure 4. This module is inspired by SE-block's squeeze-Excitation module [24]. To solve the problem of a large amount of context information parameters, we perform a 3×3 depth separable convolution [13], [18], [41] at the beginning of the module. The depth separable convolution, which is proposed by MobileNet, is made up of two layers: depthwise convolutions and pointwise convolutions. We use depthwise convolutions to apply a single filter per each input channel (input depth). Pointwise convolution, a simple 1×1 convolution, is then used to create a linear combination of

depth layer outputs. Depthwise convolution with one filter per input channel (input depth) can be written as:

$$F_{s,t,m} = \sum_{i,j} S_{i,j,m} \cdot G_{s+i-1,t+j-1,m} \tag{1}$$

where S is a depthwise convolutional kernel with the size of $3 \times 3 \times 1$, and G represents an input feature map with a size of W×H×C, and the m_{th} filter in S is applied to the m_{th} channel in G to Produce the m_{th} channel of the filtered output feature map F.

Since the first step only generates independent channel features, an additional layer that computes a linear combination of the output of depthwise convolution via 1×1 convolution is needed to generate these new features. The 1×1 point-by-point convolution in this step can be written as:

$$D_{w,h,C} = \sum_{m} V_{1\times1,m,C} \cdot F_{w,h,m} \tag{2}$$

where V is a $1 \times 1 \times$ M convolution kernel, the i and j pixels of the m_{th} channel in F are convolved and summed with the pixels on the m_{th} channel in V to obtain a number, the feature map D is obtained by convolution of C convolution kernels V with the feature map F.

While SE-block transfers the converted features to the squeeze operation, it introduces global average pooling to generate one descriptor for each feature channel, thereby allowing information from the global receptive domain of the network to be used by all its layers. Nevertheless, our purpose is to extract local regional features, so regional average pooling (the size of pooling is Size $= W/2$, where W is the width of the input feature map of the layer, and the size of the input feature map is W×H×C) is used to get the local feature map of each channel, so that each channel contains local context information. Then the average pooling calculation for the C_{th} channel is:

$$A_{\text{C}} = \frac{1}{S \cdot S} \sum_{i,j=1}^{S} \text{D}_{\text{C}}(i,j), S = W/2 \tag{3}$$

where S is the size of the pooling window, D_{c} denotes a pixel at C_{th} channel, and the local information A_{c} of the C_{th} channel can be generated by formula 3. In order to apply the useful information summarized in the compression operation and capture the correlation between channels, we took the following actions:

$$a_c = U_p \{\delta \cdot C_u [\beta C_{\text{d}} (\lambda \beta A_c)]\} \tag{4}$$

where λ, β and δ denote respectively ReLU function, BatchNormal and Sigmoid function, C_{d} represents a 1×1 dimension-reduction convolution with a reduction ratio of r (r generally takes 2), C_u represents the 1×1 dimension-increased convolution that restores the number of channels to C, and U_p is the upsampling operation, Thus attention weight a_c is generated.

The attention weight is multiplied by the input feature, and then the result is summed with the input feature pixel by pixel to achieve the purpose of enhancement. This step can be expressed as:

$$O = \lambda (G + a_c \cdot G) \tag{5}$$

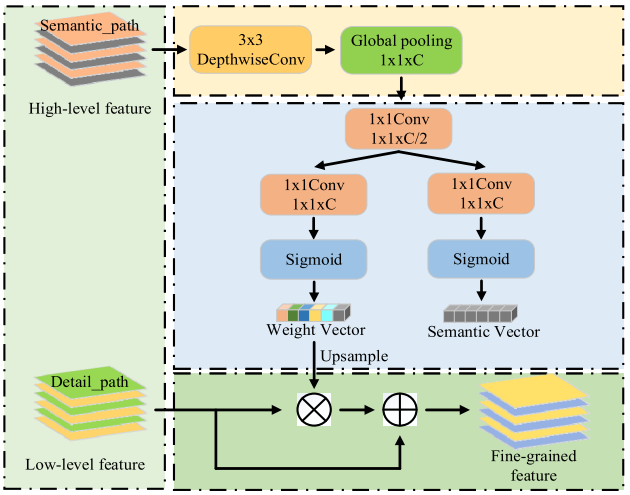

FIGURE 5. The architecture of the Global Context Embedded Module (GCEM). The high-level feature map and the low-level feature map respectively represent the semantic path and the detailed path enhanced by LAEM. C represents the number of categories.

D. GLOBAL CONTEXT EMBEDDED MODULE

The network proposed in this paper is a three-stage style encoder constructed by superposition of SCS-bt modules. According to our observation, the different stages have different cognitive abilities resulting in diverse consistency performance. In the lower stage, the spatial information of network coding is more refined. However, it has poor semantic consistency due to the small reception field and no guidance of spatial context. While in the high stage, it has strong semantic consistency because of the large acceptance field, but the prediction is spatially coarse. Generally speaking, the lower stage makes more accurate spatial predictions, while the semantic prediction in the higher stage is more accurate. So we propose a global context embedding module (GCEM), which embeds the weight vectors learned in high-level output into low-level features, and the learned weight vector is used to emphasize important details of low-level features. This information will increase the limits of the acceptance domain, while retaining low-level spatial information.

As shown in Figure 5, the module is divided into two branches, one branch is designed to change the weights of the features on each stage to enhance the consistency, and the weight vector is weighted to the low-level features to get the refined feature map. The other branch is used to generate semantic vector, which is used to calculate the loss of semantic multi-label category and further optimize the network.

As shown in formula 6, where the enhanced semantic path is subjected to depth separable convolution to obtain feature map D, and G represents global average pooling, and R represents reshape size of the feature map after global pooling. Next, we adopt the reduced dimensionality convolution C_1 compression channel, divide the compressed feature map into

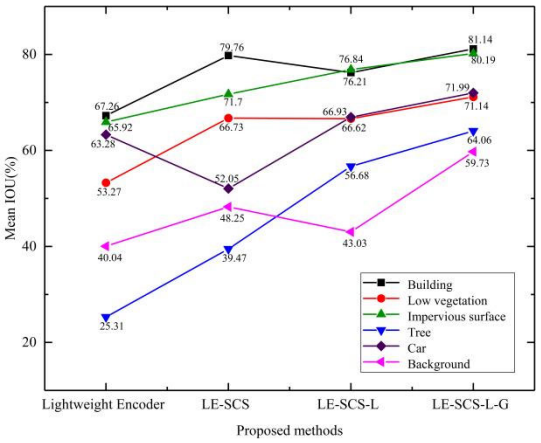

FIGURE 6. The average IOU of all categories of the proposed method on the Potsdam dataset.

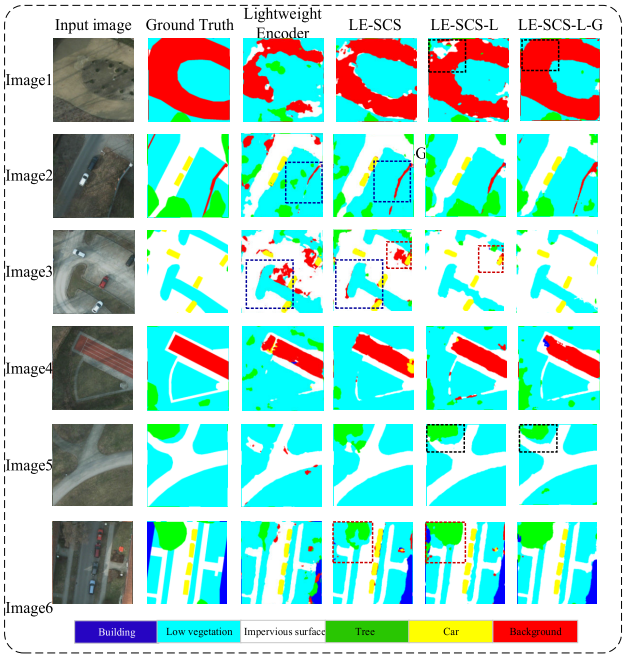

FIGURE 7. Qualitative visual results of our proposed methods on the Potsdam test set. Improved areas are marked with dashed boxes (zoomed-in view for more details).

two branches and perform an increased dimensionality convolution C_2 respectively, δ represents the Sigmoid function, U_p represents the upsampling operation, finally the weight vector γ_w is obtained, the same method is applied to obtain the semantic vector γ_s of another branch.

Since the learned weight vector is used to guide the low-level learning, The designed residual structure is used to refine the low-level L. The enhanced and refined low-level function A is obtained by formula 7:

$$\gamma_w = U_p \{\delta C_2 \{C_1 [R (G \cdot D)]\}\} \tag{6}$$

$$A = L + \gamma_1 \cdot L \tag{7}$$

E. LOSS FUNCTION

To better regulate the training process, a loss function covering the entire module should be designed. Besides, the traditional loss l_m and the multi-label category loss l_a proposed by this module are complementary to each other. On the one hand, l_m measures local pixel-wise training errors. On the other hand, l_a measures an overall loss.

Another branch of GCEM module obtains the semantic vector γ_s, which is used to calculate the semantic multi-label category loss l_a, and calculated by the following formula:

$$l_a = -\sum_{i=1}^{C} W_i \log \left(\gamma_{s(i)}\right) \tag{8}$$

where C represents the number of categories, W_i represents the category one-hot code calculated by the multi-label vector, and γ_s is the learned semantic vector of $1 \times 1 \times C$.

In this paper, we also adopt the traditional category cross-entropy loss [40], which is widely used in semantic segmentation. According to the traditional method, we define it as follows:

$$l_m = -\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{m=1}^{C} y\,(i,j,m) \log\,(p\,(i,j,m)) \tag{9}$$

where C represents the number of categories, W and H represent width and height, respectively. $p\,(i,j,m)$ and $y\,(i,j,m)$ represent the predicted value and the ground truth value. Throughout the training process, we take ρ as a hyperparameter to weigh the relationship between the main loss and the auxiliary loss, so the formula for the total loss L is as follows:

$$L = l_m + \rho l_a \tag{10}$$

IV. EXPERIMENTAL DATA SET AND EVALUATION

The experimental data includes two public data sets: Ultra-high resolution aerial images ISPRS Potsdam data set [42], Medium-resolution aerial images CCF data set [43]. In our experiments, we use three different criteria for quantitative assessment according to the data set guidelines, such as per-class average pixel-wise accuracy (Mean IoU), F1 score, and pixel accuracy (PA). In this section, we provide a brief description of both data sets and then present the design to provide experimental evaluation.

A. POTSDAM DATA SET

The Potsdam data set consists of 38 true ortho photos (TOP) and corresponding DSMs. These DSMs are collected from historical cities with large amounts of building blocks. There are four spectral bands in each TOP image (red, green, blue, and near-infrared) and one band in each DSM. The size of all images is 6000 × 6000 pixels, and the ground sampling distance (GSD) of this data set is 5 cm. The reference data are labeled according to six land-cover types: background, impervious surfaces, building, car, low vegetation and tree.

In our experiment, 10 pictures out of 16 available data blocks are used as training set and 6 pictures are used as validation set. Since the resolution cannot be too high during the training process, the large image is cropped into 512 × 512 color blocks, which enables the network to be trained in batches and saves computational costs.

Among them, 1500 patches are used to train the network, and the other patches are used to verify the proposed Module.

B. CCF DATA SET

The CCF data set is captured by a team in South China through drones, where containing four medium-resolution remote sensing images and their corresponding ground truths. There are four high-resolution images in the CCF data set: two with a resolution of 7969 × 7939, one with a resolution of 5664×5142, and the other with a resolution of 4011×2470. These images have 5 classes of high-quality pixel pole labels: background, water, road, vegetation, and building.

In our experiments, this data set also needs to be cut to obtain 512 × 512 small patches. We use 2390 patches to train the network and 783 patches to verify the proposed module.

C. EVALUATION

In this paper, we employ three different criteria for quantitative evaluation, such as F1 score, Mean Intersection over Union(MIoU) and pixel Accuracy (PA). The F1 score is a measure of the classification task, which is defined as the harmonic mean of precision and recall, and the maximum value is 1 and the minimum value is 0. Precision refers to the proportion of individuals whose prediction results belong to a certain category. Recall rate is the ratio between the number of individuals correctly predicted as a category and the total number of individuals of that category in the data set. The formulas for Precision, Recall and F1 score are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

where TP represents the number of true positives, TN represents the number of true negatives, FP represents the number of false positives, and FN represents the number of false negatives. For quantitative evaluation, we use MIoU as a measure of accuracy, which refers to the ratio of intersection and union of each category, and MIoU is the average IoU of all categories. Pixel accuracy PA is the simplest evaluation index for semantic segmentation, that is, the ratio of correctly predicted pixels to total pixels. The formula for the average MIoU and PA is as follows:

$$MIoU = \frac{TP}{TP + FP + FN} \tag{14}$$

$$PA = \frac{\sum_i^n TP_i}{\sum_i^n (TP_i + FP_i)} \tag{15}$$

where n is the number of target categories. i is the index of the target category.

V. EXPERIMENTAL DESIGN AND RESULTS

All implementations were trained on multiple Nvidia GeForce GTX 1080 Ti (11 GB) servers with CUDA 10.2, CUDNN 7.6.5, and GTX 1660Ti is used in the evaluation phase. Inspired by the previous works, we adopt the Adam optimizer [44] to optimize network, where the initial learning rate is set to 0.001 for each data set. In addition, the auxiliary loss function is a binary classification cross entropy loss function based on multi-label. The auxiliary loss function comprehensively considers the pixel level cross entropy and global level semantic information, so as to better optimize the whole segmentation model. In the training phase, the mini-batch gradient descent with the training batch size set to 2 [45], and all experiments were trained for 500 epochs. Batch normalization [46] is applied before each convolutional layer to prevent the gradient disappearing and exploding. In order to avoid overfitting, common data enhancements are applied before training the model, such as translation, rotation, noise increase, affine transformation, etc.

In this section, we show the numerical and visual results of our methods with different strategies on the Potsdam data set and CCF data set and compare them with other advanced technologies in the previous literature. The impact of each proposed strategy on the data set is quantitatively analyzed, and then the qualitative results of the test patches on the data set are displayed. More details reflected in the experimental results show that our proposed strategy can improve the accuracy of segmentation.

A. ABLATION EXPERIMENT ON POTSDAM DATA SET

This section gradually checks the effectiveness of each component in the network. In the following experiments, we take the lightweight encoder as the basic network for extracting features, which forms the baseline of the encoding-decoding form. We quantitatively evaluate the performance of the benchmark on the Potsdam data set. Table 1 shows the ablation experiments on the Potsdam data set, including the average IoU, pixel accuracy, and average metric of all categories. As shown in Figure 6, the MIoU of each category of each proposed strategy on the Potsdam data set is quantitatively analyzed in the form of a line chart. Figure 7 shows six visually segmented samples of each method for qualitative observation.

1) ABLATION STUDY FOR SCS-BT

The proposed SCS-bt block applies one-dimensional convolution pair and depth separable convolution to implement a lightweight feature extraction network. Our baseline network uses lightweight encoder as the fundamental network for extracting features, and the decoding part directly adopts progressive up-sampling for fusion output. We replace the designed SCS-bt block with the SS-nbt block in the baseline network to form Lightweight encoder(SCS-bt)(abbreviated as LE-SCS). It can be seen from Table 1 that in addition to trees and cars, other types of IoU have improved, and MIoU, PA, and F1s have increased by 6.2%, 7.57%, and 5.63%, respectively. Figure 6 shows that, except for cars, the average IoU of other categories obtained by LE-SCS on the Potsdam data set is higher than the average IoU obtained by Lightweight encoder. From the fourth column in Figure 7, it can be seen that compared to the baseline

TABLE 1. Performance comparison between SCS-bt, LAEM and GCEM. Concerning IoU for each class, mIoU, precision and F1. Highest score is marked with bold.

Model	Building	LowVeg	ImSurface	Tree	Car	Background	MIoU(%)	PA(%)	F1s(%)	Time(s)
Lightweight Encoder	67.26	53.27	65.92	25.31	63.28	40.04	52.51	67.58	67.41	0.19
LE-SCS	79.76	66.73	71.7	39.47	52.05	48.25	58.71	75.15	73.04	0.23
LE-SCS-L	76.21	66.62	76.84	56.68	66.93	43.03	64.38	80.08	77.68	0.26
LE-SCS-L-G	**81.14**	**71.14**	**80.19**	**64.06**	**71.99**	**59.73**	**71.37**	**83.79**	**83.05**	0.28

TABLE 2. Performance comparison between SCS-bt, LAEM and GCEM. Concerning IoU for each class, mIoU, precision and F1. Highest score is marked with bold.

Model	Background	Water	Vegetation	Road	Building	MIoU (%)	PA(%)	F1s(%)	Time(s)
Lightweight Encoder	55.61	62.11	36.46	82.04	52.93	57.83	80.58	72.18	0.19
LE-SCS	53.56	71.91	44.72	86.71	55.87	62.55	83.42	75.96	0.23
LE-SCS-L	63.1	75.89	47.9	88.64	61.78	67.46	86.55	79.76	0.26
LE-SCS-L-G	**64.85**	**77.23**	**48.01**	**89.32**	**62.31**	**68.34**	**87.11**	**80.37**	0.28

network, the boundary part is improved, which shows that the dilated convolution used when extracting features increases the receptive field and makes the extraction of details better.

2) ABLATION STUDY FOR LAEM.

The second experiment studies the contribution of the LAEM module to the segmentation effect, and LAEM is used to enhance extracted low and high-level feature representations. The LAEM module is added based on LE-SCS to form a LE-SCS+LAEM (abbreviated as LE-SCS-L). It can be seen from the Table 1 that after adding this module, the MIoU increases from 58.71% to 64.38%, the PA and F1 of LE-SCS-L are better than the first two ablation experiments. Figure 6 shows that the average IoU of buildings, background and low vegetation categories obtained by LE-SCS-L on the Potsdam data set is lower than the average IoU obtained by LE-SCS. From the fifth column in Figure 7, it can be seen that the network can distinguish small objects (trees, cars) and impervious surfaces well, and the segmentation of easily confused areas is improved. This shows that the added module dramatically enhances the representation ability of low-level features to understand detailed information better. Since high-level layers already have relatively large receptive field before using the LAEM, the enhancement effect is not apparent.

3) ABLATION STUDY FOR GCEM

We propose GCEM for two purposes. One is to embed high-level weight vectors into the low-level to guide low-level learning, and the other is to optimize the training network by calculating semantic multi-label category loss and traditional cross-entropy loss. The GCEM module is added based on LE-SCS-L to form a LE-SCS-L+GCEM (abbreviated as LE-SCS-L-G). Table 1 and Figure 6 show that the average IoU of all categories obtained by LE-SCS-L-G on the Potsdam data set is higher than the average IoU obtained by LE-SCS-L. From the sixth column in Figure 7, it can be seen that the boundary part has been optimized, and it is more accurate in recognition of large objects, which is enough to prove that the weight vector of the module allows the low-level features to learn more emphasized details, which is helpful for boundary extraction. To effectively merge,

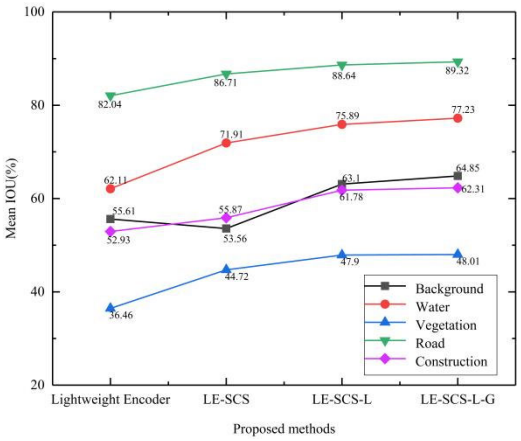

FIGURE 8. The average IOU of all categories of the proposed method on the CCF dataset.

make the fine-grained features in the shallow layer can perfectly capture the abstract features obtained by the high-level semantics, which is conducive to the understanding of the semantic context, so that the distinction of large objects is improved.

B. ABLATION EXPERIMENT ON CCF DATA SET

This section gradually checks the effectiveness of each component in the network. In the following experiments, we take the lightweight encoder as the basic network for extracting features, which forms the baseline of the encoding-decoding form. We quantitatively evaluate the performance of the benchmark on the CCF data set. Table 2 shows the ablation experiments on the CCF data set, including the average IoU, pixel accuracy, and average metric of all categories. As shown in Figure 8, the MIoU of each category on the CCF data set of each proposed strategy is quantitatively analyzed in the form of a line chart. Figure 9 shows six visually segmented samples of each method for qualitative observation.

1) ABLATION STUDY FOR SCS-BT

The designed SCS-bt block uses one-dimensional convolution pair and depth separable convolution to implement a lightweight feature extraction network. Our baseline network uses lightweight encoder as the fundamental network for

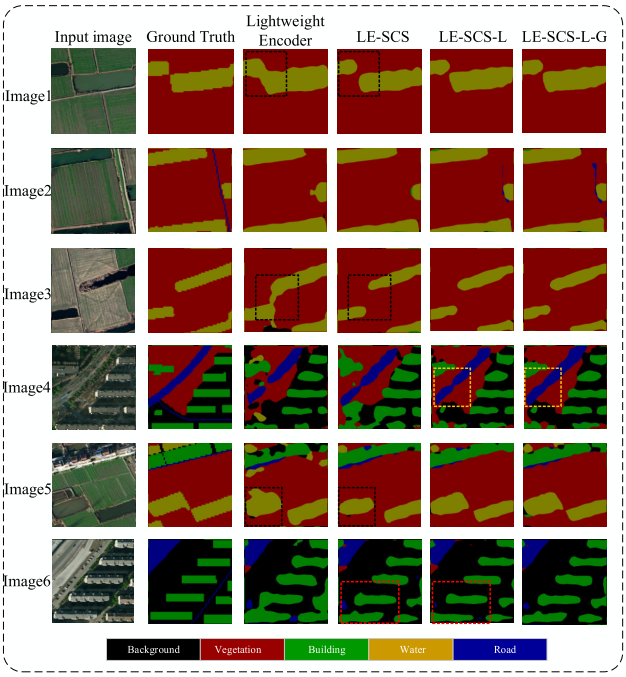

FIGURE 9. Qualitative visual results of our proposed methods on the CCF test set. Improved areas are marked with dashed boxes (zoomed-in view for more details).

extracting features, and the decoding part directly adopts progressive up-sampling for fusion output. We replace the designed SCS-bt block with the SS-nbt block in the baseline network to form Lightweight encoder(SCS-bt)(abbreviated as LE-SCS). It can be seen from Table 2 that in addition to the background, other types of IoU have improved, and MIoU, PA, and F1s have increased by 4.72%, 2.84%, and 3.78%, respectively. Figure 8 shows that, except for the background, the average IoU of other categories obtained by LE-SCS on the CCF dataset is higher than the average IOU obtained by lightweight encoder. The third and fourth columns in Figure 9 show that compared to the baseline network, the LE-SCS network has better feature extraction capabilities and more accurate segmentation.

2) ABLATION STUDY FOR LAEM

We also studied the contribution of the LAEM module to the segmentation effect. LAEM is used to enhance the extracted low-level and high-level feature representation. The LAEM module is added based on LE-SCS to form a LE-SCS+LAEM (abbreviated as LE-SCS-L). It can be seen from Table 2 that after adding this module, the IOU increases from 57.83% to 62.55%, the PA and F1 of LE-SCS-L are excellent in the first two ablation experiments. Figure 8 shows that the average IOU of all categories obtained by LE-SCS-L on the CCF data set is higher than the average IOU obtained by LE-SCS. As shown in the fifth column in Figure 9, the network can distinguish water and buildings well. However, it is not well recognized on vegetation and background in large areas. This shows that the added module dramatically enhances the representation ability of low-level features to understand detailed information better.

3) ABLATION STUDY FOR GCEM

We propose GCEM for two purposes. One is to embed high-level weight vectors into the low-level to guide low-level learning, and the other is to optimize the training network by calculating semantic multi-label category loss and traditional cross-entropy loss. The GCEM module is added based on LE-SCS-L to form a LE-SCS-L+GCEM (abbreviated as LE-SCS-L-G). It can be seen from Table 2 that the mIOU, PA, F1s (68.34%, 87.11%, 80.37%) with this module are better than the mIOU, PA, F1s (67.46%, 86.55%, 79.76%) of the previous experiment. Figure 8 shows that the average IOU of all categories obtained by LE-SCS-L-G on the CCF data set is higher than the average IOU obtained by LE-SCS-L. From the sixth column of 9, it can be seen that the boundary part has been optimized, and it is more accurate in recognition of large objects, and it is enough to prove that the weight vector of the module allows the low-level features to learn more emphasized details, which are helpful for boundary extraction. To effectively merge, make the fine-grained features in the shallow layer perfect the abstract features obtained by the high-level semantics, which is conducive to the understanding of the semantic context so that the distinction of large objects is improved.

C. COMPARISON WITH STATE-OF-THE-ART

The proposed method is compared with 4 models (including ICNet, LEDNet, BiseNet, and U-Net) on two data sets, and the comparison is made through the IoU, MIoU, parameters and time of each category.

Tables 3 and 4 report the quantitative results on the Potsdam data set and the CCF data set, respectively. Remarkably, Table 3 shows that on the Potsdam dataset, BARNet achieves 71.37% in mIoU. Compared with the other four methods, the mIoU increases by 11.39%, 9.59%, 4.49% and 2.35%, respectively. Table 4 shows that on the CCF data set, BARNet achieves 68.34% in mIoU. Compared with the other four methods, the mIoU increases by 8.39%, 5.02%, 3.6%, and 4.24%, respectively. As shown in Table 3, ICNet and LEDNet have good segmentation effect on building, low vegetation and impervious surface. ICNet accelerates the capture of semantics through low resolution, acquires details through high resolution, and merges features through cascaded networks. LEDNet adopts lightweight encoder for feature extraction, and the attention pyramid network is introduced in the decoding part to further improve the feature selection ability of the network. LEDNet is better than ICNet in the recognition of small objects (trees and cars), but both of them need to be strengthened in the segmentation of small objects. Compared with BiseNet, which is also a bilateral network, BARNet has less false alarms in the surrounding areas of buildings, which can be attributed to the embedding of contextual information. Meanwhile, the segmentation of small objects (cars, tree) is more accurate, which is due to

TABLE 3. Quantitative comparison with 4 latest technologies on the Potsdam test set.

Model	Building	LowVeg	ImSurface	Tree	Car	Background	MIoU(%)	parameters	Time(s)
ICNet	81.14	65.69	77.65	54.56	35.86	43.88	59.98	6.75M	0.98
LEDNet	76.0	64.54	75.8	60.12	58.75	35.5	61.78	3.66M	1.35
BiseNet	75.96	68.83	79.07	56.92	72.15	48.37	66.88	19.8M	1.21
U-Net	81.03	80.27	79.19	44.82	58.57	65.46	69.02	2.48M	0.44
BARNet(Ours)	82.25	71.14	80.19	64.06	71.99	59.73	71.37	1.26M	0.28

TABLE 4. Quantitative comparison with 4 latest technologies on the CCF test set.

Model	Background	Water	Vegetation	Road	Construction	MIoU (%)	parameters	Time(s)
ICNet	54.55	66.75	27.67	83.84	66.95	59.95	6.75M	0.98
LEDNet	56.3	72.33	44.83	86.05	57.07	63.32	3.66M	1.35
BiseNet	64.43	72.13	30.28	86.92	**69.96**	64.74	19.8M	1.21
U-Net	56.21	76.95	43.49	86.96	56.87	64.10	2.48M	0.44
BARNet(Ours)	**64.85**	**77.23**	**48.01**	**89.32**	62.31	**68.34**	**1.26M**	**0.28**

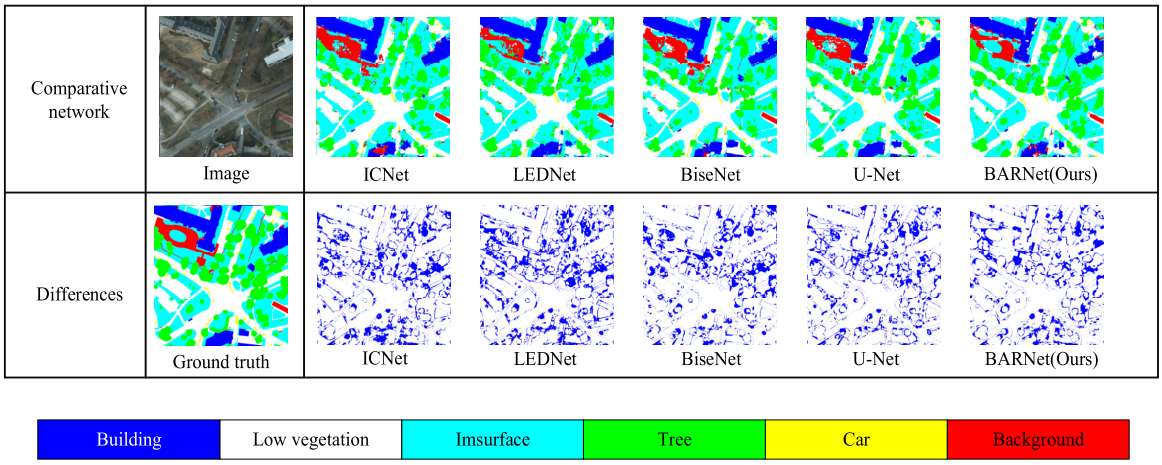

FIGURE 10. Qualitative visual comparison of the four latest methods (ICNet, LEDNet, BiseNet, U-Net) and the proposed BARNet on the Potsdam data set.

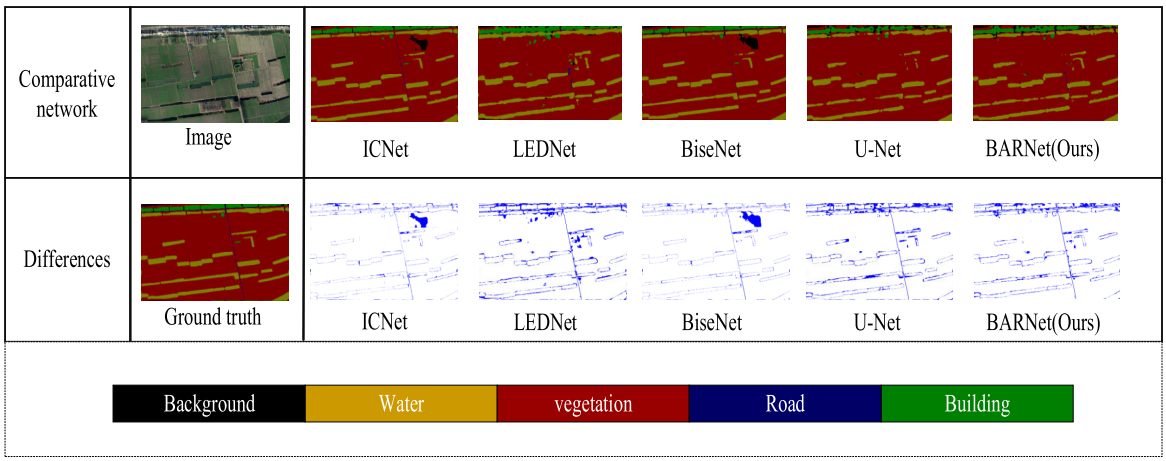

FIGURE 11. Qualitative visual comparison of the four latest methods (ICNet, LEDNet, BiseNet, U-Net) and the proposed BARNet on the CCF data set.

the incorporation of enhanced low level features. This points out that the proposed method improves both the discrimination of critical categories and the preservation of spatial details.

Simultaneously, the parameters and prediction time of our network are significantly reduced, which means that the processing speed of the model is faster. Figure 10 and Figure 11 show the intuitive comparison of the segmentation results of

BARNet and other models on the two data, respectively. The last line visually illustrates the difference between ground truth and prediction. The smaller the difference is, the better the segmentation effect is.

VI. CONCLUSION

We observe that the semantic segmentation task requires both low-level details and high-level semantics. We propose a new network architecture to deal with spatial details and categorical semantics separately, which is termed Bilateral Attention Refinement Network (BARNet). Firstly, the asymmetric SCS-bt unit realizes the lightweight and efficient feature extraction function. Secondly, we apply the local attention enhancement module (LAEM) to capture the detailed features and semantic features of remote sensing images to enhance their feature representation. Finally, to effectively integrate different levels of functions, this paper introduces the global context embedding module GCEM, which embeds attention from high-level layers into low-level ones to enrich their semantic information. Besides, the semantic vector obtained from the module is used to calculate the multi-label category loss, which is regarded as the auxiliary loss. The auxiliary loss function comprehensively considers the pixel level cross-entropy and global level semantic information to optimize the whole segmentation model better.

Experimental results on two RSIs data sets (Potsdam and CCF data sets) show that the proposed method remarkably improves the representation ability of extracted features. The use of local attention enhancement module is conducive for classifying the easily confused regions, while the embedding of attentions from high-level features to low-level ones improves the preservation of spatial details. However, one of the remaining problems in the semantic segmentation of RSIs is that how to identify large object regions and extract clear boundaries. Next, we will further optimize the enhancement module to improve the extraction and fusion of low and high-level features.

REFERENCES

[1] 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.

[2] 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.

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

[4] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: http://arxiv.org/abs/1706.05587

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

[6] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*. [Online]. Available: http://arxiv.org/abs/1606.02147

[7] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 405–420.

[8] 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," 2020, *arXiv:2004.02147*. [Online]. Available: http://arxiv.org/abs/2004.02147

[9] W. Jiang, Z. Xie, Y. Li, C. Liu, and H. Lu, "LRNNET: A light-weighted network with efficient reduced non-local operation for real-time semantic segmentation," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–6.

[10] G. Dong, Y. Yan, C. Shen, and H. Wang, "Real-time high-performance semantic image segmentation of urban street scenes," *IEEE Trans. Intell. Transp. Syst.*, early access, Mar. 19, 2020, doi: 10.1109/TITS.2020.2980426.

[11] F. Jiang, F. Guo, and R. Ji, "DSNET: Accelerate indoor scene semantic segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3317–3321.

[12] G. Li, S. Jiang, I. Yun, J. Kim, and J. Kim, "Depth-wise asymmetric bottleneck with point-wise aggregation decoder for real-time semantic segmentation in urban scenes," *IEEE Access*, vol. 8, pp. 27495–27506, 2020.

[13] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9522–9531.

[14] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Efficient ConvNet for real-time semantic segmentation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1789–1794.

[15] J.-Y. Sun, S.-W. Jung, and S.-J. Ko, "Lightweight prediction and boundary attention-based semantic segmentation for road scene understanding," *IEEE Access*, vol. 8, pp. 108449–108460, 2020.

[16] L. Wang, Q. Xu, Z. Xiong, Y. Huang, and L. Yang, "A multi-level feature fusion network for real-time semantic segmentation," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019, pp. 1–6.

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

[18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[20] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 552–568.

[21] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9190–9200.

[22] E. Romera *et al.*, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, 2017.

[23] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3146–3154.

[24] 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.

[25] H. Li, K. Qiu, and L. Chen, "SCAttNet: Semantic segmentation network with spatial and channel attention mechanism for high-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, early access, Apr. 29, 2020, doi: 10.1109/LGRS.2020.2988294.

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

[27] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[28] H. Xue, C. Liu, F. Wan, J. Jiao, X. Ji, and Q. Ye, "DANet: Divergent activation for weakly supervised object localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6589–6598.

[29] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7151–7160.

[30] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding, "ACFNet: Attentional class feature network for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6798–6807.
[31] R. Niu, X. Sun, Y. Tian, W. Diao, K. Chen, and K. Fu, "Hybrid multiple attention network for semantic segmentation in aerial images," 2020, *arXiv:2001.02870*. [Online]. Available: http://arxiv.org/abs/2001.02870
[32] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, *arXiv:1805.10180*. [Online]. Available: http://arxiv.org/abs/1805.10180
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[34] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[35] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, and L. J. Latecki, "Lednet: A lightweight encoder-decoder network for real-time semantic segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1860–1864.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[37] X. Ding, Y. Guo, G. Ding, and J. Han, "ACNet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution blocks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1911–1920.
[38] G. Li, I. Yun, J. Kim, and J. Kim, "DABNet: Depth-wise asymmetric bottleneck for real-time semantic segmentation," 2019, *arXiv:1907.11357*. [Online]. Available: http://arxiv.org/abs/1907.11357
[39] T. Ziegler, M. Fritsche, L. Kuhn, and K. Donhauser, "Efficient smoothing of dilated convolutions for image segmentation," 2019, *arXiv:1903.07992*. [Online]. Available: http://arxiv.org/abs/1903.07992
[40] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1857–1866.
[41] R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Fast-SCNN: Fast semantic segmentation network," 2019, *arXiv:1902.04502*. [Online]. Available: http://arxiv.org/abs/1902.04502
[42] *ISPRS Vaihingen 2D Semantic Labeling Dataset*. Accessed: Apr. 5, 2018. [Online]. Available: http://www2.isprs.org/commissions/comm3/wg4/2dsem-label-vaihingen.html
[43] *The Fifth AI Classification and Recognition Competition. Challenge of AI on Satellite Imaging*. Accessed: Sep. 6, 2017. [Online]. Available: https://www.datafountain.cn/competitions/270/details?tdsourcetag=spctimaiomsg
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[45] 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)*, 2018, pp. 325–341.
[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

JIALI CAI received the B.Eng. degree from Lanzhou Jiaotong University, Lanzhou, China, in 2019, where she is currently pursuing the M.S. degree. Her research interests include computer graphics, computer vision, machine learning, and image processing.

CHUNJUAN LIU received the B.Eng. degree from Lanzhou Jiaotong University, Lanzhou, China, in 2004. She is currently an Associate Professor with Lanzhou Jiaotong University. Her research interests include electronic technology applications, integrated circuit technology, and special semiconductor devices.

HAOWEN YAN received the B.Eng. degree from the Wuhan Technical University of Surveying and Mapping, Hubei, China, in 1991, and the M.Sc. and Ph.D. degrees from Wuhan University, Hubei, in 2002.

From 2005 to 2012, he was a sub-decanal with the School of Mathematics and Software Engineering, Lanzhou Jiaotong University, Lanzhou, Gansu, China. Since 2012, he has been a Dean of the School of Surveying, Mapping and Geographic Information, Lanzhou Jiaotong University.

XIAOSUO WU received the B.Sc. and M.S. degrees from Lanzhou Jiaotong University, Lanzhou, China, in 2004, and the Ph.D. degree from Lanzhou University, Lanzhou, in 2011. He is currently pursuing the M.S. degree with Lanzhou Jiaotong University. His research interests include microelectronics, solid-state electronics, computer vision, machine learning, and image processing.

WANZHEN LU received the B.Eng. degree from Hainan University, Hainan, China, in 2017. He is currently pursuing the M.S. degree with Lanzhou Jiaotong University, Lanzhou, China. His research interests include computer graphics, computer vision, machine learning, and image processing.

XIAOYU WANG received the B.Eng. degree from Hebei University, Hebei, China, in 2019. He is currently pursuing the M.S. degree with Lanzhou Jiaotong University, Lanzhou, China. His research interests include computer graphics, computer vision, machine learning, and image processing.

CHANGLIN SANG received the B.Eng. degree from the Lanzhou University of Technology, Lanzhou, Gansu, China, in 2017. He is currently pursuing the M.S. degree with Lanzhou Jiaotong University, Lanzhou. His research interests include computer graphics, computer vision, machine learning, image processing, and integrated optics.