SNLRUX++ for Building Extraction From High-Resolution Remote Sensing Images

Yanjing Lei, Jiamin Yu, Sixian Chan, Wei Wu, and Xiaoying Liu

***Abstract*—Building extraction plays an important role in high-resolution remote sensing image processing, which can be used as the basis for urban planning and demographic analysis. In recent years, many powerful general semantic segmentation models have emerged, but these models often perform poorly when transferred to remote sensing images because of the characteristics of remote sensing images. To this end, we propose a new deep learning network called Selective Nonlocal ResUNeXt++ (SNLRUX++) for building extraction. First, the cascaded multiscale feature fusion is proposed to transform the high-performance image classification network ResNeXt into the segmentation network ResUNeXt++. Second, selective nonlocal operation is designed to establish long-range dependencies while avoiding introducing excessive noise and computational effort. Finally, multiscale prediction is applied as deep supervision to accelerate training and convergence, and improves prediction performance of objects at different scales. The experimental results on two different remote sensing image datasets show the effectiveness and generalization ability of the proposed method.**

***Index Terms*—Building extraction, convolution neural network, deep learning, high-resolution image, remote sensing.**

I. Introduction

With the rapid development of remote sensing technology, the amount of high-resolution remote sensing image data has been increasing. On the one hand, the maturity of aerospace technology has made it easier to acquire large-scale, high-quality remote sensing images. On the other hand, with the development of imaging technology, the spatial resolution, spectral resolution, and temporal resolution of remote sensing images have been greatly improved. The increase in the quantity and quality of remote sensing images has made it possible to form and improve the Earth observation system, and to continuously monitor the earth's surface through remote sensing. At present, remote sensing technology has been widely used in industry, agriculture, military, economy, and other fields. For example, resource exploration [1], crop classification [2], pest monitoring [3], military target detection [4], urban planning [5], land use analysis [6], and disaster warning [7]. With the increase of remote sensing image data volume and resolution, the demand for remote sensing image processing and information extraction technology is also growing.

Building extraction is one of the important tasks of remote sensing image segmentation. In recent years, there are an increasing number of models that use deep learning methods for semantic segmentation. Mainstream deep learning segmentation networks are typically based on UNet [8] architecture using a fully symmetric encoder–decoder structure. Their main components include downsampling, upsampling and skip connection. The network encodes the image through downsampling, compressing the image into a latent-space representation, which contains semantic information useful for prediction. In the decoding part, the decoder uses the compressed feature representation by upsampling to recover the resolution and make prediction. In the skip connection part, after upsampling, the deep feature maps are concatenated with the shallow feature maps of the same resolution, which alleviates the losing of location information caused by the encoding to some extent. In such a network structure, the deep feature maps extract sufficient semantic information to be better used for prediction, but due to the low resolution, the object boundary cannot be well localized; the shallow feature maps retains higher resolution, so it can locate objects more accurately, but the lack of semantic information may lead to prediction errors. How to use features of different scales, fuse high-level semantic information and low-level location spatial information to construct high-resolution, high-semantic feature maps, and improve network prediction performance is a problem worthy of study.

To construct high-resolution and high-semantic feature maps, many semantic segmentation methods focus on increasing the receptive field, the most famous of which are DeepLab [9] and DilatedNet [10], to obtain contextual information and establish long-range dependencies. They use dilated convolution to avoid downsampling and improve the receptive field while maintaining the resolution of the feature maps. Another way to construct feature maps with contextual semantic information is to use nonlocal operation [11]. Nonlocal is a simple and general operation for capturing long-range dependencies in deep neural networks. In simple terms, the nonlocal operation obtains the feature representation of each position by performing a weighted summation of all the position features of the input. In computer vision, this position usually refers to each pixel. The advantage of the nonlocal operation over the dilated convolution is that it directly establishes long-range dependencies by computing

Manuscript received September 29, 2021; revised November 4, 2021; accepted December 12, 2021. Date of publication December 15, 2021; date of current version December 30, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61906168 and Grant 61902351 and in part by the Natural Science Foundation of Zhejiang Province under Grant LY21F020023. *(Corresponding author: Sixian Chan.)*

The authors are with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: leiyj@zjut.edu.cn; 876319691@qq.com; sxchan@zjut.edu.cn; wuwei@zjut.edu.cn; xiaoyingliu@zjut.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2021.3135705

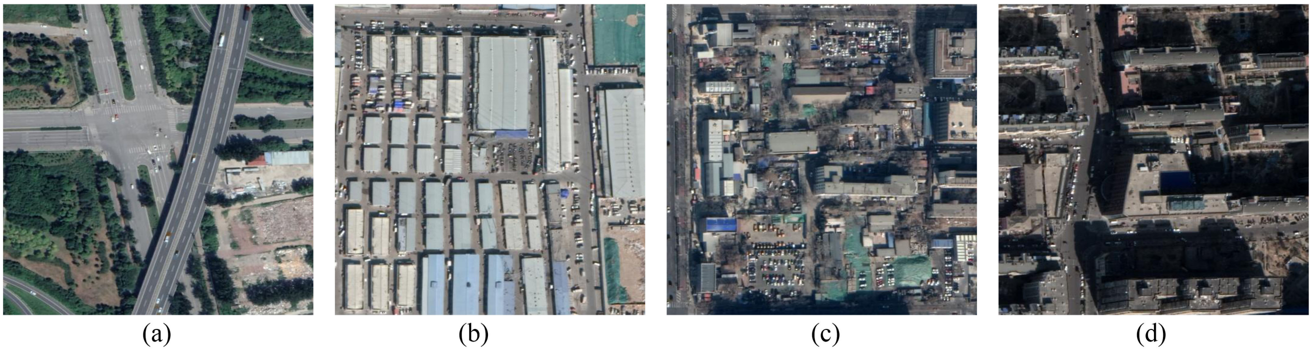

Fig. 1. Examples of remote sensing images with different type of challenges. (a) Imbalance between foreground and background. (b) Small and numerous objects. (c) Complex and diverse foreground and background. (d) Shadow occlusion.

the similarity between two positions, regardless of the distance between them.

Although these methods have shown good performance in natural image segmentation, there are many difficulties when applying them to remote sensing image datasets. In addition to the problem of varying scales of objects in most semantic segmentation datasets, remote sensing images also suffers from problems such as imbalance between foreground and background [12], small and numerous objects [13], complex and diverse foreground and background, and shadow occlusion [14]. Some typical remote sensing images and challenges are shown in the Fig. 1. In Fig. 1(a), the number of pixels of the buildings is only a small part of the whole image, which is often found outside urban areas, and such images are prone to false positives. In Fig. 1(b), the image contains very dense buildings, each occupying very few pixels, and the buildings are very close to each other, so the boundaries are often not accurately located. In Fig. 1(c), there are many subcategories within buildings such as residential buildings, office buildings, shopping malls, and schools. Different buildings have different sizes, heights, colors, and forms, showing large intraclass differences. There are also many subcategories within the background, such as roads, low vegetation, trees, and rivers. Complex and diverse foreground and background can cause some methods to introduce excessive noise, leading to performance degradation. In Fig. 1(d), some tall buildings produce long shadows at certain times of day, and areas covered by shadows are prone to false negatives.

Downsampling will lose location information. Also, since most objects in remote sensing images are very small, the enlarged receptive field will contain more complex and diverse background information, so the introduction of noise will lead to performance degradation. In other words, it is not that deeper semantic information is more useful for prediction. In the recent literature, some scholars have experimented with UNet-8 s (stride stands for the aspect ratio of the original image to the minimum feature map) and obtained better performance than the original UNet-16 s with deeper depth and more parameters [15]. For different datasets, the requirements for network depth are different. As for nonlocal operation, due to the high resolution of remote sensing images, performing nonlocal operation in the whole image will incur huge computational costs. Because the foreground and background are unbalanced and the background is complex and diverse, performing nonlocal operation directly can cause the contextual information overwhelming by background noise. Moreover, the need for dense prediction of each pixel makes manual annotation both time-consuming and costly, rendering remote sensing segmentation datasets much smaller than classification or detection tasks [16]. Complex and high-capacity models are more likely to overfit due to the scarcity of training samples.

In this article, we propose a novel network called Selective Nonlocal ResUNeXt++ (SNLRUX++) for remote sensing building extraction to address the problems stated previously. Our network is based on ResNeXt [17], a powerful network for image classification task. First, we follow the encoder–decoder architecture in UNet, add decoders and skip connections to it, and therefore converting it to ResUNeXt for the segmentation task. Then, we use the cascaded multiscale feature fusion method to fuse features at different scales to obtain high-resolution and high-semantic feature representation. Since the traditional skip connections are replaced by residual connections, we can use only short connections to reduce the network complexity and alleviate the overfitting problem caused by the small remote sensing dataset. Besides, if useful information cannot be learned when the network goes deeper, it can still maintain the original information, alleviating the problem of losing location information. Second, we use selective non-local operation to extract key points from the feature maps, perform nonlocal operation between these key points, and propagate contextual information near the key points by convolution operation to capture long-range contextual dependencies. In addition to obtaining global contextual information, the advantages of selective nonlocal operation include: the computational cost is greatly reduced because the nonlocal operation are performed at only a small number of positions; key point extraction resamples the foreground and background ratios, reducing the noise introduced by the nonlocal operation; the module does not change the resolution of the feature map, so it can be easily plugged into various positions of the network, enhancing the feature map representation. Finally, we use multiscale prediction methods to deeply supervise the network. The feature maps at different scales are upsampled to the highest resolution, the semantic information at different scales is merged, and the predicted masks are output through the prediction head. The advantages of this approach include enabling the feature maps of the intermediate layers, especially the shallow ones, more transparent and making the features learned in the intermediate layers more discriminative and robust. In addition, it can accelerate the convergence of the network and alleviate training problems such as gradient disappearance that may be caused by the depth of the network.

In general, the main contribution of this article are as follows.

1) We propose a cascaded multiscale feature fusion method to fuse semantic information multiple times and thus extend the high-performance backbone ResNeXt in image classification task to construct extraction task.
2) We design a selective nonlocal operation, which extracts the key points from the feature maps, performs nonlocal operation only between these key points, and follows a convolution block to obtain a feature representation containing global contextual information.
3) We apply multiscale prediction to achieve deep supervision, so the network can adaptively adjust the prediction weights at different scales according to the scale of the object, thus improving the performance of objects at different scales.

The rest of this article is organized as follows. Section II reviews the previous work on general semantic segmentation and the improvements made by scholars to use it in remote sensing images. Section III introduces the proposed network SNLRUX++ for remote sensing building extraction task in detail. Section IV introduces the datasets and experimental details, and discusses the experimental results. Finally, Section V concludes this article.

II. Related Work

A. General Semantic Segmentation

Semantic segmentation has been a very fundamental problem in computer vision [18]. Before the maturity of deep learning, scholars used traditional methods including thresholding [19], region growing [20], k-means clustering [21], and more advanced algorithms such as graph cuts [22], superpixel methods [23], sparsity-based methods [24], and conditional and Markov random fields [25]. In recent years, scholars have gradually shifted their attention to deep learning methods because their performance has been significantly improved compared to traditional methods.

FCN [26] is an important milestone in deep learning semantic segmentation. Based on image classification networks such as AlexNet [27], VGGNet [28], and GoogLeNet [29], it removes the final fully connected layer and adopts a fully convolutional architecture, allowing it to perform dense prediction task on inputs of any size. UNet [8] and SegNet [30] follow the connection paradigm used by FCN and reorganize the decoder section to achieve better performance. DeepLab [9] used dilated convolution to address the problem of decreasing network resolution caused by maxpooling and striding, and then uses atrous spatial pyramid pooling, which probes the feature map with filters of multiple sampling rates, thus capturing multiscale context to robustly segment objects at multiple scales. Another method once used by DeepLab but later abandoned is the postprocessing of object boundary using a fully connected CRFs. Feature pyramid network (FPN) [31] was originally developed for object detection, it can easily be used for segmentation as well [32]. In order to generate segmentation output from FPN multilevel features, a simple design is used to merge information from each level of the FPN pyramid into a single one. The merging is done by upsampling features of different FPN levels several times to the lowest level and merging them using an elementwise addition method, and then making predictions after the merging. Pyramid scene parsing network [33] is a multiscale network that focuses on better learning of global contextual representation. It uses a pyramid pooling module to extract different subregion representation, and upsamples the output of multilevel pooling and connects with the initial feature map to capture local and global contextual information. UNet++ [34] is a deeply-supervised encoder–decoder network in which the encoder and decoder subnetworks are connected by a series of nested, dense skip pathways. HRNet [35] maintains high-resolution representation throughout by connecting multiresolution subnetworks in parallel and iteratively exchanging the information across different resolutions. UNet 3+ [36] uses full-scale skip connections, incorporating low-level details with high-level semantics from feature maps at different scales, to replace the dense skip pathways used in UNet++. The abovementioned model lays the foundation for remote sensing image segmentation and explores different feature fusion methods. Although these models show good performance in natural images or medical images, there is still much room for improvement when transferring them to remote sensing segmentation due to the characteristics of remote sensing images.

B. Remote Sensing Segmentation

According to the characteristics of remote sensing images, many scholars have made targeted adjustments to the network architecture. Michael *et al.* [13] proposed a network focusing on the accuracy of small objects to alleviate the problem of class imbalance. The architecture includes patch-based method and pixel-to-pixel method, and then combine their strengths using model ensemble. And the uncertainty map is used to indicate that the difficulty of remote sensing segmentation is the boundary of the object. building residual refine network [37] consists of two parts, namely the prediction module and the residual refinement module. Among them, the prediction module based on an encoder–decoder structure introduces dilated convolution of different dilation rates to extract more global features. While the residual refinement module takes the output of the prediction module as input, and further refines the residual between the result of the prediction module and the real result. Zhang *et al.* [38] enhanced the low-to-high features extracted from different branches of HRNet to enhance the embedding of scale-related contextual information. The low-resolution branches are incorporated in a spatial-reasoning module to learn the long-range spatial correlations, while the high-resolution branches are enhanced by adaptive spatial pooling module to aggregate local contexts. Cheng *et al.* [39] proposed an end-to-end cross-scale feature fusion (CSFF) framework, which is used for detection but can be easily applied to segmentation as well. CSFF uses FPN to obtain multilevel feature maps and then inserts a squeeze and excitation block at the top layer to model the relationship between different channels. The feature maps of all stages are then passed into the CSFF module to fuse different scale information to obtain a multilevel feature representation. Chen *et al.* [40] proposed an improved semantic segmentation network based on DeepLabv3 with addition augmented atrous spatial pyramid pool and FC fusion path layers to deal with the problems of ambiguous classification and unclear boundary of small objects caused by the characteristics of the remote sensing images. DenseU-Net [41] builds on UNet by replacing the VGG blocks with Dense connections [42] in both the downsampling and upsampling sections to enhance the feature extraction capability of the network. And uses the focal loss weighted by the median frequency balancing to improve predication accuracy of the small object classes. Based on the feature pyramid network (FPN) framework, PFNet [43] uses a module called PointFlow to propagate semantic information from high to low features at salient and edge points. The dual point matcher module is designed to extract these salient and edge points. And in this way, it solves the problems of foreground-background imbalanced distribution and multiple small objects in remote sensing images. D-CNNs [44] uses metric learning method to solve the problem of within-class diversity and between-class similarity in remote sensing images by adding metric learning regularization term to the objective function to supervise the learned features to be more discriminative. FENet [45] proposes the DAFE module and the CFE module, where DAFE module is used to highlight the network to focus on the distinctive features of the objects of interest and suppress useless ones, and CFE module is used to capture global context cues and selectively strengthen class-aware features. In summary, most remote sensing image segmentation models are exploring methods for obtaining multiscale feature maps, multiscale fusion methods, attention mechanisms for enhancing feature representation, and postprocessing methods to deal with the characteristics of remote sensing images.

III. Method

In this section, we will introduce in detail the Selective Nonlocal ResUNeXt++ (SNLRUX++) for remote sensing building extraction proposed in this article. Including: (see Section III-A) The backbone ResNeXt [17]; (see Section III-B) Using the cascaded multiscale feature fusion method, ResNeXt used for classification task is transferred to ResUNeXt++ for segmentation task; (see Section III-C) Use selective nonlocal operation to introduce global contextual information in the feature maps of all stages of the decoder; (see Section III-D) Predict on feature maps at different scales to achieve deep supervision. The overall structure of the network and the location of each module is shown in the Fig. 2.

A. Resnext

SNLRUX++ uses ResNeXt as its backbone and uses the final feature map of each stage in the downsampling. On the basis of ResNet [46], ResNeXt introduces the "split-transform-merge" aggregation transformation used in Inception [47], a network architecture that uses multibranch parallel convolution and then merges the results of each branch. Typical ResNet block, Inception block and ResNeXt block are shown in the Fig. 3. Unlike Inception, the size and number of convolution filters in each parallel branch have been carefully designed and experimentally adjusted. ResNeXt uses the same topology on each branch, so the structure can be reshaped into grouped convolution used in AlexNet [27]. In addition, the difference also includes that ResNeXt uses an addition operation to merge the results of different branches, while Inception uses the channel-dimensional concatenation.

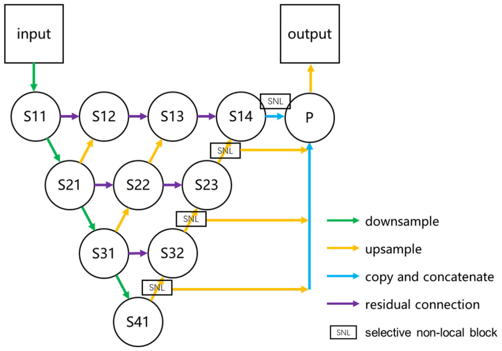

Fig. 2. Structure of SNLRUX++.

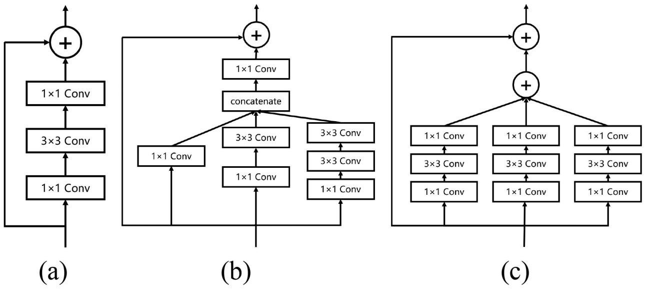

Fig. 3. Typical convolution block. (a) ResNet. (b) Inception. (c) ResNeXt.

In ResNeXt, the introduced hyperparameter is called "cardinality," which means the number of parallel transformation branches. Because all branches of ResNeXt use the same topology, there is no need to adjust the size and number of convolution filters, thereby reducing the number of hyperparameters in the model and improving the generalization ability of the network in different datasets. In addition to adjusting the width and depth, it provides a new way to adjust the capacity of the network model. Experiments show that increasing the "cardinality" can improve the network performance more effectively than increasing the width and depth for the same amount of computation, especially when the marginal benefit of increasing the depth gradually diminishes.

Formally, aggregation transformation can be expressed as follows:

$$\mathcal{F}(\mathbf{x}) = \sum_{i=1}^{C} \mathcal{T}_i(\mathbf{x}) \tag{1}$$

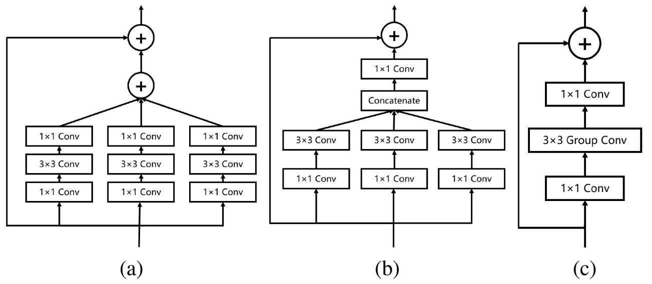

Fig. 4. Three equivalent forms of ResNeXt block.

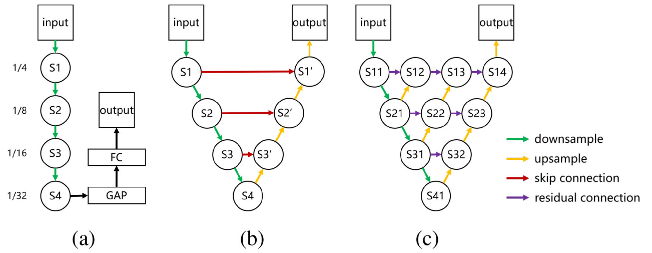

Fig. 5. Transformation between the three models. (a) ResNeXt. (b) ResUNeXt. (c) ResUNeXt++.

where $\mathbf{x}$ represents the input feature, $\mathcal{T}_i(\mathbf{x})$ represents the parallel transformation function, which can be any function. Similar to a simple neuron, $\mathcal{T}_i(\mathbf{x})$ should project $\mathbf{x}$ into an usual low dimensional embedding and then transform it. C is the number of the transform set to be aggregated, which is cardinality in ResNeXt. C is used to control the complexity and capacity of the network.

In ResNeXt, first, a simple transformation function design is used, where all $\mathcal{T}_i$ share the same topology. This follows the VGG style strategy of repeating the same structure, making it possible to control the network capacity with only a few hyperparameters. Second, the individual transformation $\mathcal{T}_i$ is set as a bottleneck structure. In this case, the first 1×1 layer in each $\mathcal{T}_i$ is used to produce low-dimensional embeddings. Third, use residual connections to establish direct path between inputs and outputs. In general, the aggregation transformation in ResNeXt can be expressed as

$$\mathbf{y} = \mathbf{x} + \sum_{i=1}^{C} \mathcal{T}_i(\mathbf{x}) \tag{2}$$

where $\mathbf{y}$ is the ResNeXt block output.

ResNeXt block can be expressed in three equivalent forms, as shown in the Fig. 4. In this article, we use the form of Fig. 4(c), because of its concise expression and easy implementation.

B. Resunext++

ResNeXt is used for image classification task, which is illustrated in Fig. 5(a), and the width and height of its last stage (stage is used to distinguish the resolution of different feature map) feature map is 1/32 of the original image, which is too coarse for building extraction task. Therefore, following the encoder-decoder architecture used in UNet and FCN, we removed the last global average pooling and FC layer of ResNeXt. Starting from the last stage, the feature maps are continuously upsampled, and merged with the feature map of corresponding stages through skip connection to restore the resolution of the feature maps. By now the network model is transformed into ResUNeXt (note the U) for building extraction task, the network architecture is illustrated in Fig. 5(b). For the sake of simplicity, we only show the upsampling, downsampling and skip connection parts in the figure. Each stage is represented by S, and the ResNeXt blocks in each stage are not drawn in detail.

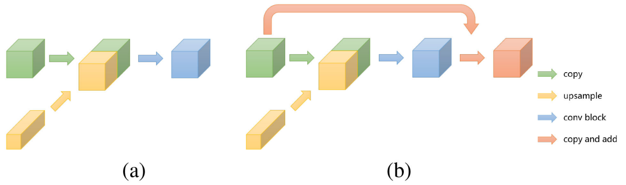

Fig. 6. Difference between the two types of connections. (a) Skip connection. (b) Residual connection.

In ResUNeXt, since the initial stride convolution and max-pooling, the width and height of the feature map with the highest resolution are 1/4 of the original image. Unlike UNet, the network makes predictions at this scale and uses bilinear interpolation to upsample the predictions to the size of the original image to obtain the final prediction mask. Because of the regularity of the boundary of the building object, the prediction under the feature map with lower resolution than the original image will not be seriously damaged.

Inspired by UNet++ [34], UNet 3+ [36], and HRNet [35], we redesigned the skip connections between the encoder and decoder, and used cascaded multiscale feature fusion method to obtain high-resolution and high-semantic feature maps for building extraction task. We call ResUNeXt that combines cascaded multiscale feature fusion as ResUNeXt++, and its structure is illustrated in Fig. 5(c).

In UNet, the feature maps of the encoder are directly copied and concatenated with the feature maps of the corresponding decoder, and then convolution is used to reduce the number of channels. The feature maps of each stage in the encoder will be fused only once, which is detrimental to the shallow stages. We believe that for shallow stage feature maps, since they are the most lacking in semantic information, they should be fused several times, and each fusion can improve the semantic information.

In ResUNeXt++, the feature maps of the encoder undergo multiple feature fusions, and the number of fusions depends on the stage level. For example, the shallowest feature map will undergo three feature fusions, while the penultimate stage feature map will only undergo one feature fusion. UNet++ also concatenates the previous layers and lower stage features, and the architecture of UNet++ has even more connections than ResUNeXt++ showed in Fig. 5(c). Following the ResNet design, we use residual connections instead of dense skip connections. The difference between skip connection and residual connection is illustrated in Fig. 6. The advantage of residual connection is that the feature maps ensure the retention of valid information and avoids the noise introduced by feature fusion, if the deeper stage of the feature map does not learn meaningful semantic information, which often happens in remote sensing images. For these reasons, it is natural to remove the dense skip connections used in UNet++ and keep only short ones. Therefore, the complexity of the parameters and calculations of the model are reduced, as well as the possibility of overfitting.

Formally, we formulate the residual connection as follows: let $x^{i,j}$ denote the output of node $X^{i,j}$ where i indexes the stage along the encoder and j indexes the intermediate residual convolution block of cascaded feature fusion along the residual connection pathway. The stack of feature maps represented by $x^{i,j}$ is computed as

$$\mathbf{x}^{i,j} = \begin{cases} \mathcal{R}\left(\mathbf{x}^{i-1,j}\right), & j = 0 \\ \mathbf{x}^{i,j-1} + \mathcal{H}\left(\left[\mathbf{x}^{i,j-1}, \mathcal{U}\left(\mathbf{x}^{i+1,j-1}\right)\right]\right), & j > 0 \end{cases} \tag{3}$$

where function $\mathcal{R}(\cdot)$ is stage layer in ResNeXt, which contains multiple ResNeXt blocks according to the network capacity. $\mathcal{H}(\cdot)$ is channel reduction convolution block, which consists of multiple groups of convolution operation, batch normalization and ReLU, and [ ] denotes the concatenation operation. Basically, nodes at level $j = 0$ only receive one input from the previous stage of the encoder, which is essentially the backbone in ResNeXt. Nodes at level $j > 0$ receive two inputs, one is the output of the previous nodes in the same residual connection pathway and the other is the upsampled output from the adjacent lower residual connection pathway. We do not use all the lower stage outputs, which are used in HRNet and UNet 3+. First, in remote sensing images, deeper features generally do not contain better semantic information. Second, the resolution and semantic gap is so large that direct use will introduce more noise. Finally, overly complex models are more prone to overfitting due to the lack of labeled datasets.

From another perspective, ResUNeXt of different depth is integrated in ResUNeXt++. The network can adjust the depth of the network adaptively. When the deep layer of the network does not learn useful information for prediction, the residual connection in feature fusion will discard this part of information, making the network more versatile in different datasets and different tasks.

C. Selective Nonlocal Operation

Formally, we can define a generic nonlocal operation as

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f\left(\mathbf{x}_i, \mathbf{x}_j\right) g\left(\mathbf{x}_j\right) \tag{4}$$

where i and j are the indexes of an position, it usually refers to pixel position of feature map in computer vision. $\mathbf{x}$ is the input feature map and $\mathbf{y}$ is the output feature map with the same size as $\mathbf{x}$. A pairwise function f computes a scalar representing affinity between the features at positions i and all j. The unary function g computes the embedded representation on the position j of the feature map. The output is normalized by a scalar factor $\mathcal{C}(\mathbf{x})$. It can be seen from the formula that each position on the output can be expressed as a linear combination of the input after embedded representation. Position j can be any point on the feature map, so it can ignore the distance and establish a long-range contextual dependencies. There are several versions of f and g, we use dot product similarity as function f, which can be expressed as

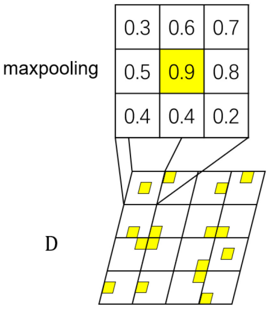

Fig. 7. Process of extracting key points. Each yellow quadrilateral on $\mathbf{D}$ represents a key point.

$$f\left(\mathbf{x}_i, \mathbf{x}_j\right) = \theta\left(\mathbf{x}_i\right)^T \theta\left(\mathbf{x}_j\right). \tag{5}$$

Here $\theta(\mathbf{x}_i)$ means the embedded representation of $\mathbf{x}_i$. For simplicity, we use 1×1 convolution over the whole feature map, so all positions share the same linear embedded function θ, so is the function g. And $\frac{1}{\mathcal{C}(\mathbf{x})} f(\mathbf{x}_i, \mathbf{x}_j)$ becomes the softmax computation along the dimension j. In order to obtain the affinity matrix, it is necessary to calculate the affinity between two-by-two at each position on the feature map. The complexity of this process is $O(H^2\, W^2\, d)$. Where H and W are the height and width of the feature map, and d is the number of channels. Although the d can be reduced by feature embedding, the main factor affecting the computational effort is the resolution of the feature map.

Different from traditional nonlocal operation, selective nonlocal operation do not use all the positions j on the feature map, but perform nonlocal operation between key points. In order to get the key points, we take the feature map $\mathbf{x}^d$ from decoder as input, and then perform one 3×3 convolution following with sigmoid function to get descriptor $\mathbf{D}$ of each position, the process is shown as

$$\mathbf{D} = \text{Sigmoid}\left(conv\left(\mathbf{x}^d\right)\right). \tag{6}$$

Descriptor $\mathbf{D}$ has the same shape as $\mathbf{x}^d$, but only one channel. Intuitively, it represents the importance of each position, or the difficulty of prediction. Because the $\mathbf{D}$ is acquired in a learnable manner, the network can adaptively determine the importance of each position based on the local feature representation. Then, perform a maxpooling on the descriptor $\mathbf{D}$ to obtain the most salient position by recording the indices in the maxpooling operation. The process of extracting key points is shown in the Fig. 7. The kernel size and stride of maxpooling can be seen as a hyperparameter to control the number of key points. For simplicity, we use adaptive-maxpooling, i.e., the kernel size and stride are the same, so the hyperparameters are compressed into one. For the hyperparameters of adaptive-maxpooling, one design strategy is to sample the same number of key points in each stage, i.e., to set the same output size in different stages, so the deeper the stage, the smaller the kernel size and stride. Another design strategy is to use the same kernel size and stride in each stage, so the deeper the stage, the fewer the key points, which forces the network to pay attention to the key points of different scales. The computational complexity of the affinity matrix in the selective nonlocal operation is only $O(k^2\, d)$ by performing nonlocal operations only between key points. Where k denotes the number of key points, which is usually much smaller than the resolution $H \cdot W$ of the feature map. Since most pixels in an image can be well classified, there is no need to reinforce semantic information at all positions and the limited computational resources should be allocated to the most important positions first.

Then perform nonlocal operation between key points i and j. The result of the operation is reincorporated into the original feature map according to the position of each key point. Instead of replacing the features at each position with the computed results, we use a residual style to fuse the original features and the computed features, because this results in better performance and more stable training. The calculation process can be expressed as

$$\mathbf{y}_i = \mathbf{x}_i + \frac{1}{\mathcal{C}(\mathbf{x})} \sum_{\forall j} f\left(\mathbf{x}_i, \mathbf{x}_j\right) g\left(\mathbf{x}_j\right). \tag{7}$$

Here, i and j are the indexes of the key points. The key points are obtained by performing maxpooling operations on $\mathbf{D}$. So far, the key points have been enriched with global semantic information. Using the residual convolution block, this global semantic information is then propagated around the key points.

Because the maxpooling evenly sample points on the feature maps, the ratio of foreground and background is rebalanced, and the noise caused by background diversity is alleviated. It also greatly reduces the amount of computation of similarity calculation between two points. We perform nonlocal operations on the decoder part right after residual connection to enhance the feature representation of the feature maps participating in the multiscale prediction, thus improving the network performance.

D. *Multiscale Prediction and Deep Supervision*

Traditional deep supervision is implemented by connecting auxiliary classifiers to the intermediate layers during training, and optimizing the main loss and auxiliary supervision loss at the same time, so the gradient can be directly backpropagated from the auxiliary loss to the intermediate layers. The total loss can be expressed as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{main}} + \alpha_t \mathcal{L}_{sup} \tag{8}$$

where α_t controls the tradeoff between the two terms. In normal practice, in order to use the second term mainly as regularization, α always decays as a function of epoch t. For example, we can use a simple linear decay function, which can be expressed as

$$\alpha_t = \alpha_{\text{init}} * (1 - t/N) \tag{9}$$

where α_{init} denotes the weight of the initial auxiliary loss, which is a hyperparameter. t indicates the current epoch, N indicates the total number of epochs. Between each epoch, α_t is updated. It can be seen that the auxiliary loss item will approach zero in the later training process.

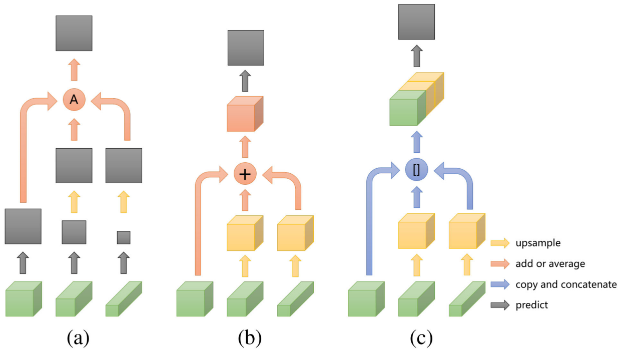

Fig. 8. Three methods of multicale prediction (a) predict and average, (b) add and predict, (c) concatenate and predict.

In UNet++, the auxiliary predictor is attached to feature maps in highest skip pathway to performs mask prediction as deep supervision. Different from UNet++, we fuse multiscale feature maps from the decoder, perform prediction, and conventional loss calculations to achieve deep supervision. Because we found that the prediction of different scales have different focuses, this multiscale prediction method is also retained in the testing phase to obtain better performance.

There are many ways to fuse feature maps from different stages, as illustrated in Fig. 8. We use bilinear interpolation to upsample all feature maps that have undergone selective nonlocal operations to the highest resolution, with a width and height of 1/4 of the original image. Then, perform the normal segmentation prediction head on the concatenated feature maps to obtain the final prediction mask, the process is shown in Fig. 8(c). Because it has the best performance in the experiment, and the amount of calculation introduced is very small compared to the overall amount of calculation.

IV. Experiment

A. *Dataset*

The dataset used in the experiment comes from [48], which uses 18-level Google Earth images with a spatial resolution of 0.522 m taken in Beijing, the capital of China. Beijing is a typical urban city, containing a variety of buildings, which is very suitable for building extraction research. The dataset contains a total of 344 labeled images, which has been divided into training set (80%) and test set (20%) by authors. We randomly select 20% from the training set as the validation set, so the ratio of training set, validation set and test set is 3:1:1. It is worth mentioning that the size of this dataset is much smaller than many other public remote sensing datasets, such as Vaihingen and Potsdam, so the model is more prone to overfit. All images have been cut into 512×512 patches, including three channels of RGB. Some typical patches in the dataset are shown in Fig. 9. As shown in the figure, the buildings are very small and dense, which poses a challenge to the extraction task.

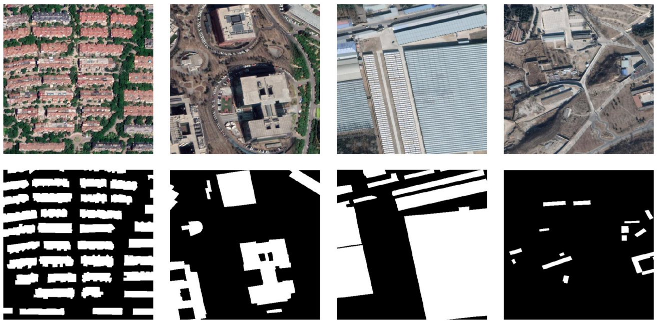

Fig. 9. Sample patches from the dataset taken in Beijing from Google Earth.

B. Implementation Details

For all models, due to the high resolution of remote sensing images and memory limitations, we use the same batch size set to 8. The optimizer is SGD, the initial learning rate is 0.1, the momentum is 0.9, and the weight decay is 1e-4. The loss function is binary cross entropy. All models are trained for 300 epochs, and the learning rate decays to 0.01 at 180 epoch. For the backbone ResNeXt, we use the ResNeXt-101 32×8d [17] pretrained on ImageNet as the encoder part, and randomly initialize the decoder and residual connection part. For data augmentation, we use random horizontal and vertical flips, and random rotations of 90°, 180°, and 270° during training, and do not use data augmentation at testing phase. Our experiment is based on the open source deep learning framework PyTorch. The experimental environment is Ubuntu20.04. The GPU is GeForce RTX 3090 with 24 G memory. The CPU is AMD Ryzen 9 5900X.

C. Evaluation

In this article, we use the IoU and F1 score to evaluate the results. In order to evaluate the effectiveness of image pixel-level prediction task, we compare the prediction results with the corresponding ground truth, and divide each pixel into true positive (TP), false positive (FP), false negative (FN), and true negative (TN). The evaluation metrics used to measure the effectiveness of our method are calculated based on these four indicators.

IoU is a commonly used evaluation metric in semantic segmentation. The numerator part calculates the number of pixels that are correctly predicted in the foreground, and the denominator part calculates the number of pixels in the union of the real foreground and the predicted foreground. The calculation process of IoU can be expressed as

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP} + \mathrm{FP}}. \tag{10}$$

F1 score is another evaluation metric in the statistical analysis of binary classification. In the segmentation task, it is equivalent to Dice similarity coefficient. Before calculating the F1 score, we need to calculate precision and recall. Precision refers to the proportion of pixels that are predicted to be true positives among all predicted positives. Recall refers to the proportion of pixels that are predicted to be true positives among all positives. F1 score is the harmonic average of precision and recall, when the weight of accuracy and recall is the same. The calculation process of F1 score can be expressed as

TABLE I
EXPERIMENTAL RESULTS OF DIFFERENT NETWORKS ON [48]

method	IoU	F1 score
UNet-8s [8]	77.67	86.11
UNet-16s [8]	76.54	85.21
SegNet [30]	74.87	84.07
UNet++ [34]	77.84	86.38
HRNet [35]	68.79	80.04
BiSeNet [49]	69.68	81.31
DeepLabv3 [9]	80.07	87.46
Res-FCN [26]	81.89	89.83
Res2-UNeXt [50]	75.91	84.83
PFNet [43]	80.60	88.43
Ours with ResNet	83.10	90.57
Ours	**84.87**	**91.65**

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

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

$$\mathrm{F1} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{13}$$

D. Comparisons With Baseline Methods

In order to verify the effectiveness of the SNLRUX++, we compared it with some baseline methods. Table I shows the experimental results of different networks. Fig. 10 provides some visualization results. The experimental results show that our proposed network has significant performance advantages and better localization on the boundary compared to other networks. Moreover, to make a fair comparison with other methods, we also used ResNet as backbone for our experiments, and its performance is the best except SNLRUX++. This further demonstrates the effectiveness of the proposed method and that the use of a strong backbone in remote sensing images does not lead to significant overfitting and can result in considerable performance gains.

Among them, UNet-16 s is the original network model proposed in the article [8]. It downsamples the image four times, so the width and height of the minimum resolution feature map is 1/16 of the original image. On the basis of UNet-16 s, UNet-8 s removes the final downsampling and corresponding upsampling, so that the width and height of the minimum resolution feature map is 1/8 of the original image. We have not adjusted the width of the UNet-8 s network, thus, it has a smaller amount of parameters and calculations than UNet-16 s. The experimental results show that UNet-8 s outperforms UNet-16 s, which indicates that for remote sensing images, not the deeper the network means the better the performance. This may be attributed to the small size of the objects in the remote sensing images, and the location information lost by downsampling is less than the semantic information it brings.

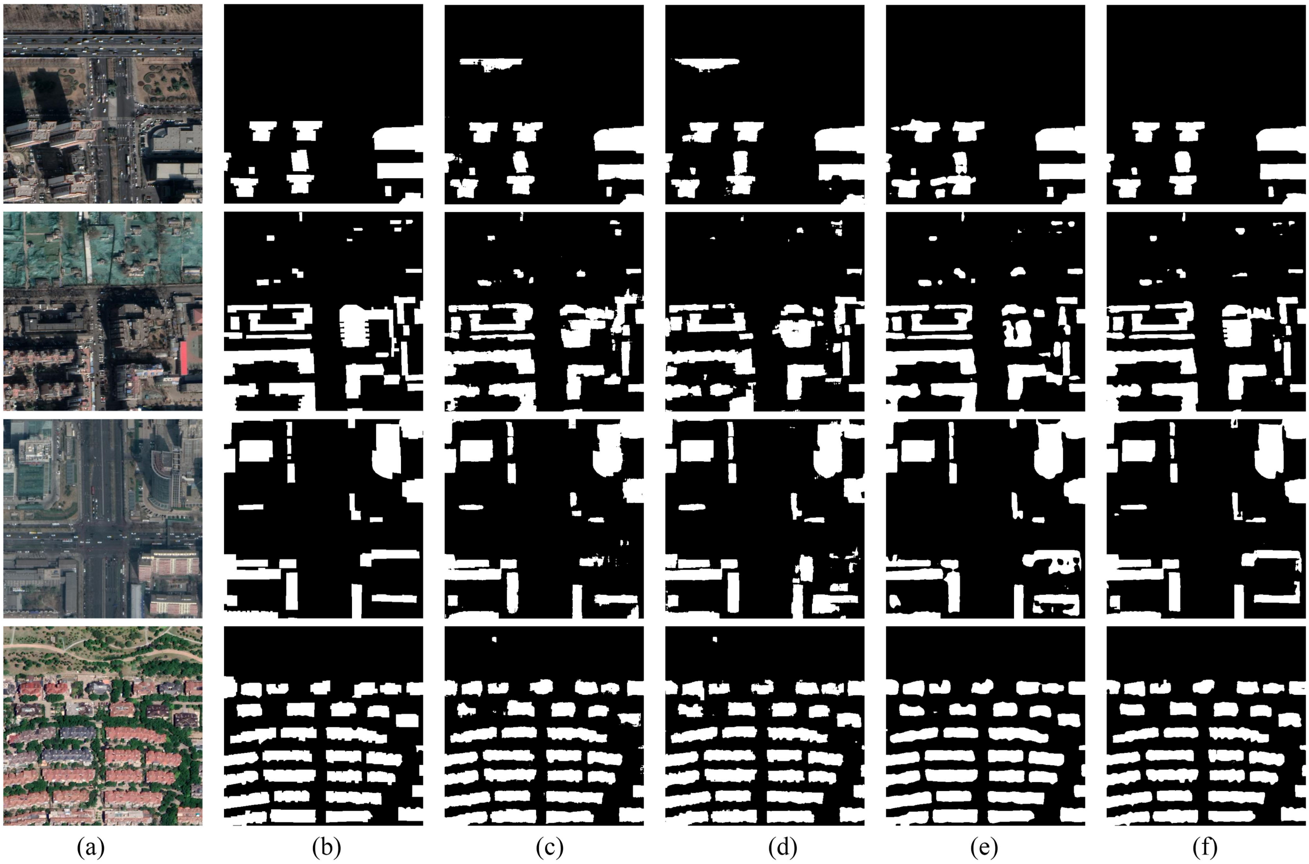

Fig. 10. Some visualization results. Compared with previous works, our method has obvious advantages in performance and boundary location. Best view it on screen and zoom in. (a) Images. (b) GT. (c) UNet-8s. (d) UNet++. (e) DeepLabv3. (f) SNLRUX++.

It is worth mentioning that UNet++ has no significant performance improvement compared to UNet. They use the same encoder and decoder. In the training process, the loss of UNet++ is much smaller than that of UNet. It can be inferred that UNet++ suffers from overfitting, which often occurs in small remote sensing datasets. Besides, the dense skip connection in UNet++ causes huge memory consumption and computational effort. Compared with this, our proposed cascaded multiscale feature fusion method is faster to train and more memory-efficient. BiSeNet and HRNet are high-performance general segmentation networks, but they perform poorly when applied to remote sensing images, even worse than the baseline UNet. This is also because the model capacity is too large for the information provided by a small dataset to support such a model.

The Res2-UNeXt [50] is a medical image segmentation model. Since medical images and remote sensing images share the problems of unbalanced foreground background and small and numerous objects, we tried to apply this model directly to remote sensing images and obtained relatively poor results. This may indicate that although there are many commonalities, there are more semantic gaps between remote sensing images and medical images.

Comparing the experimental results of DeepLabv3 and Res-FCN, they use the same backbone ResNet101, but the performance of Res-FCN has a greater advantage over DeepLabv3. This suggests that using dilated convolution directly on remote sensing images may not yield good results. Because in remote sensing images, the actual distance of the object on the image is relatively far, and the semantic relevance is not as good as that of natural images. Directly using dilated convolution to increase the receptive field will cause excessive noise to be introduced and cause model degradation. Interestingly, when using ResNet as the backbone, the most traditional and concise FCN model even outperforms many later proposed models. This may suggests that a backbone with powerful feature extraction capability is very important for complex and variable remote sensing images. In addition, some techniques in general segmentation model have inductive biases based on natural images, such as segmentation objects occupying a large portion of the entire image, which does not occur often in remote sensing images and, thus, causes performance degradation instead. This also shows that when processing remote sensing images, overly complex modules should be avoided and simple structures will instead give better results, especially when the dataset is small.

E. Ablation Experiment

In order to quantitatively analyze the contribution of different components in ResUNeXt++, we conduct ablation experiments. In general, our model contains three components, cascaded multiscale feature fusion (CMFF), selective nonlocal operation

TABLE II
ABLATION EXPERIMENT OF DIFFERENT COMPONENTS. CMFF: CASCADED MULTISCALE FEATURE FUSION, SNL: SELECTIVE NONLOCAL, MSP: MULTISCALE PREDICTION

CMFF	SNL	MSP	IoU	F1 score
			80.41	88.91
✓			82.84	90.39
	✓		81.78	89.78
		✓	80.86	89.23
✓	✓		83.95	91.01
	✓	✓	82.44	90.02
✓		✓	83.18	90.64
✓	✓	✓	**84.87**	**91.65**

(SNL), and multiscale prediction (MSP). Here, we take ResUNeXt as the baseline, replace skip connection with residual connection and use "concatenate and predict" [see Fig. 8(c)] multiscale prediction method by default, and add different components for experimentation. The ablation experiment results of different components are shown in the Table II.

From the experimental results, all three components improve network performance. Among them, CMFF has the best performance improvement and no overfitting like the dense connections in UNet++. This shows that CMFF can indeed effectively fuse features at different scales and enhance the feature representation at each stage, even on small datasets. It also indirectly demonstrates that the feature maps at different stages have different focuses. With this as a basis, it is feasible to construct high-resolution and high-semantic feature maps. SNL validates its effectiveness by achieving considerable performance gains at the cost of a small number of operations. MSP has the smallest performance improvement, this is because in the case of the same height shot, the volume of different buildings does not present a very obvious difference due to architectural standards. In natural images, the distance and proximity of things can greatly affect the size in the image. So MSP in remote sensing images may not be as significant for performance improvement as in natural images. However, the evaluation metrics cannot reflect the network training speed and stability. During the experiment, with the deep supervision of multiscale prediction, the network training is significantly faster, especially directly using addition to fuse different feature maps, that is Fig. 8(b).

In order to verify the effectiveness of the residual connection and the performance difference of different multiscale prediction methods, we designed related experiments. The experimental results are shown in the Table III. From the experimental results, the residual connection can significantly improve network performance. And the training becomes more efficient due to the nature of the structure itself. The performance of the three multiscale prediction methods is similar, and "concatenate and predict" is slightly better than the other two. This is possible because the "concatenate and predict" method gives different weights to the feature maps of different scales, so that it can adjust the weights adaptively according to the size of the object scale of the datasets, and give more weight to the feature maps that are most conducive to prediction. The remaining two can be regarded as special cases where all feature map weights are the same, and converge faster. Since the calculation amount of the three methods is very small compared to the entire network, we choose "concatenate and predict" by default.

TABLE III
EXPERIMENTAL RESULTS OF DIFFERENT CONNECTION AND MULTISCALE PREDICTION METHODS

connection	multi-scale prediction	IoU	F1 score
skip	predict and average	82.55	90.21
skip	add and predict	82.55	90.18
skip	concatenate and predict	82.77	90.31
residual	predict and average	83.79	90.93
residual	add and predict	84.30	91.23
residual	concatenate and predict	**84.87**	**91.65**

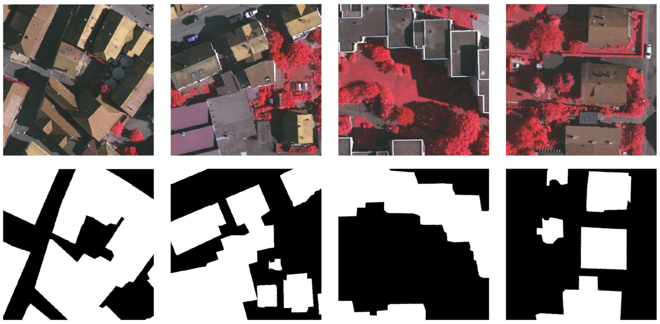

Fig. 11. Sample patches from Vaihingen dataset.

F. Experiments on Benchmark Vaihingen

In order to further verify the effectiveness of SNLRUX++, we conducted experiments on the remote sensing segmentation benchmark Vaihingen [51]. The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing, and Geoinformation (DGPF). The spatial resolution of the image is 9 cm. This dataset has 33 high-resolution remote sensing images with different sizes, the average size of which is about 2000×2500. We used the same experimental configuration as before, cutting the images into 512×512 patches. A total of 2554 patches were obtained, which is much larger than the dataset from [48], so we only trained for 150 epochs. We randomly select 25% of patches as the test set, 25% as the validation set, so the ratio of training set, validation set and test set is 2:1:1. This dataset has a total of six label categories. In order to focus on the building extraction task, we set all categories except buildings as the background. Some sample patches are shown in Fig. 11. Compared to the previous dataset, Vaihingen has a higher spatial resolution, which makes the details of the buildings more prominent. Also the dimensions in the images are larger, which makes the task of building extraction easier.

Table IV shows the experimental results of different networks. Fig. 12 provides some visualization results. As a whole, all models have better results on this dataset than before. This indicates that the number and spatial resolution of remote sensing images can significantly reduce the difficulty of building extraction. Our SNLRUX++ still maintains its best performance. This indicates

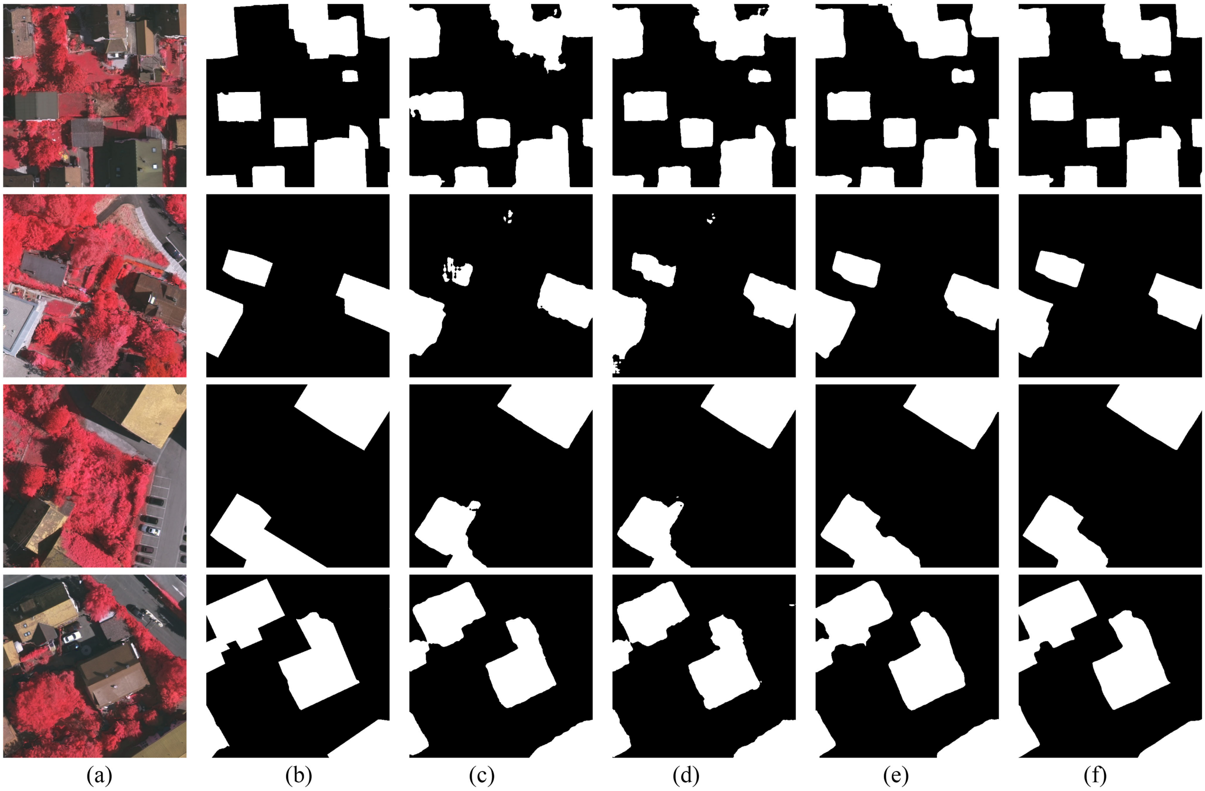

Fig. 12. Some visualization results on Vaihingen. Best view it on screen and zoom in. (a) Images. (b) GT. (c) UNet-8s. (d) UNet++. (e) DeepLabv3. (f) SNLRUX++.

TABLE IV
EXPERIMENTAL RESULTS OF DIFFERENT NETWORKS ON VAIHINGEN [51]

method	IoU	F1 score
UNet-8s [8]	91.58	94.98
UNet-16s [8]	91.78	95.27
SegNet [30]	89.99	94.19
UNet++ [34]	92.37	95.54
HRNet [35]	92.08	95.40
BiSeNet [49]	93.19	96.02
DeepLabv3 [9]	94.05	96.73
Res-FCN [26]	93.69	96.42
Res2-UNeXt [50]	90.81	94.67
PFNet [43]	92.68	95.39
Ours	**96.31**	**97.86**

that the proposed method maintains its effectiveness in both small and large datasets. UNet-16 s outperforms UNet-8 s on this dataset, which verifies our hypothesis that deeper networks are more advantageous when the objects in the images are larger in size. Models that are prone to overfitting, such as HRNet and BiSeNet, have significantly improved performance on this dataset. This illustrates that the increase in data volume can effectively mitigate model overfitting, but the cost of acquiring data in remote sensing images is often enormous, making these high-capacity models perform poorly. Due to the larger building size, the dilated convolution method DeepLabv3 performs better on this dataset compared to the previous dataset. When the spatial resolution is higher, the semantic correlation between objects becomes higher, resulting in a better performance of the expanded receptive field approach.

V. Conclusion

In this article, we proposed a new end-to-end deep learning network SNLRUX++ to solve the problems encountered when general segmentation models were transferred to high-resolution remote sensing images building extraction task. We explored the feature map fusion methods and proposed a cascaded multiscale feature fusion method to improve the performance of the network on small but numerous buildings. We proposed selective nonlocal operation to establish long-range contextual dependencies. It alleviates the introduction of additional noise and the huge amount of calculation caused by using nonlocal operation directly on the entire feature map. We used multiscale prediction as deep supervision, which makes training more stable and accelerates network convergence. And used the prediction advantages of feature maps of different scales to improve the network's performance for different scale buildings. We conducted comparative experiments of different methods and ablation experiments of the ResUNeXt++ components on the public dataset [48]. Further experiments on Vaihingen segmentation dataset also prove the generality of our method.

References

[1] J. Cardoso-Fernandes, A. C. Teodoro, and A. Lima, "Remote sensing data in lithium (Li) exploration: A new approach for the detection of Li-bearing pegmatites," *Int. J. Appl. Earth Observ. Geoinformat.*, vol. 76, pp. 10–25, 2019.

[2] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.

[3] F. Vanegas, D. Bratanov, K. Powell, J. Weiss, and F. Gonzalez, "A novel methodology for improving plant pest surveillance in vineyards and crops using UAV-based hyperspectral and spatial data," *Sensors*, vol. 18, no. 1, 2018, Art. no. 260.

[4] M. Shimoni, R. Haelterman, and C. Perneel, "Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 101–117, Jun. 2019.

[5] T. Wellmann *et al.*, "Remote sensing in urban planning: Contributions towards ecologically sound policies?," *Landscape Urban Plan.*, vol. 204, 2020, Art. no. 103921.

[6] C. Liping, S. Yujun, and S. Saeed, "Monitoring and predicting land use and land cover changes using remote sensing and GIS techniques-a case study of a hilly area, Jiangle, China," *PLoS One*, vol. 13, no. 7, 2018, Art. no. e0200493.

[7] M. A.-A. Hoque, S. Phinn, C. Roelfsema, and I. Childs, "Tropical cyclone disaster management using remote sensing and spatial analysis: A review," *Int. J. Disaster Risk Reduction*, vol. 22, pp. 345–354, 2017.

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

[9] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[10] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *ICLR*, 2016.

[11] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 60–65.

[12] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Urban land cover classification with missing data modalities using deep convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 6, pp. 1758–1768, Jun. 2018.

[13] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 1–9.

[14] T. Zuo, J. Feng, and X. Chen, "Hf-FCN: Hierarchically fused fully convolutional network for robust building extraction," in *Proc. Asian Conf. Comput. Vis.*, Springer, 2016, pp. 291–302.

[15] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet : Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.

[16] X. Yao, X. Feng, J. Han, G. Cheng, and L. Guo, "Automatic weakly supervised object detection from high spatial resolution remote sing images via dynamic curriculum learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 675–685, Jan. 2021.

[17] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.

[18] A. Rosenfeld, *Digital Picture Processing*. New York, NY, USA: Academic, 1976.

[19] Y. Zhang and L. Wu, "Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach," *Entropy*, vol. 13, no. 4, pp. 841–859, 2011.

[20] M. M. S. J. Preetha, L. P. Suresh, and M. J. Bosco, "Image segmentation using seeded region growing," in *Proc. Int. Conf. Comput., Electron. Elect. Technol.*, 2012, pp. 576–583.

[21] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using k-means clustering algorithm and subtractive clustering algorithm," *Procedia Comput. Sci.*, vol. 54, pp. 764–771, 2015.

[22] F. Yi and I. Moon, "Image segmentation: A survey of graph-cut methods," in *Proc. Int. Conf. Syst. Informat.*, 2012, pp. 1936–1941.

[23] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1356–1363.

[24] Y. Yu, J. Huang, S. Zhang, C. Restif, X. Huang, and D. Metaxas, "Group sparsity based classification for cervigram segmentation," in *Proc. IEEE Int. Symp. Biomed. Imag., Nano Macro*, 2011, pp. 1425–1429.

[25] P. Ghamisi *et al.*, "New frontiers in spectral-spatial hyperspectral Image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 10–43, Sep. 2018.

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

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[29] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[30] 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.

[31] 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.

[32] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6399–6408.

[33] 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.

[34] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet : A nested u-net architecture for medical image segmentation" in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. New, York, NY, USA: Springer, 2018, pp. 3–11.

[35] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5693–5703.

[36] H. Huang *et al.*, "Unet 3+: A full-scale connected UNET for medical image segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 1055–1059.

[37] Z. Shao, P. Tang, Z. Wang, N. Saleem, S. Yam, and C. Sommai, "BRRNet: A fully convolutional neural network for automatic building extraction from high-resolution remote sensing images," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 1050.

[38] J. Zhang, S. Lin, L. Ding, and L. Bruzzone, "Multi-scale context aggregation for semantic segmentation of remote sensing images," *Remote Sens.*, vol. 12, no. 4, 2020, Art. no. 701.

[39] G. Cheng, Y. Si, H. Hong, X. Yao, and L. Guo, "Cross-scale feature fusion for object detection in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 431–435, Mar. 2020.

[40] G. Chen *et al.*, "Fully convolutional neural network with augmented atrous spatial pyramid pool and fully connected fusion path for high resolution remote sensing image segmentation," *Appl. Sci.*, vol. 9, no. 9, 2019, Art. no. 1816.

[41] R. Dong, X. Pan, and F. Li, "DenseU-net-based semantic segmentation of small objects in urban remote sensing images," *IEEE Access*, vol. 7, pp. 65347–65356, 2019.

[42] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[43] X. Li *et al.*, "Pointflow: Flowing semantics through points for aerial image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4217–4226.

[44] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[45] G. Cheng, C. Lang, M. Wu, X. Xie, X. Yao, and J. Han, "Feature enhancement network for object detection in optical remote sensing images," *J. Remote Sens.*, vol. 2021, 2021.

[46] 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.

[47] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017.

[48] L. Xia, X. Zhang, J. Zhang, H. Yang, and T. Chen, "Building extraction from very-high-resolution remote sensing images using semi-supervised semantic edge detection," *Remote Sens.*, vol. 13, no. 11, 2021, Art. no. 2187.

[49] 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.*, 2018, pp. 325–341.

[50] S. Chan, C. Huang, C. Bai, W. Ding, and S. Chen, "Res2-UNeXt: A novel deep learning framework for few-shot cell image segmentation," *Multimedia Tools Appl.*, pp. 1–14, 2021.

[51] F. Rottensteiner *et al.*, "The ISPRs benchmark on urban object classification and 3D building reconstruction" *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 1, no. 1, pp. 293–298, 2012.

Yanjing Lei received the Ph.D degree in parallel and distributed computing from Northwestern Polytechnical University, Xi'an, China, in 2009.

Her current research interests include computer vision, crowd-sensing, and edge computing.

Jiamin Yu received the B.B.A degree in managerial theroy and servant leadership from the Zhejiang University of Finance and Economics, Hangzhou, China, in 2017. He is currently working toward the M.E. degree in computer vision and remote sensing segmentation with the Zhejiang University of Technology, Hangzhou, China, in 2019.

His current research interests include computer vision and remote sensing information extraction.

Sixian Chan received the Ph.D. degree in deep learning, image processing, tracking and segmentation from the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China, in 2018.

He is currently a Lecturer of the computer science and technology with the Zhejiang University of Technology. His research interests include image processing, machine learning, deep learning, and video tracking.

Wei Wu received the B.E. degree in land resource management from Anhui Normal University, Wuhu, China, in 2007, and the Ph.D. degree in cartography and geographic information system from the University of Chinese Academy of Sciences, Beijing, China, in 2013.

He is currently an Associate Professor with the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His research interest includes remote sensing information extraction.

Xiaoying Liu received the B.E. degree in electronic engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2013, and the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018.

She is currently an Associate Professor with the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Her research interests include green wireless communication networks and the mathematical modeling of images.