An Object-Oriented CNN Model Based on Improved Superpixel Segmentation for High-Resolution Remote Sensing Image Classification

Zhiqing Li, Erzhu Li, Alim Samat, *Member, IEEE*, Tianyu Xu, Wei Liu, and Yihu Zhu

***Abstract*—Object-oriented convolutional neural network (CNN) has been proven to be an effective classification method for very fine spatial resolution remotely sensed imagery. It can obtain higher accuracy and well edge preservation results due to the combination of advantages of image segmentation and deep network at the same time. However, the mismatch with the real boundary of the ground object is still a problem that needs to be solved further. In addition, a specific CNN model that can learn better feature representations also plays an important role in improving classification accuracy. For these purposes, we proposed an improved sample linear iterative cluster (SLIC) to obtain better segmentation edges. This algorithm overcomes the limitation of the input feature dimension in SLIC and improves the boundary performance by using more features. Besides, in order to obtain better feature representations, a new CNN model has also been developed, which can make full use of spectral information to learn first-order and second-order fusion features for classification. This method has been verified on four real remote sensing images. Compared with other methods, the proposed method achieves better performance in terms of edge and classification accuracy.**

***Index Terms*—Convolutional neural network (CNN), second-order pooling, superpixel segmentation.**

I. Introduction

LAND use and land cover (LULC) information can effectively reflect the economic level and development status, hence, it is widely used in urban planning and management. Meanwhile, it can also provide essential reference information for evaluating the transportation model and is necessary information for studying human activities and environmental change [1]. With the rapid development of remote sensing technology, the threshold for obtaining very fine spatial resolution (VFSR) images has been gradually decreased. Consequently, it has become possible to automatically obtain high-precision LULC information [2]. Nevertheless, it is still difficult to identify LULC information only based on the spectral features of VFSR images due to the complex and heterogeneous characteristics in images. For example, the same land-use type (such as residential areas) may have different physical properties and land cover materials; conversely, different land-use types (such as asphalt and parking lots) may exhibit the same or similar reflective spectrum and texture [3]. Moreover, some urban land-use classes are usually defined according to their functions and uses, so it is hard to capture their unique characteristic information from the spectrum, texture, shape, or spatial structure [4]. Therefore, it is necessary to develop efficient and accurate feature extraction and classification techniques to automatically obtain LULC information from VFSR images.

So far, many methods have been proposed to complete the LULC classification task [5], [6]. These methods can be roughly divided into four categories based on the spatial unit of representation, namely, pixels, moving windows, objects, and scenes [7]. Among them, the pixelwise approaches are the earliest LULC classification methods, which mainly rely on the spectral feature information to classify each pixel of the entire image [8], [9]. However, spectral features used in the pixelwise methods are not enough to accurately distinguish complex land use or cover categories in cities [10]. Subsequently, methods of using spectral-spatial information are proposed. This kind of methods compute a moving window over the whole image to extract more relative information, including texture features [11] and background information [12]. By classify pixels with more information, the window-based methods achieved better performance than the pixelwise methods. Nevertheless, this method still needs to define the shape and size of windows in advance but these characteristics of the object are often irregular in reality [11]. In addition, the above two methods often have redundant calculations due to their essentially analyzing the image at the pixel level. To this end, object-based image analysis methods are proposed. This kind of methods have been considered the dominant paradigm over the last decade, as it greatly reduces processing units and excellently preserves

Manuscript received March 2, 2022; revised May 5, 2022; accepted June 1, 2022. Date of publication June 10, 2022; date of current version June 21, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 41801327 and Grant 41071424, in part by the Youth Innovation Promotion Association Foundation of the Chinese Academy of Sciences under Grant 2018476, in part by the Xuzhou Science and Technology Key R&D Program (Social Development) under Grant KC20172, and in part by the Jiangsu Province Land and Resources Science and Technology Plan Project under Grant 2021046. *(Corresponding author: Erzhu Li.)*

Zhiqing Li, Erzhu Li, Tianyu Xu, and Wei Liu are with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou 221116, China (e-mail: lizhiqing97@126.com; lierzhu2008@126.com; xuty96@163.com; liuw@jsnu.edu.cn).

Alim Samat is with the State Key Laboratory of Desert and Oasis Ecology, Xinjiang Institute of Ecology and Geography, Chinese Academy of Sciences, Urumqi 830011, China (e-mail: alim_smt@ms.xjb.ac.cn).

Yihu Zhu is with the Jiangsu Geologic Surveying and Mapping Institute, Nanjing 211102, China (e-mail: zhuyihoo@jsdzch.com).

Digital Object Identifier 10.1109/JSTARS.2022.3181744

boundary of ground truth [13], [14]. Moreover, since the shapes of its segmentation results are different, it can also introduce some deeper feature information, such as shape attributes and area attributes, to enhance the description of segmented objects [15], [16]. Nevertheless, not all attributes can correctly enhance feature representations. Sometimes they only work on a specific shape, and sometimes they even have a counter-effect [17]. Apart from that, oversegmentation and undersegmentation are also common in all current segmentation algorithms and, thus, they cannot provide actual shape features [18]. To accurately describe segmented units in classification, some researchers tried to combine well-defined land information and spatial features to enhance features [19]. Although this kind of scenario-based methods offer a clear segmentation unit description, it is rarely able to obtain relevant land information in practical production. In addition, due to the different application goals and different auxiliary information involved in feature construction, it is also difficult to formulate a set of general rules to apply this method [17]. Therefore, completing a successful LULC classification based on VFSR remote sensing images remains a challenge.

Recently, deep learning (DL) has become the mainstream method of feature learning and classification. As a supervising method, it has more parameters and more complex hierarchical structure than traditional methods and, thus, it can learn deeper semantic features of data and obtain higher accuracy [20]. Among these DL methods, the convolutional neural network (CNN) is the most outstanding one in the field of image analysis. Due to its innovative uses of convolution operation, the CNN methods have achieved many success in applications, such as image classification [21], image retrieval [22], scene annotation [23], and target detection [24]. In remote sensing (RS) image analysis, CNN methods also have an outstanding performance due to its powerful semantic feature extraction and abstraction ability. For example, it is widely used in remotely sensed scene classification [25], vehicle detection [26], road network extraction [27], and semantic segmentation [28]. However, the need for a large number of samples in CNN methods is still a disadvantage. Specifically, in specific deep learning applications, the feature extraction and classification of images are completed through the learning of massive samples, but the category information in the image is usually complex and hard to collect enough labeled samples. For this purpose, some researchers try to fine-tune the fully-trained CNN from scratch to implement model transfer applications [29], such as adjusting the pretrained CNN model with a small number of samples to adapt to the task in the target domain [30]. Although these works are effective in practical applications, further improvements are still needed to solve more complex tasks and it remains a challenge to train an ideal model with few samples.

There are two kinds of CNN methods that can be used in LULC classification task. One approach is using an end-to-end CNN model to directly predict the category of each pixel in the RS image (such as UNet [31], PSPNet [32], PSANet [33], SegFormer [34], and KNet [35]), which can be called semantic segmentation method. However, most of these methods are designed for nature image and, thus, they cannot make full use of the spectral information of RS image. Moreover, complex distributed objects and various sizes also need to be processed in a special way. To this end, recently researches try to fuse additional texture features from RS image to improve the classification performance. For instance, NDVI and DSM are fused in AFNet [36], foreground information are related in FANet [37], co-occurrence relations between different ground truth objects are constructed in CGFDN [38], the information of pixel-to-pixel and pixel-to-object are fused in HCNet [39]. Moreover, reinforcement learning on small-scale objects has been also prove its effectiveness in improving the classification performance [40], some other operations, like multilayer perceptron to make full use of the neighbor pixels [41], filter operation to defect noisy pixels [42], more features used [43], and metric learning regularization [44], are also valid to enhance features. These methods have all been proved to be effective for the RS image classification. However, the samples are usually not enough to train a new model in a real classification task. Moreover, since the CNN model has no direct constraints on the boundaries, the boundaries in their classification results are often difficult to match with the ground objects.

For this purpose, the other method for LULC classification is proposed, namely, the object-oriented method. Compared to the above end-to-end semantic segmentation methods, these methods try to decouple the classification process into segmentation and classification. Specifically, the first part uses segmentation algorithm or a slide window to obtain basic analysis unit and the second part uses classification CNN model to predict the category of each unit. In earlier studies, some research try to use moving window to classify its central pixel to complete LULC classification [45]. However, the extraction and analysis of a large number of overlapping image patches will inevitably lead to the computational redundancy and the misclassification of pixels at the boundary of ground objects. In addition, since the target information usually diverse in reality, and thus, fixed window cannot clearly reflect the various characteristics. To better reflect character of pixels, researchers subsequently proposed a multiscale image patch generation method, which sets multiple windows of different sizes for the same pixel to extract its feature representations. Experiments have proved this method is indeed better than the single-scale method [46]. Nevertheless, these methods still adopt the same pixel-by-pixel classification strategy as the single-scale method and thus it is still a pixelwise method. Therefore, it cannot avoid the problems of redundant calculation and blurred edges.

For the abovementioned problems, researchers proposed a new method that converts the basic analysis unit from pixel to superpixel. It uses superpixel algorithm to complete segmentation and uses CNN to complete classification. Compared with pixelwise methods, a superpixel usually consists of several adjacent and consistent pixels, which can significantly reduce the number of the analysis units. Moreover, due to their excellent edge preservation of the ground object boundary, this method can also obtain better edge performance than pixelwise methods [47]. Furthermore, this method can also extract deeper features from its segmentation results. For instance, using a superpixel block and its surrounding blocks for feature construction [48]

or using the correlation between the superpixel block and the overall segmentation result to further construct feature [47]. However, although many superpixel segmentation algorithms (such as simple linear iterative clustering (SLIC) [49], quick shift [50], SEEDS [51], CCS [52], Felzenszwalb [53], and LSC [54]) have been proposed these years, there is still no one segmentation method that can work perfectly for all scenarios. In addition, these traditional algorithms are usually designed for the natural images and cannot be directly applied to remote sensing images with more spectral bands. Therefore, it is necessary to design a specific superpixel segmentation algorithm that suitable to process RS images for better classification by oriented-object methods.

In the classification process, there also exists insufficient utilization of spectral information. For example, the commonly used CNN models, including AlexNet [20], VGG [55], GoogleNet [56], MobileNet [57], and DenseNet [58] are usually designed for the input images with three channels. Therefore, they cannot make full use of the spectral features of RS images with more bands contained for learning. Furthermore, these models usually directly use their extracted features for classification while ignoring the difference in weights between interesting regions and backgrounds. For this purpose, the second-order pooling network model are proposed, such as bilinear CNN [59], RFS CNN [60], and GSoP-Net [61]. With using quadratic polynomial in their convolutional operations, these networks can build more robust feature representations. These CNN methods often use the second-order pooling to improve their learning ability of nonlinear features, and thus, enable them to obtain better semantic expression in the classification task. Nevertheless, these methods are seldom used in remote sensing image classification. In addition, existing methods usually pay too much attention on the construction of high-order features while ignoring the comprehensive utilization of different order features. Therefore, how to fully use spectral information and construct a first-order and second-order feature fusion model is still an open question.

In this article, we proposed an object-oriented CNN LULC classification method. Improvements from two aspects are implemented in it. The first part is the improved SLIC algorithm, which is proposed to better segment RS images. It overcomes the input dimension limitation of the traditional SLIC algorithm and makes full use of the multiband feature of RS images. At the same time, the improved SLIC also uses the quick shift segmentation results to construct texture features to better guide cluster centers movement. Moreover, a noisy pixel filter operation is also involved in the improved SLIC to avoid the error guidance of gross errors during iteration. The second part is the first-order and the second-order features fusion network, which aims to construct more discriminative features to represent RS images. Specifically, the improved model expands the dimensionality of its input data to four bands, and at the same time, uses the first-order and second-order fusion feature for the classification. Therefore, the main contributions of this article can be summarized as follows: 1) proposing an improved superpixel segmentation algorithm suitable for remote sensing images with more features; and 2) constructing a CNN model to learn the first-order and second-order fusion features for better classification.

II. Related Works

Recently, CNN has become the most popular method to complete the task of pixel-level classification for RS image. From the perspective of the workflow of different methods, these methods can be divided into two types. The first is to use an end-to-end CNN model to directly predict the category of each pixel in a RS image, such as AFNet [36], FANet [37], CGFDN [38], and HC-Net [39]. These CNN models are usually trained with samples that are cut out from the RS image and use their well-designed decoder and encoder to predict a whole image. However, since these CNN models are all highly coupled, it is hard to improve them by modifying parts of them. Besides, due to no explicit limitation of boundary performance in the CNN model, these methods are usually hard to preserve the boundary of the ground objects and may contains a lot of holes that composed of misclassified pixels in their classification. For these purposes, the other methods that named object-oriented classification method have been proposed. In contrast to these end-to-end semantic segmentation CNN methods, object-oriented methods decouple the classification into the segmentation and the classification [62]–[64]. With using this strategy, these methods can easily use state-of-the-art segmentation to limit the object boundary and can directly use state-of-the-art classification CNN model to improve the classification accuracy. However, there still existed some problems that make these methods hard to be applied for RS images.

The first problem is that there is not an appropriate segmentation algorithm can well segment RS images with more features. Since RS images usually contain the near-infrared band and most of segmentation algorithm are designed for nature images, including SLIC [49], quick shift [50], SEEDS [51], CCS [52], Felzenszwalb [53], and LSC [54], thus, it is difficult to obtain well segmented units to represent objects. Apart from this reason, the problem of oversegmentation and undersegmentation in the current algorithms also defects the classification results [18]. Therefore, it is a challenge to design a specifically segmentation algorithm for RS image.

As with the first problem, the classification CNN model also suffers from the lack of utilization of images. Recently, some researches try to construct higher order statistical features to learn discriminative features for classification. For instance, second-order features are extracted to train the CNN model in bilinear CNN [59], RFS CNN [60], and GSoP-Net [61]. However, these methods usually ignore the using of the basic first-order features in their classification. Besides, due to the spectral particularity of RS images, the traditional network model can also be improved from the perspective of band input.

III. Methods

The object-oriented CNN method consisted of two steps, the first step is to generate objects through image segmentation, and the second step is to complete the classification using a CNN

model. In this section, we will introduce the improved SLIC segmentation method and the constructed CNN model.

A. Improved SLIC

1). Simple Linear Iterative Clustering: As the basis of object-oriented CNN, the segmentation algorithm determines the edge quality of the result. SLIC [49] is one of the most famous superpixel segmentation methods. Compared with other segmentation algorithms, its generated segmentation units are more uniform and compact, while it also has faster and higher memory efficiency. Therefore, this method is more suitable to perform image preprocessing on large-scale images.

Specifically, SLIC is a grid search algorithm that uses K-means to cluster and generate superpixels. By default, the input image is first converted from RGB color space to CIELAB color space, which aims to make the numerical differences more prominent and easier for the computer to distinguish. Then, each pixel of the image is first expressed as $(l_i,\ a_i,\ b_i,\ x_i,\ y_i)$, where $(l_i,\ a_i,\ b_i)$ is the color representation in CIELAB color space, and $(x_i,\ y_i)$ is the coordinates of the corresponding pixel. At the same time, the image is covered with grid mesh, and each center in the grid will be considered as the initial clustering center. In one iteration of clustering, each clustering center and pixels in the rectangle around it are measured distance with the Euclidean distance, and finally each pixel will be associated with its nearest cluster center. Like K-means, the iteration will continue until all the clustering centers do not move or the maximum number of iterations is reached. Note, to avoid the problem of inconsistent dimension between color distance and spatial distance, the two distances are combined by the following formula:

$$\mathrm{d_{xy}} = \sqrt{(\mathrm{x_k} - \mathrm{x_i})^2 + (\mathrm{y_k} - \mathrm{y_i})^2} \tag{1}$$

$$d_{\mathrm{Lab}} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \tag{2}$$

$$\mathrm{D} = \sqrt{\left(\frac{d_{xy}}{N_s}\right)^2 + \left(\frac{d_{\mathrm{Lab}}}{m}\right)^2} \tag{3}$$

where m is a constant parameter to normalize the dimension of the color distance d_{xy}; $N_s = \sqrt{N\ /\ k}$ denotes the maximum distance between seed points and is used to normalize the dimension of the spatial distance, N is the number of total pixels, and k is the expected number of superpixels. Therefore, there are two parameters that need to be set for SLIC, k, and m.

It is worth noting that SLIC usually requires postprocessing operations to correct its clustering results. Specifically, clustering segmented units that are too small into their similar surrounding superpixels and using a specific threshold to segment the superpixels that are too large.

2) Improving SLIC Method With More Features: Generally, SLIC only supports three-band input due to its color space conversion mechanism. However, RS images usually contain four or more bands, and thus, only using three of them may lead to insufficient utilization of spectral information. In addition, due to the limitation of the rectangle size during clustering, SLIC also have the problem of blurred boundary.

In this article, more spectral features are used to measure the similarity between pixels to improve the segmentation performance. Besides, a texture feature is also constructed to guide the movements of cluster centers for better segmentation. We compared several current mainstream segmentation methods, including SEEDS [51], CCS [52], Felzenszwalb [53], LSC [54], and quick shift [50], and found that quick shift obtained the best boundary performance with high efficiency. Unfortunately, the segmentation results of this algorithm have obvious differences in shape and size, which is not conducive to the training of deep learning. In order to take advantage of the excellent performance of the quick shift method on the segmentation boundary, we tried to construct specific texture features based on the quick shift segmentation result and introduce them into the SLIC method.

Specifically, the texture feature construction includes two steps: 1) convert the input image to a grayscale image and normalize it, and then send it to quick shift for segmentation; 2) construct texture feature by counting average value of each superpixel. In order to control the influence of texture features in clustering, we defined a constant parameter as ω to scale them. As a result, the features of a pixel can be represented as $u_i = (r_i,\ g_i,\ b_i,\ nir_i,\ t'_i)$, where $(nir,\ r,\ g,\ b)$ is the value of four spectral bands (near-infrared, red, green, and blue) in the RS image and $t' = \omega \times t$ is the scaled value of the texture feature, ω is the texture influence factor, and t is the value of the original texture feature. Therefore, the distance metric changes and is expressed as

$$d_{rgbnirt} = \|u_k - u_i\|_2 \tag{4}$$

$$\mathrm{D} = \sqrt{\left(\frac{d_{xy}}{N_s}\right)^2 + \left(\frac{d_{rgbnirt}}{m}\right)^2} \tag{5}$$

where u_i and u_k denote the feature vector of a pixel and the kth cluster center, respectively.

3). Gross Error Filtering: For SLIC method, cluster centers need to be updated continuously during the iteration process. Therefore, if there are some pixels with a large distance from the cluster center, the cluster center may move in a wrong direction. To overcome this drawback, a filtering method based on the standard deviation is proposed to filter these noisy pixels. Specifically, assuming an image I has been segmented into k superpixels through one iteration and the segmented result can be expressed as $\mathrm{I} = \{\mathrm{S_1}, \mathrm{S_2}, \cdots, \mathrm{S_k}\}$. If the ith superpixel $S_i = \{p_{i1},\ p_{i2},\ \ldots,\ p_{in}\}$ contains n pixels and the coordinates of its cluster centers are denoted as $\{x_i,\ y_i\}$, and thus the distance between a randomly selected pixel and its cluster center can be calculated as d_i by (5) and the distance set from all pixels to the cluster center can be represented as $\{d_{i1},\ d_{i2},\ \cdots, d_{in}\}$. Therefore, the threshold σ_i of the noisy pixel can be calculated by

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(d_j - \bar{d}\right)^2} \tag{6}$$

where $\bar{d}$ means the mean of distance set $\{d_{i1},\ d_{i2},\ \ldots, d_{in}\}$.

Then, the filtered pixels set Ω_i can be defined by

$$\Omega_i = (\|p_{ij} - p_k\| < \lambda \cdot \sigma_i) \cap S_i \tag{7}$$

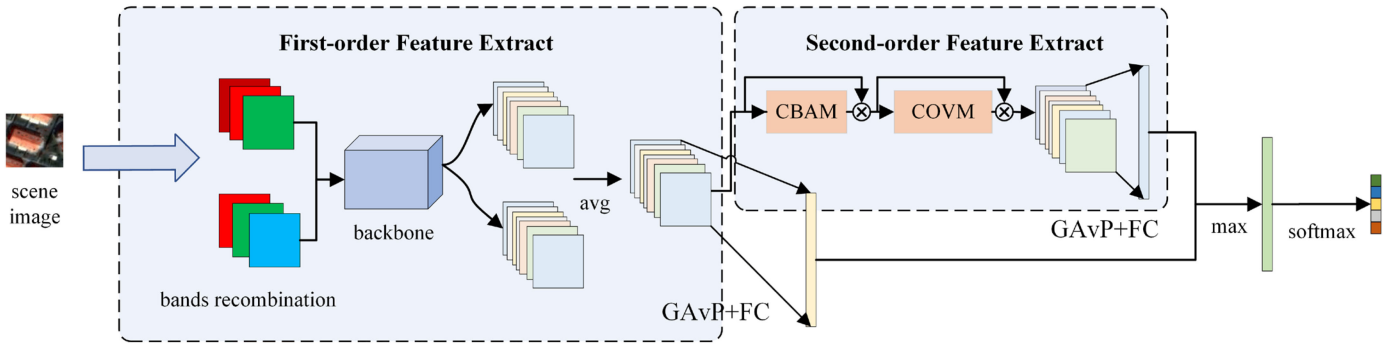


Fig. 1. Overview of the constructed CNN model.

where λ is a constant parameter that is used to control the severity of filtering and p_{ij} is a randomly chosen pixel in the ith superpixel S_i. Finally, the true cluster center φ_j after filtering can be calculated by

$$\varphi_i = \frac{1}{N'} \sum_{i \in \Omega_j}^{i} [d_{xy},\ d_{rgbnirq}]^T \tag{8}$$

where N' denotes the number of non-noise pixels.

B. CNN Model

1) Overview of the CNN Model: Fig. 1 shows the architecture of the reconstructed CNN model, which consists of two main parts: the first part is the basic first-order features extractor; and the second part is the second-order features extraction and the fusion of these two kinds of features. In the first-order features extraction, a dual-stream structure is constructed to obtain convolutional features of a four-bands RS image. In the second part, the attention mechanism and the covariance pooling module are combined to extract the second-order features. Finally, the first-order and second-order features are fused for the classification.

2). Bands Recombination: To make full use of the spatial information of a RS image, the bands recombination is adopted in the head of the CNN model. Let an input RS image contains four bands $(nir,\ r,\ g,\ b)$. According to the amount of information contained in each band, the four-bands image is split into two three-bands images with bands $(nir,\ r,\ g)$ and $(r,\ g,\ b)$. Then, the two images are input into the backbone to extract features, respectively. Finally, the mean feature of the two features is computed through global average pooling and full connect layer (FC) to obtain the first-order features.

3). Attention Mechanisms: For the input image of the CNN model, most of the regions are background information that does not need to be considered. Nevertheless, the traditional CNN model still usually abstracts the whole image while ignoring the hierarchical relationship between different regions. In recent researches, the convolutional block attention module [65] (CBAM) has been considered an effective way to strengthen the semantic representation of interesting regions. This module is composed of two parts: 1) the channel attention module (CAM) and 2) the spatial attention module (SAM).

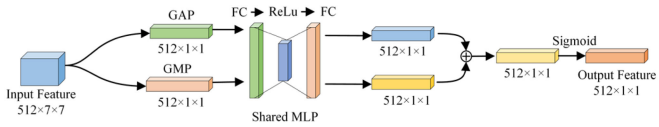


Fig. 2. Structure of the channel attention module.

The first module in CBAM is the CAM, its structure is shown in Fig. 2. Let the feature extracted by the backbone be defined as $X \in R^{C \times H \times W}$, where C is the number of channels in the convolutional features, H and W denote the height and width of the features, respectively. In the process of this module, X will first pass through two pooling layers, average-pooling and max-pooling layers. The first layer is the average-pooling layer, which is commonly used for spatial information to help adjust the weights and biases of interesting regions [66], [67]. Then, there is the max-pooling layer, which collects clues from distinctive regions to infer more effective channelwise attention [68]. After these two pooling operations, two diverse features $\mathbf{F}_{avg}$ and $\mathbf{F}_{\max}$ are obtained, which can be expressed as

$$\mathbf{F}_{(\text{avg},\ c)} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathrm{X}_c(i,\ j) \tag{9}$$

$$\mathbf{F}_{(\max,\ \mathrm{c})} = \max(\mathbf{X}_{\mathrm{c}}(i,\ j)) \tag{10}$$

where $\mathbf{F}_{(\text{avg},\ \mathrm{c})}$ represents the result of global average pooling for cth channel, $\mathbf{F}_{(\max,\ \mathrm{c})}$ denotes the result of global average max-pooling cth channel, and $\mathrm{X}_{\mathrm{c}}(i,\ j)$ means the feature map of cth channel of $\mathbf{X}$ at spatial position $(i,\ j)$.

After that, the shared multilayer perceptron (MLP) is used to enrich the interaction between the spectral bands and promote generalization. Finally, the features are elementwise summated and processed by the sigmoid function. Therefore, the process in CAM can be expressed as

$$\mathbf{F}_{\text{out}} = \sigma(\mathbf{W}'(\mathrm{ReLU}(\mathbf{W}(\mathbf{F}_{\text{avg}})))(\mathbf{W}'\mathrm{ReLU}(\mathbf{W}(\mathbf{F}_{\max}) \tag{11}$$

where $\sigma(\cdot)$ means the sigmoid function, $W \in R^{C/r \times C}$ denotes the weight extracted by the first FC layer in MLP, and

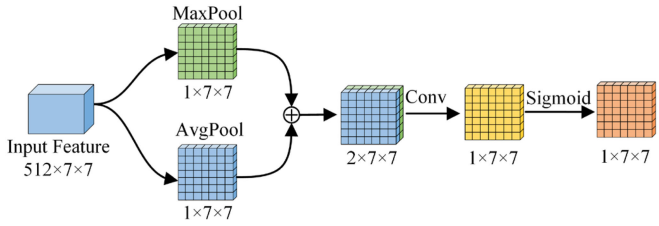


Fig. 3. Structure of the spatial attention module.

$W' \in R^{C \times C/r}$ means the weight extracted by the last FC layer in MLP.

The second module in CBAM is the SAM, which is displayed in Fig. 3. Similar to the CAM, SAM is also composed of two pooling layers. However, it did not have the shared MLP and just collected clues of position. The two features extracted by two pooling layers in this module, $\mathbf{F}_{\text{avg}}$ and $\mathbf{F}_{\max}$, can be represented as

$$\mathbf{F}_{\text{avg}}(i,\ j) \ = \ \frac{1}{C}\sum_{k\ =\ 1}^{C} \mathrm{X}_k(i,\ j) \tag{12}$$

$$\mathbf{F}_{\max}(i,\ j) \ = \ \max(\mathrm{X}(i,\ j)) \tag{13}$$

where $\mathbf{F}_{\text{avg}}(i,\ j)$ means the result of average pooling at position $(i,\ j)$, $\mathbf{F}_{\max}(i,\ j)$ denotes the result of max pooling at position $(i,\ j)$, $\mathbf{X}_{\mathrm{k}}(i,\ j)$ represents value in the kth channel feature map at position $(i,\ j)$, $\mathbf{X}\ (i,\ j)$ means values of all channels at position $(i,\ j)$, C is the total number of channels of the input.

Finally, the abovementioned two features are concatenated by channel superposition to analyze the spatial weight comprehensively and the process in the SAM can be expressed as

$$\mathbf{F}_{\text{out}} = \ \sigma(\mathbf{W}(\{\mathbf{F}_{\text{avg}},\ \mathbf{F}_{\text{avg}}\})) \tag{14}$$

where $\sigma(\cdot)$ means the sigmoid function, $W \in R^{1\times H\times W}$ denotes the weight of feature maps after convolution operation, and $\{\cdot\}$ represents the concatenation to channels.

4) Covariance Matrix Analysis Module: Although CBAM has strengthened the weight of interesting regions, its obtained features are still the first-order statistics of convolutional features. Recently researches proved that the second-order feature could effectively improve the abstract ability of the model by using the second-order statistics of convolutional features [59]. Therefore, we developed a covariance matrix analysis module (COVM) to obtain the second-order statistics of convolutional features. As shown in Fig. 4, this module is composed of two steps, covariance pooling and meta-layer.

As the first step in this module, the covariance pooling is used here to calculate the covariance matrix. Specifically, given an enhanced feature $X \in R^{c\times d\times w}$ generated by CBAM, it is reshaped into a two-dimensional matrix $X \in R^{c\times d}$, where $d = h \times w$. Then, the covariance matrix of X can be calculated by the following formula:

$$\mathbf{Cov} = \mathbf{X}\hat{\mathbf{I}}\mathbf{X}^{\mathbf{T}} \tag{15}$$

$$\hat{\mathbf{I}} = \frac{1}{d}\left(\mathbf{I} - \frac{1}{d}\mathbf{i}\mathbf{i}^{T}\right) \tag{16}$$

where $\mathbf{I}$ is the identity matrix with the size of $d \times d$, $\mathbf{i}$ is a row vector with a dimension of d and all elements are 1.

Then, a meta-layer is used to compute the approximate square root of the covariance matrix. It is composed of three parts. The first part is the prenormalization, which is to ensure the convergence of subsequent iteration. After that, there is the most important calculation in this module, Newton–Schulz iteration, which can quickly calculate the square root of the covariance matrix. Due to the limitation of the NVIDIA CUDA platform, fast implementation of Eigen decomposition or singular value decomposition is often slower than their CPU counterparts [69]. For this purpose, the optimized Newton–Shultz iterative formula is used here to faster the computation of the matrix square root, which can finish computation with no more than five iterations [70]. Specifically, given the initial iteration value $\mathbf{Y}_0 = \mathbf{A}$, $\mathbf{Z}_0 = \mathbf{I}$, where $\mathbf{A}$ is the covariance prenormalized by its trace. The square root $\mathbf{Y}$ of $\mathbf{A}$ can be computed as the following iterative forms:

$$\mathbf{Y}_k = \frac{1}{2}\mathbf{Y}_{k-1}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1}) \tag{17}$$

$$\mathbf{Z}_k \ = \ \frac{1}{2}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1})\mathbf{Z}_{k-1}. \tag{18}$$

Then, as the last component of the meta-layer, the postcompensation is used to offset the adverse effects of prenormalization. Like the attention mechanism, its result is also multiplied by the input feature through shortcut connection.

Finally, the resulting covariance matrix can be expressed as $Cov' \in R^{\mathrm{c}\times\mathrm{c}}$, each row represents the statistical correlation between one channel and all channels. Then, a convolutional kernel with shape $\mathrm{c} \times 1$ is used to learn the weight of each channel. Therefore, the process of COVM can be expressed as

$$\mathbf{F}_{\text{out}} = \ \sigma(\mathrm{ReLU}(\mathrm{Conv}(\mathrm{BN}(\sqrt{\mathbf{F}_{\text{cov}}(N)})))) \tag{19}$$

where $\sigma(\cdot)$ means the sigmoid function, BN denotes the batch normalization, and $\mathbf{F}_{\text{cov}}(N)$ is the final generated feature maps after N times iterations. In this article, the number of the Newton–Shultz iterations is set to 5.

5) Multiorder Feature Fusion and Loss Function: Although the second-order features enhanced the ability of semantic expression, it is still a kind of regional enhancement information, which may cause some unnecessary regions to be enhanced. For this reason, the first-order and the second-order features are comprehensively considered in the CNN model. Specifically, the feature fusion mechanism is adopted to construct the final representation, which chooses the maximum value of the two-order features as the final judgment feature.

In addition, since multiorder statistical information was considered in the classification, the loss function of the model also needs to be redesigned to help the backpropagation process. For this purpose, the cross-entropy function was adopted to calculate the loss of second-order and first-order features. As a result, the loss function of the CNN model can be expressed as

$$\mathrm{loss}\,(t,\ g) \ = \ -\log\left(\frac{\exp\left(\max(t_1, t_2)\,[g]\right)}{\sum_{\mathrm{j}}\exp\left(\max(t_1, t_2)\,[g]\right)}\right) \tag{20}$$

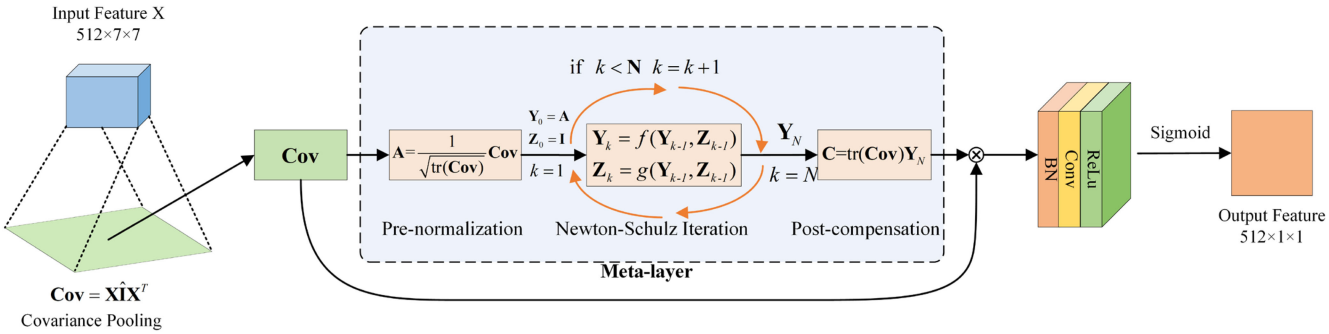


Fig. 4. Structure of the covariance pooling module.

where t_1 and t_2 are the first-order and second-order feature representations, g is the true class of the input image, and j is the sample index in each minibatch.

IV. Experiments and Results

A. Datasets

In this article, four VFSR remotely sensed images with different scales were chosen as the experimental data to test our proposed method.

The first two data, zh1 and zh9, are two small-scale remotely sensed images derived from the Zurich dataset [71], collected by QuickBird satellite in August 2002, available at https://sites.google.com/site/michelevolpiresearch/. Both images have the same ordered four spectral bands (red, green, blue, and near-infrared) with a spatial resolution of approximately 0.6 m. The size of zh1 is 1364×1295 pixels, and the size of zh9 is 1447×1342 pixels. There are a total of eight categories and each image contains seven different categories. The color images (RGB) and their corresponding ground truth images are shown in Fig. 5.

The other two experimental images, GF4454 and GF8839, are two large-scale remotely sensed images from the GID dataset collected by the Chinese GF-2 satellite, which can be obtained at https://x-ytong.github.io/project/GID.html. Both of them have four spectral bands, with the same spatial resolution of 0.6 m and the same size of 7200×6800 pixels. GF4454 includes 9 different classes, and GF8839 contains 11 different classes. Their color images (RGB) and corresponding ground truth images are shown in Fig. 6.

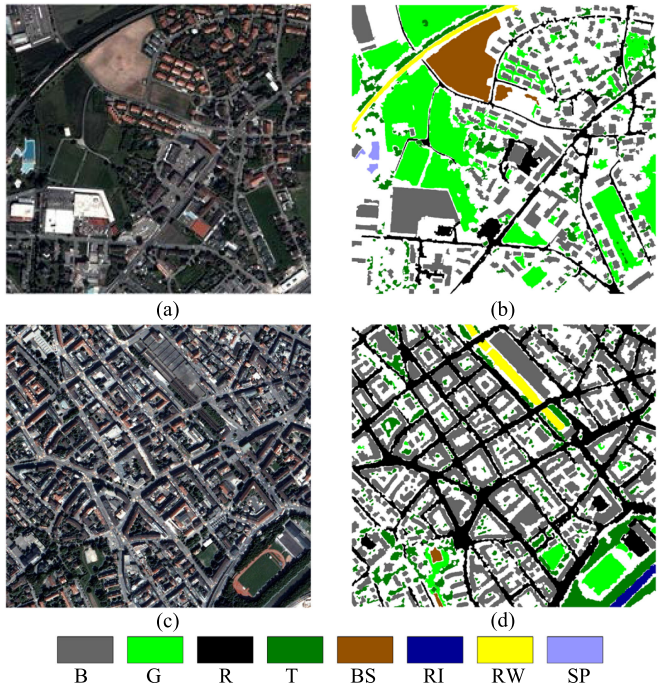


Fig. 5. Experimental images from the Zurich dataset and their corresponding ground truth images. B: Building, G: Grass, R: Roads, T: Trees, BS: Bare Soil, RI: River, RW: Railways, SP: Swimming Pools.

B. Sampling Rules

For the methods that using segmented algorithm, the same sample rules are employed here to obtain samples as the input of CNN model. In this rule, the locations of seeds of each segmented unit are first calculated by K-means. After that, sample images are cut out from the four-bands RS images with these seeds as the centers. What is worth noting is that different number of seeds are defined to correspond to oversegmented and undersegmented algorithm. For oversegmented algorithm, such as SLIC and ISLIC, one seed is enough to cut out an image to represent the segmented unit. In contrast, undersegmented algorithm, such as the method of ecognition, needs more seeds to cut out its representative images. For this reason, the number of seeds in the method of ecognition is calculated with the shape and area of each segmented unit and finally determine its category by majority voting [62].

C. Parameter Settings

During the experiment, some model parameters need to be set. For the segmentation method, four segmentation methods are involved, including SLIC [49], the improved SLIC (ISLIC), quick shift [50], and ecognition [62]. For SLIC, the maximum number of iterations was set to 10, the compactness m was set to 10 and the expected number of the superpixels k was set to $(h \times w)/100$. The improved SLIC uses the same values as SLIC for the above three parameters. Besides, the constant parameters

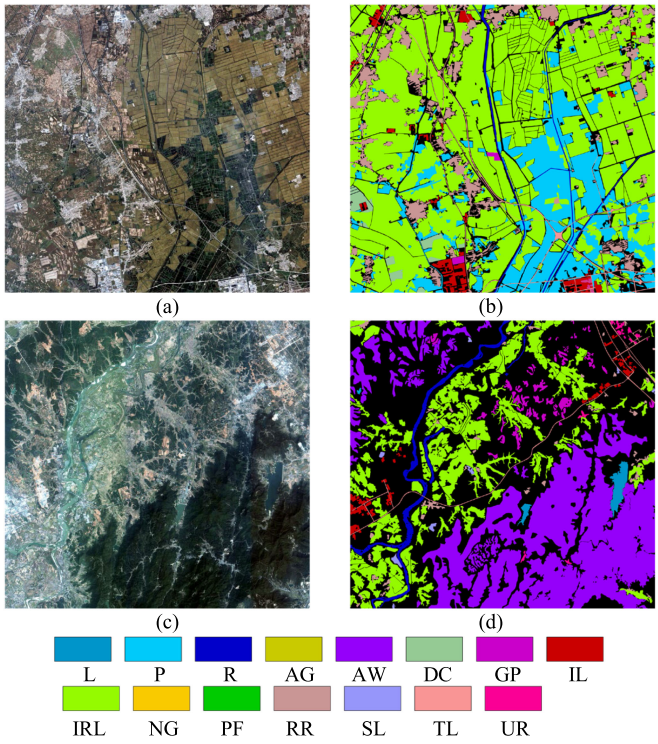


Fig. 6. Experimental images from the GID dataset and their corresponding ground truth images. L: Lake, P: Pond, R: River, AG: Artificial Grassland, AW: Arbor Woodland, Dc: Dry Cropland, GP: Garden Plot, IL: Industrial Land, IRL: Irrigated Land, NG: Natural Grassland, PF: Paddy Field, RR: Rural Residential, SL: Shrub Land, TL: Traffic Land, UR: Urban Residential.

ω and λ in the proposed method were tuned by the metrics of boundary performance to ensure that they have the best segmentation performance. In quick shift, the maximum clustering distance was set to 6, and the distance balance parameter ratio was set as 0.5. In ecognition, the best setting of segmentation parameters were calculated by the external plugin, ESP2 tool. In this plugin, the parameter of shape was set to 0.1, compactness was set to 0.5, and the number of iterations was set to 10. Through cross-validation and were finally set to 8000 and 5, respectively. The parameters of other segmentation methods were obtained through debugging to ensure that they have the best segmentation performance. In quick shift, the maximum clustering distance was set to 6, and the distance balance parameter ratio was set as 0.5. In ecognition, the segmentation parameters were calculated by its external plugin, ESP2 tool, where the parameter of shape was set to 0.1, compactness was set to 0.5, and the number of iterations was set to 10.

In the classification stage, four different methods were applied to classify images, including patchwise CNN [72], random forest [73], ResNet-50 [74], and the improved object-oriented CNN (IOCNN). The patchwise CNN is a common LULC classification method that uses densely overlapping image patches to train and predict all pixels in the image. Its most important influencing factors are the input image patch size and the CNN depth. As suggested by Chen [72], the number of this CNN layers was chosen as six to balance the complexity and robustness of the network, and its input size is 750×750 pixels. It is trained through cross-validation, the number of epochs was set to 100, the learning rate was set to 0.001, the drop rate was set to 0.01, and the SGD with a momentum of 0.9 was used to optimize. For the random forest method, the decision trees were generated by CART and the number of learning cycles was set to 100. For our proposed CNN model, ResNet-50 was chosen as the backbone to extract the first-order information. During the training stage, the learning rate was set to 0.005 to fine-tune the model, using the SGD with a momentum of 0.9 and a weight decay of 0.0001 to optimize. All procedures for the proposed methods and comparison methods are implemented based on python and PyTorch (1.9.0), and all the experiments are performed on a 64 bits machine with 64 GB RAM and an NVIDIA GeForce GTX 1080ti GPU with 11 GB memory size.

The same sampling rules are used to train and test all CNN models in the experiments. Each sample is cut out at the center of the segmentation units, with a fixed size of 150×150 pixels. In total, 60% of the labeled samples are selected for training the models and the remaining samples are used as test samples. The specific numbers of train and test samples of all experimental data are shown in Tables I and II.

TABLE I
TRAIN AND TEST DATA FOR ZH1 AND ZH9

Class	zh1		zh9	
	train	test	train	test
Roads	742	495	2524	1683
Building	1534	1024	3383	2256
Trees	456	304	896	598
Grass	1852	1235	256	172
Bare Soil	432	288	110	15
River	-	-	118	40
Railways	105	71	132	88
Swimming Pools	126	21	-	-

TABLE II
TRAIN AND TEST DATA FOR GF4454 AND GF8839

Class	GF4454		GF8839	
	train	test	train	test
River	3034	2022	5000	4222
Lake	-	-	2007	1338
Pond	8000	8000	50	32
Dry Cropland	2715	1810	-	-
Arbor Woodland	-	-	5000	5000
Shrub Land	-	-	621	414
Paddy Field	8000	8000	5000	5000
Industrial Land	4603	3068	2374	1582
Garden Plot	514	342	3874	2582
Rural Residential	8000	8000	2430	1619
Urban Residential	197	131	749	498
Traffic Land	4128	4128	1391	927

D. Results Analysis

1) Comparison of Segmentation Methods: To demonstrate the effectiveness of the improved SLIC, several ablation experiments are set up here to test the influence of the three independent improvements in our segmentation method, including the using of four bands, texture, and filter operation. In this section, the zh1 is chosen as the experimental image to be segmented and the boundary recall [75] is chosen as the metric to evaluate the

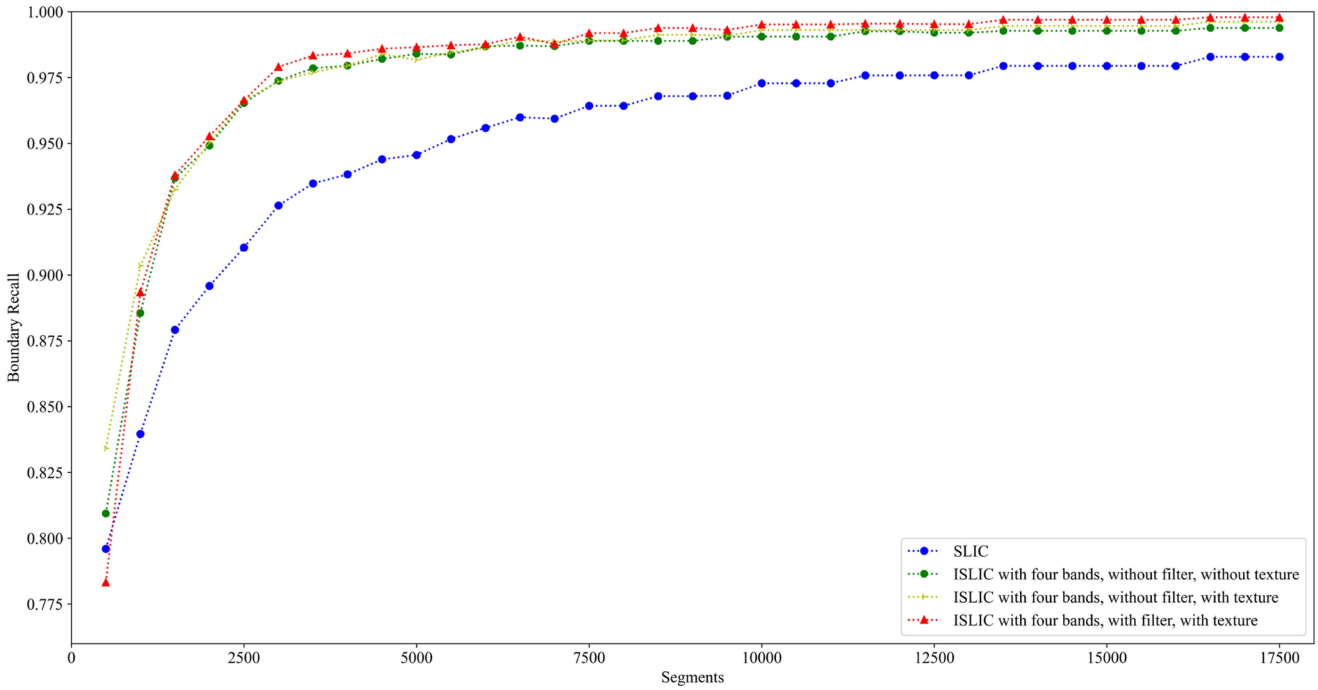


Fig. 7. Boundary performance with different methods on zh1. For SLIC, only three of the four bands RGB are used. For ISLIC, both bands of RGB and the infrared band are all used.

boundary performance of the segmented results. The formula of the boundary recall used here can be expressed as

$$\text{Recall} = \frac{\sum_{ij} p_{ij}^2}{\sum_k s_k^2} \tag{21}$$

where p_{ij}^2 is the probability of a randomly chosen pixel belongs to the boundary of a ground truth object and also belongs to the boundary of a segmented unit, s_k^2 is the probability that a randomly chosen pixel belongs to the boundary of a segmented unit.

As shown in Fig. 7, a desirable improvement of the boundary recall is obtained when ISLIC uses more spectral features. Compared with the original SLIC that only uses three of four bands, the boundary recall increased about 3% with using four-bands and the highest improvement can even reach about 6%. Besides, it can be found that the smaller the number of expected segmentation units, the better boundary improvement the proposed method can bring.

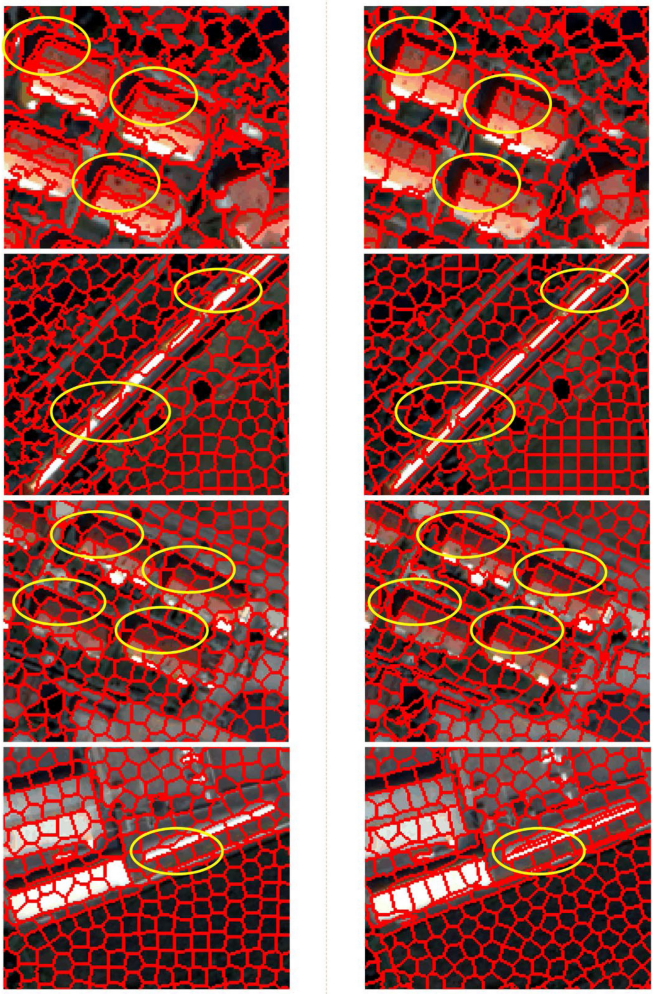
Then, with the fixed improvement of using four-bands features, we tested the performance of adding texture in ISLIC. As the yellow line shown in Fig. 7, it can be seen that boundary recall increased a lot when the number of its expected segmentation units is small. However, as the number of the expected segmentation units increases, the improvement is not obvious. For this, we think it is caused by the shrink of the slide window in ISLIC. Specifically, SLIC uses a fixed size of slide window to cluster pixels and the width is calculated by the size of the input image and the number of expected segmentation unit. Therefore, when then number of expected segmentation units increases, the width of the slide window become smaller than before and, thus, the segmentation process is hard to be influenced by the well-defined texture features.

Finally, filter operation has been evaluated here. Compared ISLIC uses three improvements with ISLIC uses four-bands features and the texture, it can be seen that the filter operation can bring a slight and stable improvement to the boundary performance. Although, there existed the problem of reduced boundary recall when the number of expected segmentation units is small, but this limitation quickly disappeared as the number of expected segmentation units increased.

Besides, in order to better display the advantage of the proposed method, the segmentation results of ISLIC and SLIC are labeled with red on the original image. As shown in Fig. 8, it is evident that the segmentation units generated by ISLIC are more uniform and compact than that of SLIC. Furthermore, ISLIC also has a better fitting effect on the edge of the objects, such as buildings and railways.

In conclusion, it can be seen that the proposed three improvements in ISLIC can all effectively improve the segmentation performance. Therefore, we adopt all of these improvements in ISLIC in the remaining experiments.

2) Comparison of Methods in Classification: This section details the classification performance of different classification methods. The proposed CNN model was compared with ResNet-50, as well as the benchmark patchwise CNN and the random forest. Two metrics of overall accuracy (OA) and Kappa coefficient (κ) are used to measure boundary accuracy of the classification results at the same time, three indicators of boundary recall, boundary precision, and F1 value [75] are selected to evaluate boundary performance. For comparison, all experimental data



(a) (b)

Fig. 8. Boundary performance of SLIC and ISLIC.

have been tested with different segmentation methods and different classification models. The metric of boundary recall is defined as (21) and the boundary precision can be expressed as

$$\text{Precision} = \frac{\sum_{ij} p_{ij}^2}{\sum_k t_k^2} \tag{22}$$

where p_{ij}^2 is the probability that a randomly chosen pixel belongs to the boundary of a ground truth object and also belongs to the boundary of a segmented unit and t_k^2 is the probability that a randomly chosen pixel belongs to the boundary of a ground truth object. The metric of F1-score is used to weight recall and precision, which can be expressed as

$$\mathbf{F} = \frac{\sum_{ij} p_{ij}^2}{\alpha \sum_k s_k^2 + (1-\alpha) \sum_k t_k^2}. \tag{23}$$

We set $\alpha = 0.5$ to weight recall and precision equally.

As the classification results shown in Tables III and IV, our proposed method achieves the best performance in classification, in contrast, the patchwise CNN got the worst performance. This gap was particularly prominent on GF8839, for the method

TABLE III
ACCURACY COMPARISON OF ZH1 AND ZH9

Method	zh1		zh9	
	OA (%)	Kappa	OA (%)	Kappa
Patch-wise CNN	93.66	0.8848	85.30	0.7762
SLIC + Random Forest	93.83	0.9478	89.43	0.8847
SLIC + ResNet-50	96.73	0.9410	95.10	0.9670
Ecognition + IOCNN	96.59	0.9812	89.66	0.9656
SLIC + IOCNN	97.57	0.9856	95.11	0.9673
ISLIC + IOCNN	**98.17**	**0.9885**	**96.07**	**0.9708**

TABLE IV
ACCURACY COMPARISON OF GF4454 AND GF8839

Method	GF4454		GF8839	
	OA (%)	Kappa	OA (%)	Kappa
Patch-wise CNN	89.28	0.8069	87.55	0.8059
SLIC + Random Forest	96.88	0.9505	97.85	0.9702
SLIC + ResNet-50	98.05	0.9673	99.51	0.9928
Ecognition + IOCNN	95.70	0.9926	98.26	0.9990
SLIC + IOCNN	98.96	0.9918	99.43	0.9915
ISLIC + IOCNN	**98.99**	**0.9931**	**99.85**	**0.9990**

TABLE V
PARAMETER SIZE AND COMPUTATION COMPLEXITY COMPARISON

Method	Parameter size (MB)	Flops (G)
ResNet-50	23.55	4.11
IOCNN	23.84	4.12

composed of ISLIC and IOCNN achieved 99.85% OA and κ of 0.9990 while patchwise CNN achieved 87.55% OA and κ of 0.8059. Compared with patchwise CNN, other methods that combine the segmentation algorithm and classification model performed more stable. Specifically, the patchwise CNN method only achieved a relatively ideal accuracy assessment on zh1 (93.66% OA and κ of 0.8848), but the classification accuracies on other images were consistently less than 90%. By contrast, the SLIC and random forest method obtained higher accuracy on three images. With the SLIC, ResNet-50, and IOCNN were also tested, respectively. As the results show, our constructed CNN model obtained the best classification assessments, and its robustness is more excellent. It can be seen from its higher accuracy that it exceeds 95% on each image. The methods composed of the same classification model (IOCNN) and different segmentation algorithms (ecognition, SLIC, and ISLIC) were also tested and ISLIC achieves the best results. Furthermore, by adapting the ISLIC, the classification accuracy was successfully increased by about 1%. All the abovementioned comparisons indicate that the proposed segmentation algorithm and CNN model could effectively improve the classification accuracy, respectively.

The boundary performances of the results that generated by different methods were also evaluated here. As shown in Tables VI and VII , it can be seen that the evaluation results are consistent with the results of classification assessments. Specifically, the improved method achieved the best boundary performance, in contrast, the patchwise CNN method obtained the worst boundary performance. Furthermore, compared with the patchwise CNN method, the results of other methods that use the object-oriented image analysis strategy are dramatically

TABLE VI
BOUNDARY COMPARISON OF ZH1 AND ZH9

Method	zh1			zh9		
	Recall (%)	Precision (%)	F1-Score (%)	Recall (%)	Precision (%)	F1-Score (%)
Patch-wise CNN	91.83	87.45	89.59	74.88	71.99	73.41
SLIC + Random Forest	89.60	90.33	89.97	79.70	81.44	80.56
SLIC + ResNet-50	93.52	95.46	94.48	90.77	91.36	91.06
Ecognition + IOCNN	93.45	92.97	93.21	80.28	81.17	80.72
SLIC + IOCNN	96.17	96.16	96.16	90.71	91.44	91.07
ISLIC + IOCNN	**97.00**	**97.08**	**97.04**	**91.89**	**92.25**	**92.07**

TABLE VII
BOUNDARY COMPARISON OF GF4454 AND GF8839

Method	GF4454			GF8839		
	Recall (%)	Precision (%)	F1-Score (%)	Recall (%)	Precision (%)	F1-Score (%)
Patch-wise CNN	97.38	81.10	88.50	97.78	76.58	85.89
SLIC + Random Forest	95.14	97.83	96.47	96.72	99.10	97.90
SLIC + ResNet-50	98.55	96.03	97.28	99.83	99.22	99.52
Ecognition + IOCNN	93.33	95.84	94.56	98.23	99.06	98.64
SLIC + IOCNN	**98.64**	98.79	**98.72**	99.84	99.03	99.43
ISLIC + IOCNN	98.15	**98.90**	98.53	**99.85**	**99.89**	**99.87**

TABLE VIII
CLASSIFICATION COMPARISON OF ZH1

Method	Recall per category (%)							
	Road	Building	Tree	Grass	Dryland	River	Railway	SP
U-Net [31]	92.57	92.93	78.3	97.19	99.38	-	89.72	89.66
K-Net [35]	91.98	91.17	84.01	97.37	99.15	-	93.34	97.00
PSPNet [32]	89.65	91.45	76.50	96.37	98.87	-	88.09	87.55
Segmenter [34]	62.98	80.38	41.57	89.4	96.31	-	82.45	75.78
Ours	**96.91**	**98.74**	**94.46**	**98.8**	**99.99**	-	**96.10**	**99.11**

TABLE IX
CLASSIFICATION COMPARISON OF ZH9

Method	Recall per category (%)							
	Road	Building	Tree	Grass	Dryland	River	Railway	SP
U-Net [31]	93.13	91.83	85.58	93.44	96.14	96.06	96.98	-
K-Net [35]	**95.71**	92.39	88.15	96.58	95.93	**97.07**	**97.83**	-
PSPNet [32]	93.39	91.78	83.21	93.86	92.74	95.2	95.60	-
Segmenter [34]	86.76	80.47	65.77	86.38	81.48	96.35	92.46	-
Ours	95.34	**96.51**	**93.75**	**98.64**	**97.37**	96.66	97.42	-

TABLE X
BOUNDARY COMPARISON OF ZH1 AND ZH9

Method	zh1			zh9		
	Recall (%)	Precision (%)	F1-Score (%)	Recall (%)	Precision (%)	F1-Score (%)
U-Net [31]	90.40	98.23	94.15	85.25	96.96	90.73
K-Net [35]	89.75	**98.45**	93.90	87.66	**97.45**	**92.29**
PSPNet [32]	88.29	97.62	92.72	85.21	96.61	90.55
Segmenter [34]	73.63	84.42	78.66	70.08	87.67	77.90
Ours	**97.08**	97.00	**97.04**	**92.25**	91.89	92.07

improved. Besides, it can be also seen that the methods that use the CNN method can improve the boundary performance.

The improvements of boundary performance can be seen in the classification maps. As shown in Fig. 9(b) , the classification map generated by the patchwise method contained a lot of salt-and-pepper noise, which blurred the edge of the ground object. Turning to Fig. 9(c), the misclassification of some small objects, especially railways, was also improved due to using of object-oriented method. However, the problem of noisy pixels still existed here, and there is a lot of voids in classified objects. Fortunately, in Fig. 9(d), by using ecognition segmentation, the problem about noisy pixels is significantly successfully solved. Nevertheless, the problem of misclassification still existed and some details in regions are misclassified due to the undersegmentation strategy used in the method of ecognition. By using the opposite strategy, oversegmentation, the methods objects [see Fig. 9(f)]. Moreover, with using the proposed that composed of SLIC and CNN seems to have a better classification performance classification with fewer holes on segmentation algorithm, these defects were further improved [see Fig. 9(g)], which indicated

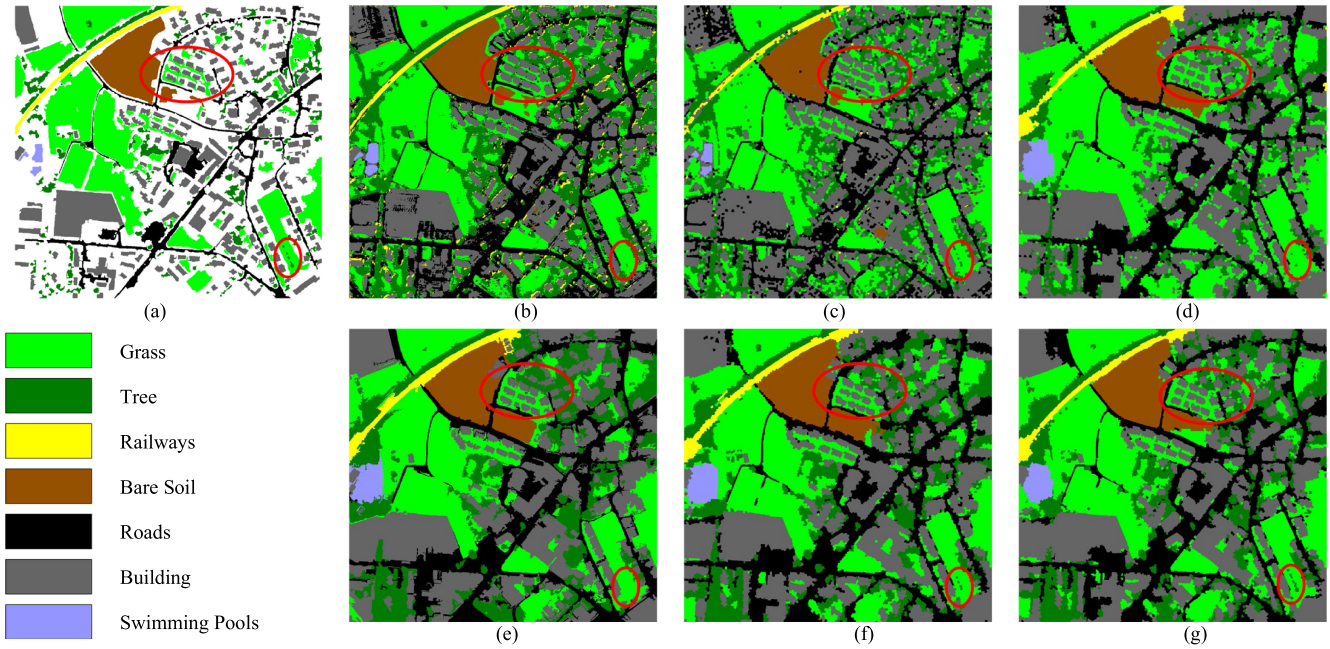


Fig. 9. Classification results on zh1 data set, with (a) the ground truth, the classification results based on method composed of (b) patch-wise CNN, (c) SLIC and radom forest, (d) SLIC and ResNet-50, (e) ecognition and IOCNN, (f) SLIC and IOCNN, (g) ISLIC and IOCNN.

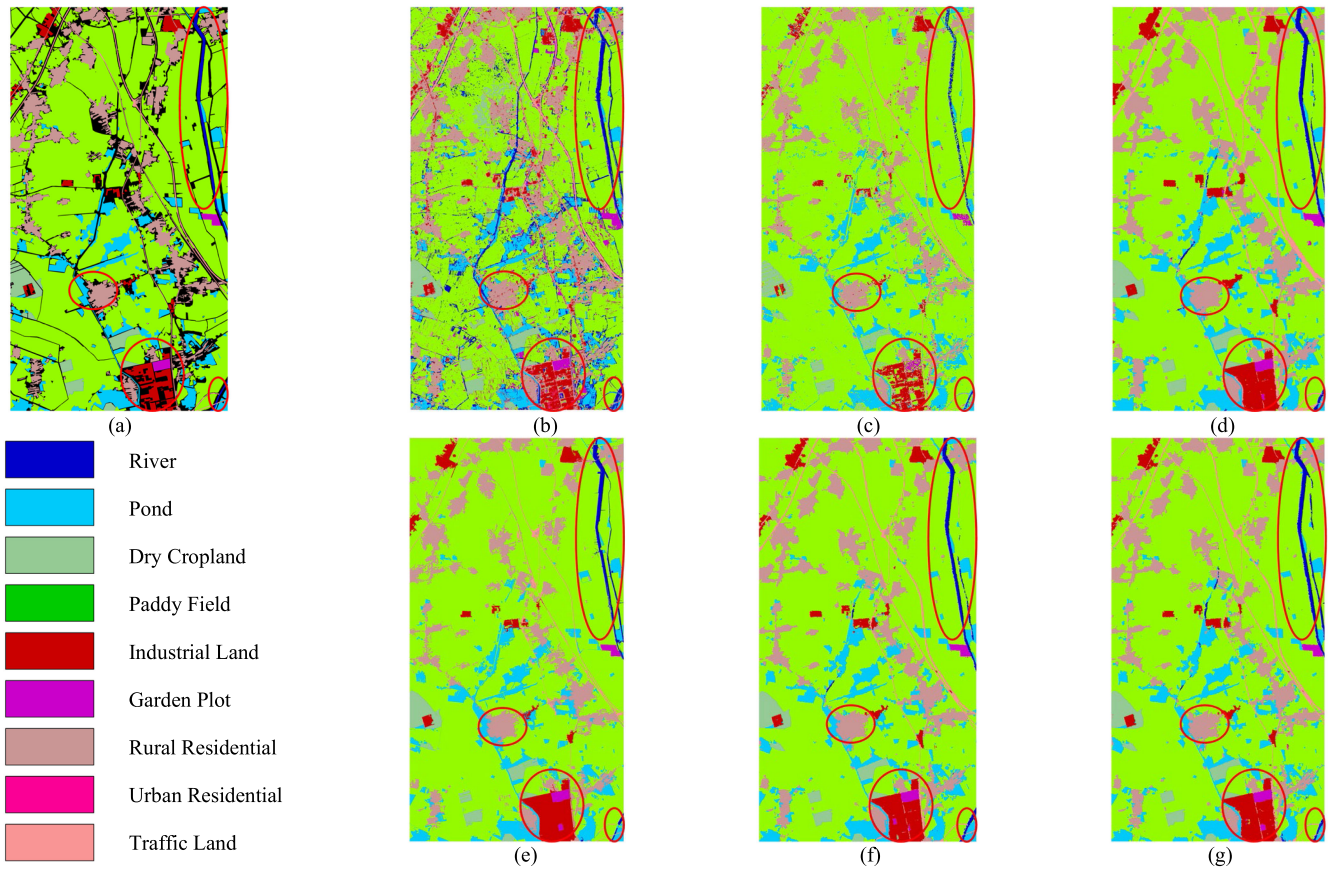


Fig. 10. Classification results on GF4454 data set, with (a) the ground truth, the classification results based on composed of (b) patchwise CNN, (c) SLIC and radom forest, (d) SLIC and ResNet-50, (e) ecognition and IOCNN, (f) SLIC and IOCNN, (g) ISLIC and IOCNN. For show more details, only the left half of them are shown here.

the proposed method could effectively help to more accurate classification. The same improvements can also be discovered in the resulting classification maps of GF4454, which are shown in Fig. 10. From these figures, it can also be seen that the problems of misclassification and voids have been well decreased and the object boundaries are also preserved well.

Besides, we compared the effectiveness of the improved CNN model with ResNet-50. As shown in Table V, it can be found that the improved CNN model increases the parameter size of the original ResNet-50 by 0.29M, and the floating-point calculation operations per second (Flops) is almost unchanged.

3). Comparison to General Methods: To further evaluate the effectiveness of our method, we conducted comprehensive experiments on zh1 and zh9. Four state-of-the-art end-to-end semantic segmentation CNN model are compared in this section, including U-Net [31], K-Net [35], PSPNet [32], and Segmenter [34]. Due to the framework of these semantic segmentation methods, 20 images of Zurich Dataset are all used as the train images to predict the classification result of zh1 and zh9. Random crop with size of 512×512 and random flip are used to augment samples. Each model is trained with 40 000 iterations instead of epoch, using the SGD with a momentum of 0.9 and a weight decay of 0.0005 to optimize an initial learning rate 0.1. Noting, compared with these methods, our method is a kind of full pixel classification method and thus the value of background/clutter is not contained here. Therefore, only recall is used as the metric to evaluate their classification performance. Besides, the boundary performance is also evaluated here with metrics of boundary recall, boundary precision, and f1-score.

As shown in Tables VIII and IX, apart from the classification of railway on zh9, our proposed method achieves the highest recall. For the railway classification on zh9, our method can also nearly achieve the same performance. Comparisons for boundary performance of these methods are shown in Table X. It can be also seen that our proposed method obtains the best boundary performance on zh1. For zh9, our method achieves the highest boundary precision and the second best f1-socre 92.07%. Therefore, it can be seen our method is more stable and accurate than these state-of-the-art methods in classification, moreover, it also has smoother boundaries in segmentation.

V. Conclusion

In this article, an improved object-oriented analysis method has been presented to complete the task of land use and land cover classification. Two basic steps in this improved method are improved, i.e., the segmentation and classification. Specifically, for the better segmentation of RS images, an improved SLIC method that considers more spectral features and uses filtering is proposed to fully utilize arbitrary bands. In addition, for the better classification, an improved CNN model that integrates the first-order and second-order features is proposed. By using attention mechanism and the fusion representation of different order features, the improved CNN model obtains better feature expression capabilities to improve the classification accuracy. To verify the effectiveness of the abovementioned two improvements, the improved segmentation and classification parts are all compared through ablation experiments. Through these experiments on four real RS images, it is proved that the three improvements in the proposed ISLIC are all effective ways to improve the boundary performance and the proposed CNN model is also an effective way to better express semantic information to improve the classification accuracy. Besides, the improved object-oriented method has also been compared with some general end-to-end CNN methods and it can be found that our method can obtain better classification performance. However, since our method divides the process into two parts, the computational complexity of our method will be higher than these end-to-end methods. In conclusion, the proposed object-oriented CNN method is an effective image classification algorithm, and a good segmentation method and CNN model can effectively improve the classification performance.

References

[1] J. E. Patino and J. C. Duque, "A review of regional science applications of satellite remote sensing in urban settings," *Comput., Environ. Urban Syst.*, vol. 37, pp. 1–17, 2013.

[2] A. Karpatne, I. Ebert-Uphoff, S. Ravela, H. A. Babaie, and V. Kumar, "Machine learning for the geosciences: Challenges and opportunities," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 8, pp. 1544–1554, Aug. 2019.

[3] G. Pan, G. Qi, Z. Wu, D. Zhang, and S. Li, "Land-use classification using taxi GPS traces," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 113–123, Mar. 2013.

[4] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015.

[5] F. Agüera, F. J. Aguilar, and M. A. Aguilar, "Using texture analysis to improve per-pixel classification of very high resolution images for mapping plastic greenhouses," *ISPRS J. Photogrammetry Remote Sens.*, vol. 63, no. 6, pp. 635–646, 2008.

[6] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[7] X. Liu, C. Kang, L. Gong, and Y. Liu, "Incorporating spatial interaction patterns in classifying and understanding urban land use," *Int. J. Geographical Inf. Sci.*, vol. 30, no. 2, pp. 334–350, 2016.

[8] U. Maulik and D. Chakraborty, "Remote sensing image classification: A survey of support-vector-machine-based advanced techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 33–52, Mar. 2017.

[9] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 41–57, Jun. 2016.

[10] B. Zhao, Y. Zhong, and L. Zhang, "A spectral–structural bag-of-features scene classifier for very high spatial resolution remote sensing imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 116, pp. 73–85, 2016.

[11] M. Herold, X. Liu, and K. C. Clarke, "Spatial metrics and image texture for mapping urban land use," *Photogrammetric Eng. Remote Sens.*, vol. 69, no. 9, pp. 991–1001, 2003.

[12] S. W. Myint, "A robust texture analysis and classification approach for urban land-use and land-cover feature discrimination," *Geocarto Int.*, vol. 16, no. 4, pp. 29–40, 2001.

[13] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS J. Photogrammetry Remote Sens.*, vol. 65, no. 1, pp. 2–16, 2010.

[14] T. Blaschke *et al.*, "Geographic object-based image analysis–towards a new paradigm," *ISPRS J. Photogrammetry Remote Sens.*, vol. 87, pp. 180–191, 2014.

[15] Y. Zhong, B. Zhao, and L. Zhang, "Multiagent object-based classifier for high spatial resolution imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 841–857, Feb. 2014.

[16] Z. Lv, P. Zhang, and J. Atli Benediktsson, "Automatic object-oriented, spectral-spatial feature extraction driven by Tobler's first law of geography for very high resolution aerial imagery classification," *Remote Sens.*, vol. 9, no. 3, 2017, Art. no. 285.

[17] R. Oliva-Santos, F. Maciá-Pérez, and E. Garea-Llano, "Ontology-based topological representation of remote-sensing images," *Int. J. Remote Sens.*, vol. 35, no. 1, pp. 16–28, 2014.

[18] A. Troya-Galvis, P. Gançarski, N. Passat, and L. Berti-Equille, "Unsupervised quantification of under-and over-segmentation for object-based remote sensing image analysis," *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 8, no. 5, pp. 1936–1945, May 2016.

[19] H. Yoshida and M. Omae, "An approach for analysis of urban morphology: Methods to derive morphological properties of city blocks by using an urban landscape model and their interpretations," *Comput., Environ. Urban Syst.*, vol. 29, no. 2, pp. 223–247, 2005.

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

[21] D. P. Sullivan *et al.*, "Deep learning is combined with massive-scale citizen science to improve large-scale image classification," *Nature Biotechnol.*, vol. 36, no. 9, pp. 820–828, 2018.

[22] X. Yang, X. Qian, and T. Mei, "Learning salient visual word for scalable mobile image retrieval," *Pattern Recognit.*, vol. 48, no. 10, pp. 3093–3101, 2015.

[23] E. Othman, Y. Bazi, N. Alajlan, H. Alhichri, and F. Melgani, "Using convolutional features and a sparse autoencoder for land-use scene classification," *Int. J. Remote Sens.*, vol. 37, no. 10, pp. 2149–2167, 2016.

[24] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.

[25] G.-S. Xia *et al.*, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[26] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[27] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3322–3337, Jun. 2017.

[28] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 117, pp. 11–28, 2016.

[29] K. Nogueira, O. A. Penatti, and J. A. Dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, 2017.

[30] L. Li, J. Han, X. Yao, G. Cheng, and L. Guo, "DLA-MatchNet for few-shot remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7844–7853, Sep. 2021.

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

[32] 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.

[33] H. Zhao *et al.*, "PSANET: Point-wise spatial attention network for scene parsing," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 267–283.

[34] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, Art. no. 34.

[35] W. Zhang, J. Pang, K. Chen, and C. C. Loy, "K-net: Towards unified image segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, Art. no. 34.

[36] X. Yang *et al.*, "An attention-fused network for semantic segmentation of very-high-resolution remote sensing imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 177, pp. 238–262, 2021.

[37] Z. Zheng, Y. Zhong, J. Wang, and A. Ma, "Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4096–4105.

[38] F. Zhou, R. Hang, and Q. Liu, "Class-guided feature decoupling network for airborne image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2245–2255, Mar. 2021.

[39] F. Zhou, R. Hang, H. Shuai, and Q. Liu, "Hierarchical context network for airborne image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 4407612.

[40] X. Yao, Q. Cao, X. Feng, G. Cheng, and J. Han, "Scale-aware detailed matching for few-shot aerial image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5611711.

[41] F. Sharifzadeh, G. Akbarizadeh, and Y. Seifi Kavian, "Ship classification in SAR images using a new hybrid CNN–MLP classifier," *J. Indian Soc. Remote Sens.*, vol. 47, no. 4, pp. 551–562, 2019.

[42] N. Davari, G. Akbarizadeh, and E. Mashhour, "Intelligent diagnosis of incipient fault in power distribution lines based on corona detection in UV-visible videos," *IEEE Trans. Power Del.*, vol. 36, no. 6, pp. 3640–3648, Dec. 2020.

[43] S. Liu, Y. Zheng, Q. Du, A. Samat, X. Tong, and M. Dalponte, "A novel feature fusion approach for VHR remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 464–473, 2020.

[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] W. Zhao, Z. Guo, J. Yue, X. Zhang, and L. Luo, "On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery," *Int. J. Remote Sens.*, vol. 36, no. 13, pp. 3368–3379, 2015.

[46] W. Zhao and S. Du, "Learning multiscale and deep representations for classifying remotely sensed imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 113, pp. 155–165, 2016.

[47] W. Zhao *et al.*, "Superpixel-based multiple local CNN for panchromatic and multispectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4141–4156, Jul. 2017.

[48] C. Zhang *et al.*, "An object-based convolutional neural network (OCNN) for urban land use classification," *Remote Sens. Environ.*, vol. 216, pp. 57–70, 2018.

[49] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[50] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 705–718.

[51] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 13–26.

[52] S.-H. Lee, W.-D. Jang, and C.-S. Kim, "Contour-constrained superpixels for image and video processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2443–2451.

[53] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

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

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

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

[57] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[58] 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.

[59] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1449–1457.

[60] A. Cherian and S. Gould, "Second-order temporal pooling for action recognition," *Int. J. Comput. Vis.*, vol. 127, no. 4, pp. 340–362, 2019.

[61] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3024–3033.

[62] X. Lv, Z. Shao, D. Ming, C. Diao, K. Zhou, and C. Tong, "Improved object-based convolutional neural network (IOCNN) to classify very high-resolution remote sensing images," *Int. J. Remote Sens.*, vol. 42, no. 21, pp. 8318–8344, 2021.

[63] E. Li, A. Samat, W. Liu, C. Lin, and X. Bai, "High-resolution imagery classification based on different levels of information," *Remote Sens.*, vol. 11, no. 24, 2019, Art. no. 2916.

[64] S. Liu *et al.*, "A multi-scale superpixel-guided filter feature extraction and selection approach for classification of very-high-resolution remotely sensed imagery," *Remote Sens.*, vol. 12, no. 5, 2020, Art. no. 862.

[65] D. Stutz, "Superpixel segmentation: An evaluation," in *Proc. German Conf. Pattern Recognit.*, 2015, pp. 555–562.

[66] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[67] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
[68] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[69] C. Ionescu, O. Vantzos, and C. Sminchisescu, "Matrix backpropagation for deep networks with structured layers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2965–2973.
[70] P. Li, J. Xie, Q. Wang, and Z. Gao, "Towards faster training of global covariance pooling networks by iterative matrix square root normalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 947–955.
[71] M. Volpi and V. Ferrari, "Semantic segmentation of urban scenes by learning local class interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 1–9.
[72] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
[73] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[74] 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.
[75] I. Arganda-Carreras *et al.*, "Crowdsourcing the creation of image segmentation algorithms for connectomics," *Front. Neuroanat.*, vol. 9, 2015, Art. no. 142.

Zhiqing Li received the B.S. degree in electronic and information engineering from the School of Electronical and Information Engineering, South-West University, Chongqing, China, in 2020. He is currently working toward the M.S. degree with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China.

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

Erzhu Li received the M.S. degree in photogramme try and remote sensing from the China University of Mining and Technology, Xuzhou, China, in 2014, and the Ph.D. degree in cartography and geographic information system from Nanjing University, Nanjing, China, in 2017.

He is currently an Associate Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include high-resolution image processing and computer vision in urban remote sensing applications.

Alim Samat (Member, IEEE) received the B.S. degree in geographic information system from Nanjing University, Nanjing, China, 2009, the M.E. degree in photogrammetry and remote sensing from the China University of Mining and Technology, Xuzhou, China in 2012, and the Ph.D. degree in cartography and geography information system from Nanjing University, in 2015.

He is currently an Associate Professor with the State Key Laboratory of Desert and Oasis Ecology, Xinjiang Institute of Ecology and Geography, Chinese Academy of Sciences, Beijing, China. His main research interests include optical and PolSAR image processing and applications, machine learning, and pattern recognition.

Tianyu Xu received the B.S. degree in geographic information science from the School of Surveying and Geo-Informatics, Shandong Jianzhu University, Jinan, China, in 2019. He is currently working toward the M.S. degree with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China.

His research interest includes remote sensing, image processing, and machine learning.

Wei Liu received the M.S. degree in cartography and geographic information engineering from the China University of Mining and Technology, Xuzhou, China, in 2007, and the Ph.D. degree in cartography and geographic information engineering from the China University of Mining and Technology, Xuzhou, China, in 2010.

He is currently an Associate Professor with the School of Geography, Geomatics and Planning, Jiangsu Normal University, Xuzhou, China. His research interests include spatial data quality checking, high-resolution remote sensing image processing, and GIS development and applications.

Yihu Zhu received the Graduate degree in surveying and mapping engineering from Wuhan University, Wuhan, China, in 2005.

He is the Vice President with the Jiangsu Institute of geological surveying and mapping and a Professorate Senior Engineer. His current research interests include photogrammetry, remote sensing, and GIS.