Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 2472726, 13 pages
https://doi.org/10.1155/2021/2472726

Research Article

Semantic Segmentation of Remote Sensing Image Based on Convolutional Neural Network and Mask Generation

Binglin Niu

The School of Information Engineering, Xinyang Agriculture and Forestry University, Xinyang 464000, China

Correspondence should be addressed to Binglin Niu; nbl@xyafu.edu.cn

Received 12 April 2021; Revised 18 May 2021; Accepted 25 May 2021; Published 2 June 2021

Academic Editor: Yi-Zhang Jiang

High-resolution remote sensing images usually contain complex semantic information and confusing targets, so their semantic segmentation is an important and challenging task. To resolve the problem of inadequate utilization of multilayer features by existing methods, a semantic segmentation method for remote sensing images based on convolutional neural network and mask generation is proposed. In this method, the boundary box is used as the initial foreground segmentation profile, and the edge information of the foreground object is obtained by using the multilayer feature of the convolutional neural network. In order to obtain the rough object segmentation mask, the general shape and position of the foreground object are estimated by using the high-level features in the process of layer-by-layer iteration. Then, based on the obtained rough mask, the mask is updated layer by layer using the neural network characteristics to obtain a more accurate mask. In order to solve the difficulty of deep neural network training and the problem of degeneration after convergence, a framework based on residual learning was adopted, which can simplify the training of those very deep networks and improve the accuracy of the network. For comparison with other advanced algorithms, the proposed algorithm was tested on the Potsdam and Vaihingen datasets. Experimental results show that, compared with other algorithms, the algorithm in this article can effectively improve the overall precision of semantic segmentation of high-resolution remote sensing images and shorten the overall training time and segmentation time.

1. Introduction

The task of high-resolution remote sensing image semantic segmentation is to assign semantic labels to each pixel in the remote sensing image. In recent years, with the rapid development of remote sensing mapping technology, it is easy to obtain ultrahigh-resolution optical remote sensing images with a ground sampling distance (GSD) of 5–10 cm [1]. The urban high-resolution remote sensing image is mainly composed of artificial buildings, supplemented by some natural vegetation land. Artificial buildings mainly include houses, airports, roads, and bridges. If the housing target is accurately segmented, urban land use indicators such as residential density can be quickly obtained, which provides a basis for further urban planning. Therefore, understanding the context semantics of the image accurately and labeling the pixels above the image become the research hotspot in the field of remote sensing image segmentation.

High-resolution remote sensing image contains rich semantic information that most of the traditional methods cannot effectively represent; the segmentation effect is not ideal. In the early days, the segmentation of artificial features and buildings mainly relied on vectorization model extraction technology [2], such as region-based segmentation, line analysis, and shadow analysis. A large number of studies rely on artificial design features, and the segmentation operation is realized by the supervised classifier. Artificial features often have poor generalization when they represent high-level semantic information. Deep learning technology can not only automatically extract features [3] but also fully mine the advanced semantic information features in the image.

Deep learning technology has achieved great success in the field of computer vision, such as image classification [4], object detection [5], and semantic segmentation [6]. The deep convolution neural network receives the input of the

original image data, learns from the end-to-end structure, and obtains the final segmentation result according to the specific task. In the field of remote sensing image interpretation, deep convolution neural network has been widely studied. The semantic information contained in urban remote sensing images is complex [7], which has the characteristics of various artificial targets and many small targets [8]. Easily confused data are often adjacent or staggered in spatial distribution, which makes segmentation difficult.

The fully convolutional neural network (FCN) was first applied to remote sensing image segmentation tasks [9]. It can accept image input and testing of any size and avoid the repeated storage and calculation caused by the use of pixel blocks. The problem of convolution is more efficient than traditional convolutional neural networks (CNN) with full connections, but its segmentation results are not fine enough, and the details of the image are not complete enough. On this basis, the "hourglass-shaped network" includes DeconvNet [10], SegNet [11], U-Net [12], and DeepUNet [13]. Other methods have been proposed and applied to remote sensing image segmentation. These networks have made different adjustments in their decoder structure and achieved higher segmentation accuracy.

Based on the hole convolution, DeepLab [14] proposed a spatial pyramid module, which applies multisampling rate hole convolution, multireceptive field convolution, or pooling operations on the input feature map to explore multiscale context information. DeepLabV3+ [15] not only uses separable convolution but also introduces the encoder-decoder structure commonly used in semantic segmentation in order to fuse multiscale information, gradually recovering spatial information to capture clear segmentation target boundaries and refine the segmentation results. This is a network framework model that performs well in the field of image semantic segmentation.

Many methods have been tried to classify high-resolution remote sensing images. Literature [16] used support vector machine (SVM) to classify remote sensing image objects. Literature [17] used an unsupervised clustering algorithm to segment houses in remote sensing images. Literature [18] improved the traditional edge detection method so that small objects in remote sensing images could also be segmented. Remote sensing images contain rich spectral information, so traditional feature extraction methods cannot achieve good segmentation results. From the perspective of pattern recognition, the selection of typical features is the bottleneck to improve the recognition accuracy [19]. It is impossible to accurately classify all types of ground objects by only using a specific set of features. Therefore, automatic learning and classification of features in the corresponding dataset can improve target classification accuracy more effectively than manual design features [20].

In order to improve the semantic segmentation accuracy of high-resolution remote sensing images, a semantic segmentation method based on CNN and mask generation is proposed in this article. Firstly, an edge extraction method based on an iterative GMM model is used to fuse CNN multilayer features through the iterative method. One-layer feature map is used as feature input in each round of iterative operation. Then, the manually marked bounding box is used as the initial value of the foreground target contour, and the segmentation mask is modified step by step using the GMM model. Experiments show that the proposed algorithm has better semantic segmentation performance on the CCF dataset.

The rest of the article is organized as follows. In Section 2, the related work is introduced. In Section 3, the method proposed by this article is introduced. Section 4 gives the network training. Section 5 gives the experimental results. At last, Section 6 draws the conclusion of this article.

2. Related Work

Image semantic segmentation is to segment the input image into multiple semantic regions, that is, to assign a semantic category to each pixel in the image. In recent years, a lot of studies have been conducted on image semantic segmentation based on deep learning at home and abroad. At present, the mainstream methods are based on the fully convolution network [6], aiming at the feature encoding and decoding [21], expanding the receiving domain of convolution operation [22], and making multiscale improvements on feature fusion.

Deep learning requires a large number of training samples, and the topic of image semantic segmentation needs to label each pixel in the training samples. On the one hand, pixel-level labeling is difficult. On the other hand, a large number of sample labels mean a high cost of manual labeling. Therefore, image semantic segmentation based on weak supervision has become a research hotspot. The annotation methods used in weakly supervised image semantic segmentation include boundary box annotation, point annotation, and image-level annotation. In many labeling methods with poor supervision, the time of boundary layer labeling is short and the segmentation accuracy is high. Using the method in [23], the segmentation accuracy of the weakly supervised training result based on bounding box annotation can reach 88.2% of the training result based on pixel annotation. According to its research, the time of pixel-level annotation is 15 times that of bounding box annotation. Therefore, under the same annotation workload, more training samples can be obtained by using boundary box annotation, and a segmentation model with better generalization and robustness can be obtained.

The theory of deep learning has received extensive attention since it was put forward. The basic motivation of deep learning is to build a deep neural network to simulate the learning and analysis mechanism of the human brain. Compared with traditional machine learning algorithms, deep learning emphasizes the automatic learning of features from huge data through multilayer neuron organization. Typical deep learning structures include recurrent neural network (RNN), deep belief network (DBN), and CNN [24]. CNN has achieved remarkable results in computer vision tasks, such as image classification and target recognition, and has achieved excellent results in the competition of authoritative datasets in ImageNet, PASCAL VOC, and

other fields. Literature [25] researched and designed a 7-layer CNN model (named AlexNet) and won the championship of the LVSRC (ImageNet Large Scale Visual Recognition Challenge) competition. Many scholars conducted semantic analysis research on remote sensing images based on the CNN method. Literature [26] proposed a 5-layer network structure to complete the target classification of remote sensing images. Hu et al. [27] used a pretrained CNN model to classify different remote sensing image scenes. Mnih [28] proposed a CNN-based large-scale context feature learning structure for aerial images, but the effect still needs to be improved. The number of pixels in high-resolution images is huge, so it is very difficult to achieve pixel-by-pixel classification. The current pixel-level target classification accuracy is not ideal.

The key problem of semantic segmentation of weakly supervised images is how to use weak annotation to conduct supervised training on semantic segmentation networks. For boundary box annotation, the existing research methods can be divided into two categories: response area extraction method and pseudolabel-based method.

The method based on response region extraction is usually to train the FCN network directly by designing specific regular items. Literature [29] carried out similarity constraint on the categories of adjacent pixels in the output results of FCN and added this constraint to the loss function as a regular term. Literature [30] used the target loss function based on class activation mapping (CAM) and added a regular term of regional similarity based on global weighting pooling and a regular term of segmentation boundary based on conditional random field (CRF).

Since regular terms are usually calculated using the prediction results of image segmentation, they rely more on high-level semantic features. The high-level features have a large receptive field and are not sensitive enough to the edges of the target object. Therefore, the accuracy of the semantic segmentation model trained by such methods is relatively low. The cut and paste method generates "fake" samples by segmenting high-response regions and uses generative confrontation networks to improve the accuracy of segmentation of high-response regions [31].

The pseudolabel-based method is to generate pseudolabel masks by using multiscale combination grouping, CRF, GrabCut, and other methods and train pseudolabel masks as alternative data for pixel-level labeling [32]. In the bounding box supervision method, the multiscale combined grouping method is adopted to generate candidate pseudolabel areas. The IoU of the pseudolabel areas and the labeled areas of the boundary box are used as weight parameters, and all pseudolabels are used for weighted training [33]. According to the research, multiple iterations of training can gradually reduce the noise of pseudotags, thus improving the segmentation accuracy. The weakly semisupervised method adopted the EM algorithm as the iterative training mode and adopted the CRF algorithm to generate fake tags. GrabCut is used in mask sort method to generate pseudotags and takes the output of the FCN network as a unary constraint item in the GrabCut target function.

The probability graph model is used in the existing pseudotag generation methods, such as CRF and GrabCut. Most of them only use the underlying image features as binary constraints. Rajchl [23] used only the colors of the three channels in the RGB image, while Khoreva [34] used only the edge features. The underlying image features do not contain semantic information. If the input image includes foreground objects with more complex colors, such methods are prone to produce high-noise pseudolabels. In order to solve the problem of lack of semantic information, most of these methods use high-level features as the input of unary constraints. When high-level features and low-level features are used as part of the objective function, the training process is separated from each other, and the multiscale image features in the FCN network are not fully utilized. Although the influence of pseudolabel noise can be reduced through multiple iterations during the training process, it will also double the training time.

3. Algorithm Implementation

The framework of the weak supervision training method is shown in Figure 1. For each training sample, the training process is divided into three steps. The first step is to move forward. For the input training sample image, the residual network module is used for multilevel feature extraction. The second step is mask generation. Firstly, multilayer image features are extracted from the residual network module. The third step is backpropagation training. The mask is used as the monitoring information to complete the backpropagation training of the residual network module.

3.1. Residual Network. A deep convolution neural network has made a series of breakthroughs in the field of image classification. Recent research results show that the depth of the model plays a crucial role, and many recognition algorithms benefit from very deep models.

However, the deeper the neural network is, the more difficult it is to train it. In the case that the deep network can converge, degradation will occur, and the accuracy tends to saturation with the increase of the network depth. To solve this problem, a residual learning-based framework is used here, which simplifies training for those very deep networks. The optimal solution mapping is represented by $H(x)$, where $x$is the input to these layers. The stacked nonlinear layer is used to fit another mapping $H(x)$: $= H(x) - x$, and then the original optimal solution mapping $H(x)$ is rewritten as $F(x) + x$. Assuming that the residual mapping is easier to optimize than the original mapping, even in extreme cases where mapping is optimizable, it is easy to push the residual to 0. It is much easier to push the residuals to 0 than to approximate the map to another nonlinear layer.

The residual learning algorithm is adopted on the stacked layer. A building block is shown in Figure 2. The definition of the building block is as follows:

$$y = F(x, \{W_i\}) + x, \tag{1}$$

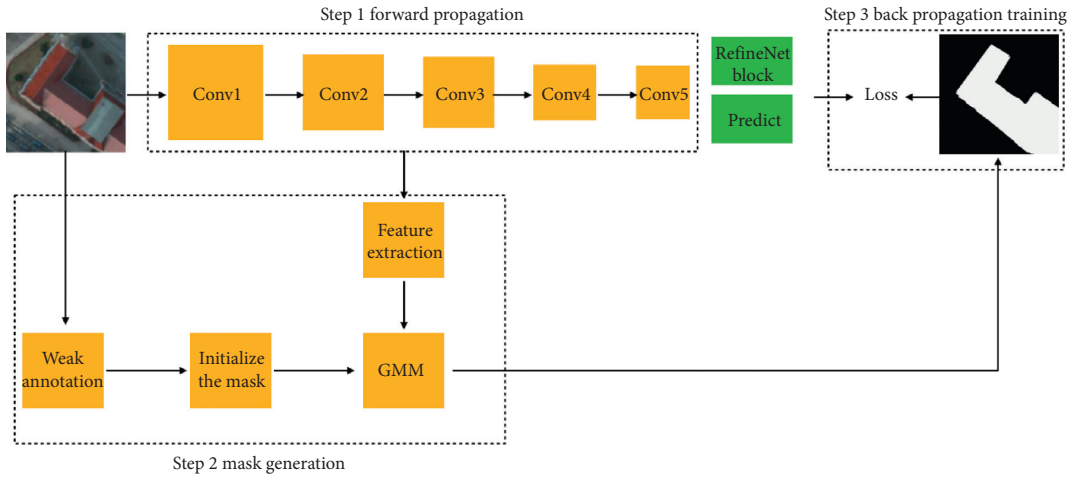

Figure 1: The image segmentation framework of this article.

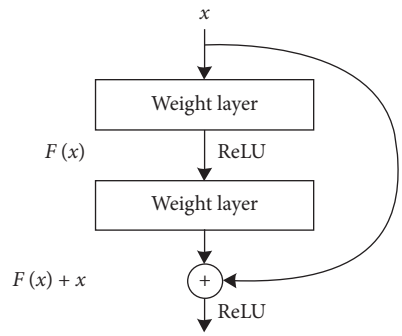

Figure 2: Residual learning building blocks.

where x and y represent the input and output of the corresponding layer, respectively.$F(x, \{W_i\})$ represents the learned residual mapping function. The example in Figure 2 contains two layers, $F = W_2\sigma(W_1x)$, where σ represents ReLU and the bias term is omitted for simplicity. $F + x$ is obtained by a quick join and element-level addition. After addition, we perform another nonlinear operation (for example, $\sigma(y)$). The quick connections described here are attractive because they do not add additional parameters or computational complexity and are especially important when comparing normal and residual networks. It allows a fair comparison of the two networks based on the same parameters, depth, width, and computational cost (except for negligible element-level addition). In the above equation, the dimensions of x and F must be the same. If there are inconsistencies, such as changing the number of channels for input and output, a linear mapping on the shortcut connection is required to match the dimensions.

$$y = F(x, \{W_i\}) + W_sx, \tag{2}$$

where W_s is a linear mapping matrix, which is only used to solve the dimension mismatch problem here.

3.2. Dynamic Mask Generation Method. The dynamic mask generation method is based on the rectangular edge marked by the boundary box, which modifies the segmentation mask used for monitoring training through iterative updating and generates the dynamic mask according to the final contour. The process is shown in Figure 1.

As shown in Figure 1, the input data of the dynamic mask generation method include boundary box annotation data and CNN multilayer feature map. Firstly, the eigenvectors of all eigengraphs need to be normalized. Second, the GMM model is initialized with the eigenvectors of each sampling point. Finally, the probability of the sampling point relative to the GMM model is calculated to complete the mask update.

3.2.1. Normalization of Feature Maps. As the midtier data extracted from the CNN network usually have no upper and lower bounds, and the eigenvalues of different dimensions vary greatly, feature bias will occur in the calculation of feature distance, so the normalization operation of feature vectors is needed.$F^0 \in R^{H\times W\times C}$ is used to represent the CNN feature graph of a layer, where H is the height of the feature graph, W is the width of the feature graph, and C is the number of channels of the feature graph, namely, the dimension of the feature vector of the sampling point. Then, the calculation equation of the normalized feature graph F is expressed as follows:

$$F_{ijc} = \frac{F^0_{ijc} - \min\limits_{x,y} F^0_{xyc}}{\max\limits_{x,y} F^0_{xyc} - \min\limits_{x,y} F^0_{xyc}}. \quad (3)$$

In the equation, $F_{ijc} = F(i, j, c)$ represents the value at coordinates (i, j, c) in the normalized feature graph F and $F^0_{ijc} = F^0(i, j, c)$ represents the value at coordinates (i, j, c) in the CNN feature graph F^0.

Due to the boundary filling operation in CNN convolution, the pixel positions in the feature map are not strictly proportional to the pixel positions of the corresponding input image. As shown in Figure 2, the data at the edge of CNN feature map are redundant data caused by the filling operation of convolution operation, so the feature map needs to be clipped during normalization.

As shown in Figure 1, in the first iteration, the boundary box annotation is taken as the initial image edge. The position and size of the bounding box scale to 1/32 of the annotated data. The foreground GMM model and background GMM model are initialized by collecting sample points based on the current edge contour. The feature vector of the GMM model adopts the normalized 1/32 feature graph F data. Then, the foreground and background GMM models were used to classify all the sampling points in the feature map, and the classification results were used as updated image edge data.

In round 1, the width and height of the mask update result graph are 1/32 of the input image, respectively. Moreover, the result graph is used as the basis for the next round of GMM model initialization, and so on. The input of the last round is characterized by the original input image. The size of the mask update result image is the same as the input image. And the classification result of the round is taken as the final output mask.

3.2.2. Mask Update. The flow of the mask update method is shown in Figure 3. Input data for each round of mask update include input foreground edge and normalized feature map. And input foreground edge is bounding box annotation or result of previous round edge update. The process of mask update is to first determine the sample of foreground GMM and background GMM, and all pixel points on the input feature map are sampling points. Two types of samples are then used to initialize the foreground GMM model G_f and background GMM model G_b, respectively. Finally, the sampling points at the edge of foreground and background are classified again.

In Figure 3, the sampling points for foreground and background are divided as follows: the sampling points inside the existing edge contour are the foreground samples, and the sampling points outside the edge contour are the background samples.

In Figure 3, the method to obtain the list of boundary sampling points is expressed as follows. If the sampling point S is adjacent to the existing edge contour, then the sampling point is added to the list of boundary sampling points.

In Figure 3, GMM is used to classify the sampling point S as follows. If $G_f(S) \rangle G_b(S)$, the pixel is classified into the foreground category. Otherwise, it is classified into the background category.

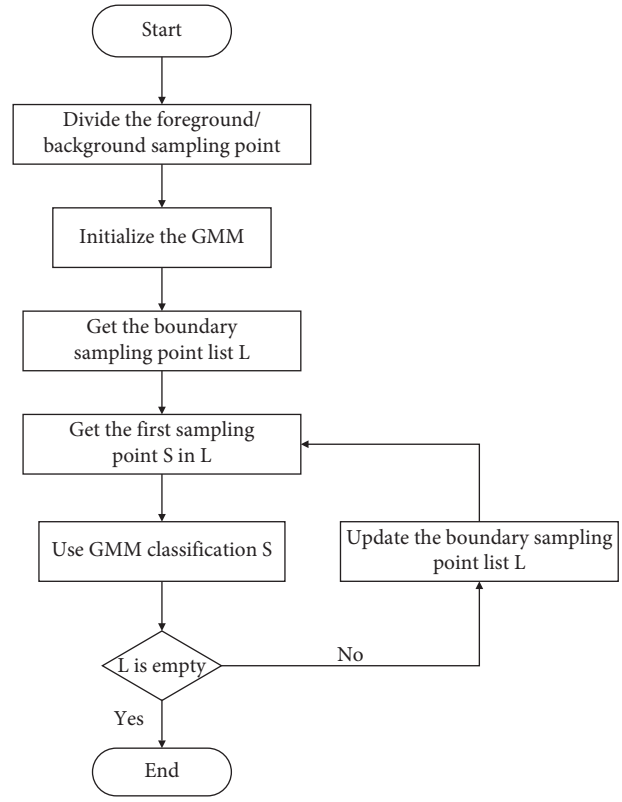

Figure 3: Mask update flow.

In Figure 3, the method to update the list of boundary sampling points is to add all sampling points adjacent to the current sampling point to the list of boundary sampling points if the classification result of the current sampling point is different from the initial value.

3.3. Semantic Segmentation Method. The dynamic mask generated in Section 3.2 serves as the monitoring information during the semantic segmentation training to provide feedback to the CNN network. In training, according to the forward propagation operation results of each input sample image, the mask is dynamically generated, and the traditional pixel-level annotation is replaced with the mask to complete the calculation of the loss function. $y \in R^{H \times W \times C}$ is used to represent the dynamic mask obtained in Section 3.2, W and H are the width and height of the input image, respectively, and C is the number of semantic categories. $h \in R^{H \times W \times C}$ is used to represent the pixel-level prediction results in Figure 3, whose width and height are also W and H. The loss function used in training is $L(\theta) = \sum_{i,j} l(h_{ij}, y_{ij})$, where $l(h_{ij}, y_{ij})$ represents softmax loss function, and h_{ij} and y_{ij}, respectively, represent the data of prediction result h and pseudolabel y at coordinates (i, j).

The weak supervisory training framework for the image semantic segmentation network is shown in Figure 2, which consists of three parts. And the three parts contain the residual network module, RefineNet module, and

segmentation mask generation module. Among them, the residual network module and RefineNet module participate in forward propagation and backpropagation during the training process, while the segmentation mask generation module only generates a temporary segmentation mask as supervision information after forward propagation. During training, a temporary mask is calculated for each forward propagation. During the test, there is no need to generate a temporary segmentation mask; only residual network module and RefineNet module participate in the calculation to obtain pixel-level prediction results.

The residual network modules were divided into five submodules of Conv1, Conv2, Conv3, Conv4, and Conv5. For the input images of width W and height H, the output feature graph dimensions of each submodule were (W/2, H/2, 64), (W/4, H/4, 256), (W/8, H/8, 512), (W/16, H/16, 1,024), and (W/32, H/32, 2048).

The RefineNet module extracts the output feature graphs of Conv2, Conv3, Conv4, and Conv5 as input data. The output of the RefineNet module is a pixel-level classification result with dimensions (W/4, H/4, $N+1$), where N represents the number of semantic categories for the foreground object. The output layer of the image semantic segmentation network is the upsampling result of the output layer of the RefineNet module with dimensions of (W, H, $N+1$).

In the split mask generation module, the output feature graphs of Conv2, Conv3, Conv4, and Conv5 were extracted from the residual network module as the input data extracted from the edges. The output mask image size of the split mask generation module was (W/4, H/4).

Because the generation process of segmentation mask relies on the semantic information contained by CNN features, the feature map extracted by CNN does not contain semantic information and cannot be used as features to extract edge information in the early stage of training. If the training method shown in Figure 2 is used directly, an effective image semantic segmentation model cannot be obtained. A pretraining process is therefore required. In the process of pretraining, only the input image is used as the feature, the mask generation method is adopted to generate the pseudolabel of all samples, and the pseudolabel is used as pixel-level labeling to complete the pretraining. After the pretraining, the training was completed using the method shown in Figure 2.

In the test process, only the residue network module and RefineNet module are needed to carry out forward propagation operation, and the image semantic segmentation results can be obtained without dynamic mask generation.

4. Network Training

In the weakly supervised semantic segmentation comparison experiment, RefineNet was used as the image semantic segmentation training framework, and ResNet101 was used as the image feature extraction network. As the size of the feature map of each CNN multilayer feature used in the segmentation mask generation module is different, GMM model parameters also need to be set differently. On the four CNN feature layers and input images extracted during the dynamic mask generation process, the number of submodels K of the GMM model is set to 3, 5, 10, 15, and 20, respectively. During the model training, the batch gradient descent method was adopted, and the sample quantity of each batch was set as 2. During the initial training, the learning rate was 5×10^{-4}, and 40 training cycles were completed. After that, the training method, as shown in Figure 2, was adopted to continue the training. The initial learning rate was 5×10^{-4}. After completing 40 training cycles, the learning rate was modified to 5×10^{-5} and 40 training cycles were completed. The momentum parameter of parameter update is 0.9, and the attenuation coefficient is 1×10^{-4}.

4.1. Data Sources and Preprocessing. The method is evaluated on the ISPRS 2D semantic labeling benchmark. There are two airborne image datasets, consisting of high-resolution true orthophoto (TOP) tiles and the corresponding digital surface models (DSMs). The Vaihingen dataset contains 33 patches (of different sizes), each consisting of a TOP extracted from a larger TOP mosaic. Each patch has 3-band IRRG (Infrared, Red, and green) image data and DSM. The selected experimental image is shown in Figure 4. The Potsdam dataset contains 38 patches (of the same size), each consisting of a TOP extracted from a larger TOP mosaic. Each patch contains 4-band IRRGB (infrared, red, green, and blue) image data and DSM. The selected experimental image is shown in Figure 5. Notably, the IRRG images from the Vaihingen dataset and RGB images from the Potsdam dataset are used in this article.

The images contained in this dataset are all large-sized high-resolution satellite remote sensing images, and the depth of CNN has a huge number of parameters, which requires high computer performance. Images like this dataset cannot be directly processed. Cut them into small-sized images first. The high-resolution images in this dataset are cut into 256×256 small-sized images. Starting from the upper left corner of the image, slide cutting with a step size of 256×256 pixels is performed.

The number of weight parameters between deep CNN neurons is huge, which has high requirements for the number of training samples. For this reason, some methods are used to expand the existing samples. The small-sized images obtained by the previous cutting are, respectively, flipped and transformed and rotated 90°, 180°, and 270° clockwise. These methods can greatly expand the data volume of the sample, thereby effectively preventing the occurrence of overfitting. The partially transformed image is shown in Figure 6.

4.2. Parameter Setting and Experimental Platform. The experiment divided the dataset according to the ratio of 5 : 1. There are 75,000 images taken as the training set and 25,000 images taken as the verification set. The learning rate is the parameter that controls the learning speed of the network. In the experiment, the learning rate was set at 0.01 according to the empirical value, at which time the model converges

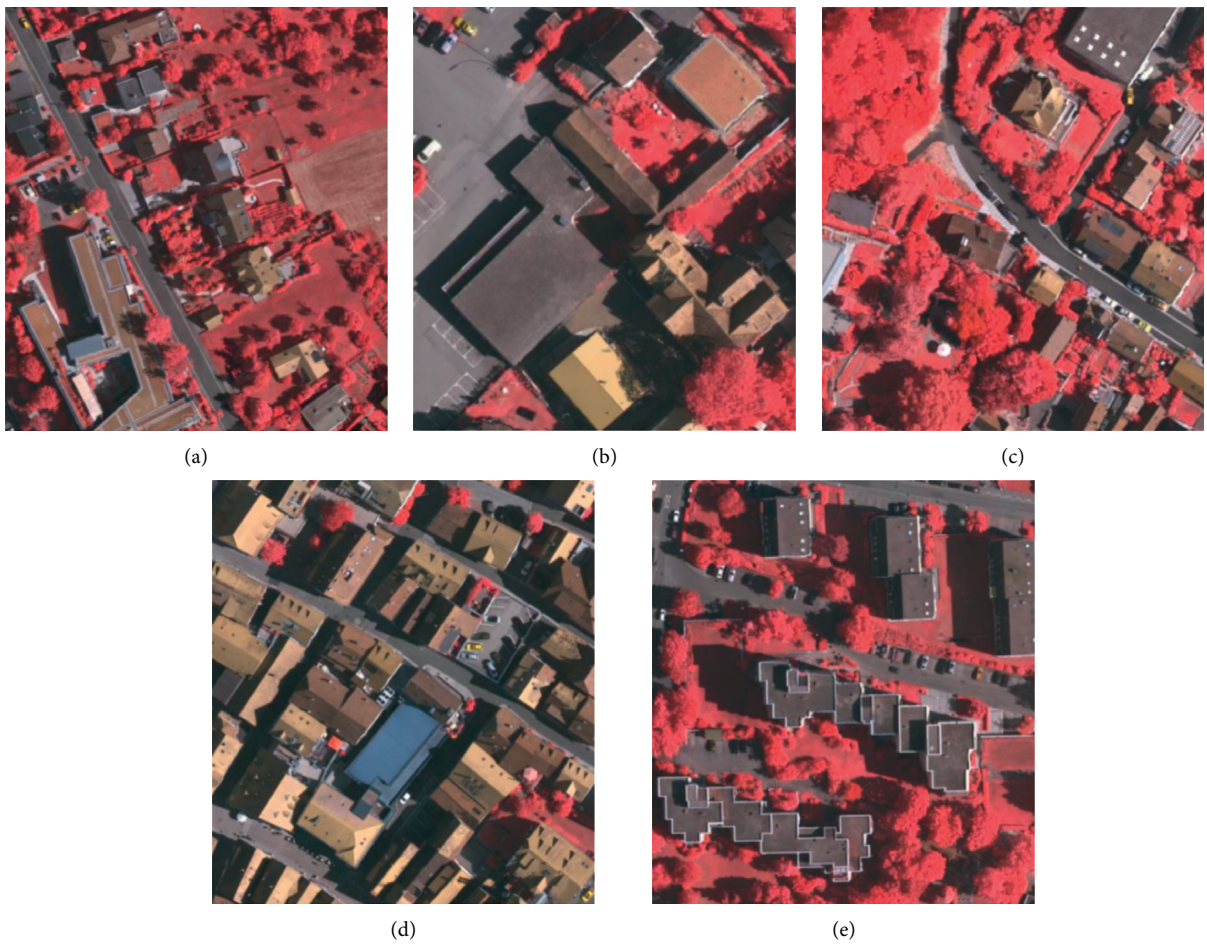

Figure 4: The selected images of the Vaihingen dataset.

slowly and overfits occur. When the improved learning rate is 0.05, the training curve oscillates greatly, and the model cannot reach the optimal value. After several times of reference adjustment, the appropriate learning rate was finally selected as 0.01, the batch scale was set as 10, the number of experimental iterations was set as 30, and the optimizer was set as random gradient descent (SGD). SGD randomly selected a sample for gradient update as follows:

$$W' = W - \alpha \nabla J_t(W), \tag{4}$$

where W' is the updated weight, W is the original weight, α is the learning rate, and $J_t(W)$ is the t-th sample loss.

The number of convolution kernels was set to increase gradually from 64 to 512, which improved the network prediction performance. In the setting of activation function, ReLU and ELU are, respectively, used in this article to train the network and compare its final semantic segmentation effect. In order to ensure the accuracy of comparison verification results, the unified loss function was adopted in all networks as follows:

$$l = -\sum_{j}\sum_{N} y_{N_j} \cdot \lg\left(y^*_{N_j}\right), \tag{5}$$

where $y^*_{N_j}$ indicates the prediction of the category of sample j by the network, while y_{N_j} indicates the true category of sample j.

The experimental environment configuration CPU is Intel(R)Core(TM)i7G 9700K processor, the graphics card is two NVIDA GeForce GTX1080Ti graphics cards, and the total memory capacity is 32 GB. All CNN models running are carried out under the Pytorch framework.

4.3. Evaluation Index. The IoU index is a common model prediction index, which reflects the interaction ratio between the target region and the real source tag. The calculation equation is as follows:

$$IoU = \frac{\sum t_p}{\sum f_n + \sum t_p + \sum f_p}. \tag{6}$$

In the equation, t_p is the actual number of pixels belonging to this class in the predicted results and it is also the number of pixels belonging to this class. f_n is used to predict the number of pixels that belong to the class, but it actually belongs to other classes. f_p is the number of pixels in the predicted results that actually belong to another category.

In general, if the IoU value exceeds 50%, the network is considered to have good image segmentation performance.

Figure 5: The selected images of the Potsdam dataset. The dataset is composed of high-resolution remote sensing images. The surface targets in the image are divided into 6 categories. (a) Low vegetation, including green plants such as grassland, woodland, and agricultural land. (b) Buildings, all types of buildings belong to this category. (c) Impervious surfaces, including road. (d) Cars, all types of cars belong to for this category. (e) Trees, all types of trees belong to this category. (d) Others, it means the clutter background. The semantic information annotation table is shown in Table 1, where *R*, G, and B, respectively, represent red, green, and blue.

Table 1: The label color of 6 categories.

Feature category	Label color (R, G, B)
Low vegetation	(0, 187, 224)
Buildings	(0, 33, 189)
Impervious surfaces	(255, 255, 255)
Cars	(255, 235, 0)
Trees	(0, 196, 0)
Others	(239, 0, 20)

The accuracy rate can represent the accuracy rate of the predicted results, which represents the proportion of the pixels actually belonging to the class predicted in the model. Its calculation equation is as follows:

$$\text{precision} = \frac{\sum t_p}{\sum t_p + \sum f_p}. \tag{7}$$

Precision rate, which is also known as precision rate, is also a common evaluation index in semantic segmentation. In addition, recall rate is also used to evaluate the semantic segmentation effect of remote sensing images in rural areas with different networks. Its calculation equation is as follows:

$$\text{recall} = \frac{\sum t_p}{\sum t_p + \sum f_n}. \tag{8}$$

Recall rate is the ratio of the number of pixels in the class to the actual number of pixels in the class in the predicted results, also known as recall rate. It reflects the classification performance of the model to all positive samples in the dataset.

5. Experimental Results and Analysis

Several classical semantic segmentation networks are used to perform semantic segmentation on the predicted images, and the segmentation effects are compared to mark different

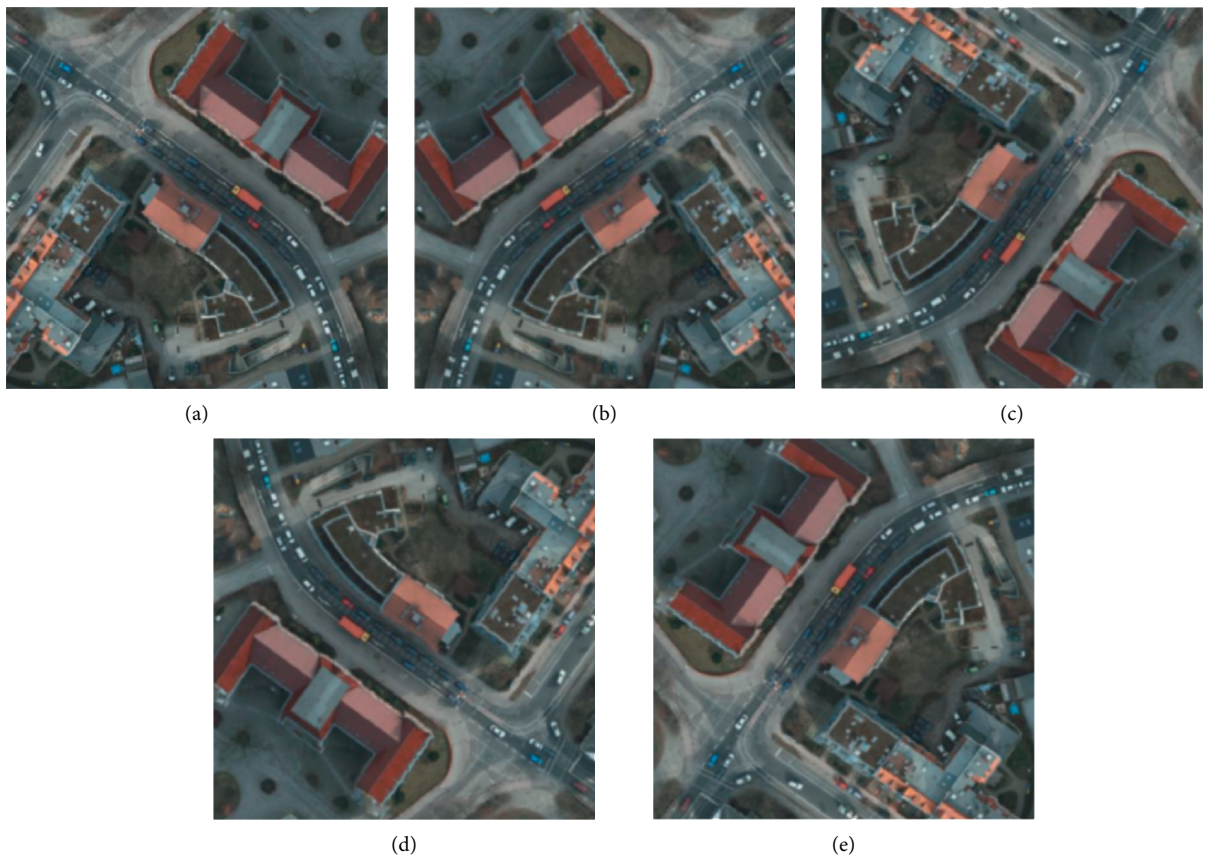

Figure 6: The original image and flipped/rotated images. (a) Image. (b) Flip horizontal. (c) Rotate 90. (d) Rotate 180. (e) Rotate 270.

Table 2: Comparison of training and prediction time of each network.

Methods	Training time (h)	Prediction time (h)
U-Net [12]	15.9	0.46
CASIA3 [35]	20.6	0.71
HUSTW5 [36]	17.1	0.56
Proposed network	**14.6**	**0.43**

colors according to the semantic classification of the data. The experiment compared the training time of U-Net [12], CASIA3 [35], and HUSTW5 [36], respectively, as shown in Table 2.

It can be seen from the training results that the network U-Net has the fastest convergence speed [37, 38], which takes 15.9 hours. CASIA3 training takes 20.6 hours. HUSTW5 training takes 17.1 hours. The proposed network used RefineNet takes 16.6 hours. RefineNet used in this article is further refined on the construct. The mean value and standard deviation of small batch are used to adjust the intermediate output of the neural network so that the value of the intermediate output of each layer of the whole neural network is more stable, and the convergence is accelerated. At the same time, the multiscale feature extraction structure is adopted to ensure the segmentation precision and obtain a relatively better segmentation effect. It is efficient in memory and computation time and easy to train. The activation function of ELU has negative values, which can push the output mean value of the activation unit closer to zero, reduce the offset effect, and then make the gradient close to the natural gradient [39]. Compared with the ReLU method, some recent works have also found that the U-Net method can make the network fit faster [40, 41].

For remote sensing images of the Vaihingen and the Potsdam dataset, the semantic segmentation results obtained by different methods are visualized, as shown in Figures 7 and 8.

According to Figures 7 and 8, the segmentation result of the improved method reduces the wrong segmentation to the minimum, which is the closest to the real value. And the segmentation effect is more intuitive and the overall visual perception is optimal. The specific data of evaluation indexes obtained by each segmentation method are shown in Tables 3 and 4.

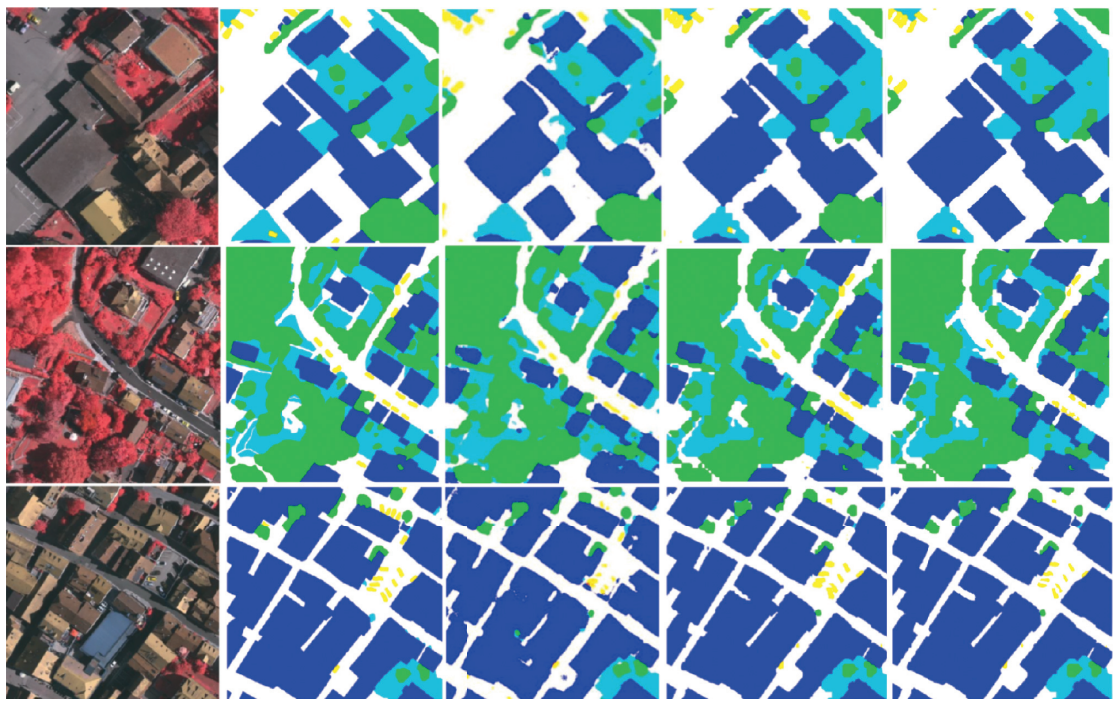

Figure 7: Results on the Vaihingen dataset.

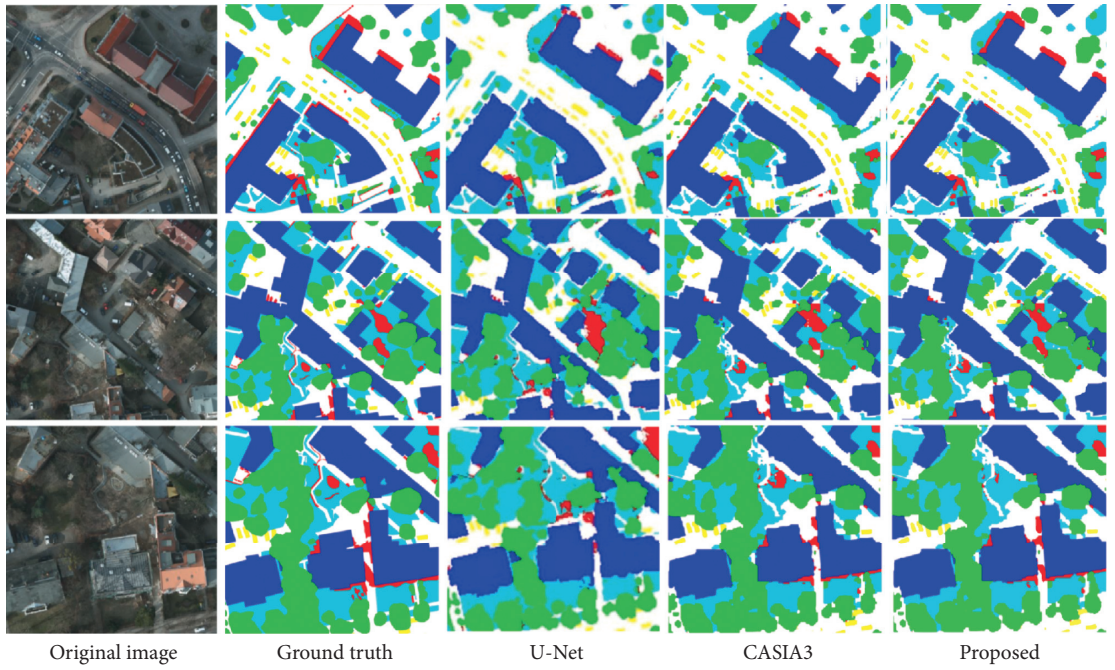

Figure 8: Results on the Potsdam dataset.

It can be seen from Tables 3 and 4 that the proposed method in this article achieves the optimal effect to segment low vegetation, buildings, impervious surfaces, cars, and trees. The activation function ELU has a better effect than ReLU because the negative exponential term in ELU can prevent the emergence of silent neurons and improve the learning efficiency.

Table 3: Evaluation indicators of each method on the Vaihingen dataset (%).

Category	Evaluation	Low vegetation	Buildings	Impervious surfaces	Cars	Trees
U-Net	Precision	69.6	60.2	78.2	70.2	78.8
	Recall	79.7	71.5	79.7	76.2	82.2
	IoU	58.7	62.6	71.7	65.4	68.9
CASIA3	Precision	81.8	71.8	79.3	73.5	81.8
	Recall	76.6	82.6	80.6	85.7	82.6
	IoU	72.2	64.6	69.9	64.8	78.8
HUSTW5	Precision	86.3	74.3	81.5	79.7	70.8
	Recall	81.2	86.9	85.6	86.5	86.6
	IoU	71.2	67.3	79.7	72.5	79.3
Proposed	Precision	**86.9**	**83.4**	**89.3**	**82.7**	**89.9**
	Recall	**87.2**	**87.3**	**87.4**	**88.2**	**91.4**
	IoU	**73.2**	**70.3**	**81.2**	**72.9**	**82.7**

The bold values represent the maximum values of evaluation index in the same column, such as precision, recall, and IoU and the index values of this paper are greater than those of other algorithms.

Table 4: Evaluation indicators of each method on the Potsdam dataset (%).

Category	Evaluation	Low vegetation	Buildings	Impervious surfaces	Cars	Trees
U-Net	Precision	75.3	52.3	73.8	69.4	79.4
	Recall	63.2	78.4	82.3	75.1	80.1
	IoU	52.5	45.7	63.6	56.2	66.2
CASIA3	Precision	83.2	72.5	85.5	75.6	82.5
	Recall	81.7	83.3	85.4	87.2	87.1
	IoU	70.2	63.3	74.6	68.9	78.8
HUSTW5	Precision	85.5	75.0	85.3	78.5	68.4
	Recall	82.3	82.2	86.5	87.6	87.3
	IoU	72.1	66.8	77.6	70.1	80.0
Proposed	Precision	**85.7**	**88.6**	**87.9**	**86.4**	**89.3**
	Recall	**85.5**	**85.9**	**89.1**	**89.4**	**90.2**
	IoU	**74.3**	**68.6**	**79.4**	**72.1**	**82.2**

The bold values represent the maximum values of evaluation index in the same column, such as precision, recall, and IoU and the index values of this paper are greater than those of other algorithms.

6. Conclusion and Future Work

Aiming at the difficulty of semantic segmentation of high-resolution remote sensing images, this article proposed a semantic segmentation method based on CNN and mask generation. The edge extraction method based on the iterative GMM model is used to fuse the multilayer features of CNN through the iterative method. One-layer feature graph is used as feature input in each iteration operation. The manually marked bounding box was used as the initial value of the foreground object contour, and the segmentation mask was modified step by step using the GMM model. At the same time, the framework of residual learning is used to solve the problem of deep network degradation after convergence. The experiment shows that the edge information of the foreground object is extracted by fusing the features of the high, middle, and bottom images. Semantic features are used to reduce semantic level errors of the target contour and the underlying features are used to improve the accuracy of the edge contour. The experiment was performed on the Potsdam and Vaihingen datasets. The results show that the proposed algorithm can effectively improve the overall precision of semantic segmentation of high-resolution remote sensing images and shorten the overall training time and segmentation time.

The future work will tackle optimizing the algorithm, designing the low-resolution image, and also completing the image recognition work.

Data Availability

The labeled datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

The author acknowledges Henan Province Science and Technology Research Project (Project Number: 172102210450) and Xinyang Agriculture and Forestry College Young Teacher Fund Project (2018LG015).

References

[1] N. Audebert, B. Le Saux, and S. Lefèvre, "Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 20–32, 2018.

[2] J. Y. Sun, Z. J. Huang, S. G. Zhou, N. Xu, H. M. Qian, and C. L. Wang, "Building outline vectorization from high spatial resolution imagery," *Journal of Remote Sensing*, vol. 21, no. 3, pp. 396–405, 2017.

[3] K. Zhang, B. Q. Hei, Z. Zhou, and S. Y. Li, "CNN with coefficient of variation-based dimensionality reduction for hyperspectral remote sensing images classification," *Journal of Remote Sensing*, vol. 22, no. 1, pp. 87–96, 2018.

[4] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, Article ID 258619, 12 pages, 2015.

[5] X. D. Li, M. Ye, and T. Li, "Review of object detection based on convolutional neural networks," *Application Research of Computers*, vol. 34, no. 10, pp. 2881–2886, 2017.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, IEEE, Boston, MA, USA, June 2015.

[7] K. Tan, X. Wang, and P. J. Du, "Research progress of the remote sensing classification combining deep learning and semi-supervised learning," *Journal of Image and Graphics*, vol. 24, no. 11, pp. 1823–1841, 2019.

[8] 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 *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 680–688, IEEE, Las Vegas, NV, USA, July 2016.

[9] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Fully convolutional neural networks for remote sensing image classification," in *Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 5071–5074, IEEE, Beijing, China, July 2016.

[10] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1528, IEEE, Santiago, Chile, December 2015.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Munich, Germany, 2015.

[13] R. Li, W. Liu, L. Yang et al., "DeepUNet: a deep fully convolutional network for pixel-level sea-land segmentation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 3954–3962, 2018.

[14] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, https://arxiv.org/pdf/1706.05587.pdf.

[15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Computer Vision—ECCV 2018*, pp. 833–851, Springer, Munich, Germany, 2018.

[16] E. Blanzieri and A. Bryl, "Instance-based spam filtering using svm nearest neighbor classifier," in *Proceedings of the FLAIRS Conference*, pp. 441-442, Key West, FL, USA, May 2007.

[17] S. Kluckner and H. Bischof, "Image-based building classification and 3d modeling with super-pixels," in *Proceedings of the ISPRS technical commission III symposium on photogrammetry computer vision and image analysis*, pp. 233–238, Paris, France, 2010.

[18] B. Chen, F. Qiu, B. Wu, and H. Du, "Image Segmentation Based on Constrained Spectral Variance Difference and Edge Penalty," *Remote Sensing*, vol. 7, no. 5, pp. 5980–6004, 2015.

[19] S. Benchaou, M. B. Nasri, and O. E. Melhaoui, "Feature Selection Based on Evolution Strategy for Character Recognition," *International Journal of Image and Graphics*, vol. 18, no. 3, Article ID 1850014, 2018.

[20] X. Song, Z. Duan, and X. Jiang, "Comparison of artificial neural networks and support vector machine classifiers for land cover classification in Northern China using a SPOT-5 HRG image," *International Journal of Remote Sensing*, vol. 33, no. 10, pp. 3301–3320, 2012.

[21] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2528–2535, IEEE, San Francisco, CA, USA, June 2010.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[23] M. Rajchl, M. C. H. Lee, O. Oktay et al., "DeepCut: object segmentation from bounding box annotations using convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 36, no. 2, pp. 674–683, 2017.

[24] A. Graves, M. Liwicki, S. Fernández et al., "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2008.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, MIT Press, Cambridge, MA, USA, 2012.

[26] T. Nguyen, J. Han, and D. C. Park, "Satellite image classification using convolutional learning," *AIP Conference Proceedings*, vol. 1558, no. 1, pp. 2237–2240, 2013.

[27] 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 Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015.

[28] V. Mnih, *Machine Learning for Aerial Image labeling*, University of Toronto, Toronto, Canada, 2013.

[29] D. Pathak, P. Krähenbühl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1796–1804, IEEE, Santiago, Chile, December 2015.

[30] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: three principles for weakly-supervised image segmentation," in *Computer Vision—ECCV 2016*, pp. 695–711, Springer, Berlin, Germany, 2016.

[31] T. Remez, J. Huang, and M. Brown, "Learning to Segment via Cut-and-Paste," in *Computer Vision—ECCV 2018*, pp. 39–54, Springer, Berlin, Germany, 2018.

[32] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[33] J. Dai, K. He, and J. Sun, "BoxSup: exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pp. 1635–1643, IEEE, Santiago, Chile, December 2015.

[34] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: weakly supervised instance and semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1665–1674, IEEE, Honolulu, HI, USA, July 2017.

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

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

[37] S. Song, J. Liu, Y. Liu et al., "Intelligent object recognition of urban water bodies based on deep learning for multi-source and multi-temporal high spatial resolution remote sensing imagery," *SENSORS*, vol. 20, no. 2, 2020.

[38] W. Zhang, P. Tang, and L. Zhao, "Fast and accurate land-cover classification on medium-resolution remote-sensing images using segmentation models," *International Journal of Remote Sensing*, vol. 42, no. 9, pp. 3277–3301, 2021.

[39] X. Li, F. Xu, X. Lyu et al., "Dual attention deep fusion semantic segmentation networks of large-scale satellite remote-sensing images," *International Journal of Remote Sensing*, vol. 42, no. 9, pp. 3583–3610, 2021.

[40] W. Wei, B. Zhou, D. Połap, and M. Woźniak, "A regional adaptive variational PDE model for computed tomography image reconstruction," *Pattern Recognition*, vol. 92, pp. 64–81, 2019.

[41] K. Qiao, J. Nowak, M. K. Rafal Scherer, and M. Wozniak, "Accurate and fast URL phishing detector: a convolutional neural network approach," *Computer Networks*, vol. 178, no. 4, Article ID 107275, 2020.