electronics

MDPI

Article

Semi-Supervised Remote Sensing Image Semantic Segmentation Method Based on Deep Learning

Linhui Li [1], Wenjun Zhang [1], Xiaoyan Zhang [1], Mahmoud Emam [2] and Weipeng Jing [1,*]

[1] College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China
[2] Faculty of Artificial Intelligence, Menoufia University, Shebin El-Koom 32511, Egypt
* Correspondence: jwp@nefu.edu.cn

Abstract: In this paper, we study the semi-supervised semantic segmentation problem via limited labeled samples and a large number of unlabeled samples. We propose a self-learning semi-supervised approach for the semantic segmentation of high-resolution remote sensing images. Our approach uses two networks (UNet and DeepLabV3) to predict the labels of the same unlabeled sample, and the pseudo labels samples with high prediction consistency are added to the training samples to improve the accuracy of semantic segmentation under the condition of limited labeled samples. Our method expands training data samples by using unlabeled data samples with pseudo labels. In order to verify the effectiveness of the proposed method, some experiments were conducted on the improved ISPRS Vaihingen 2D Semantic Labeling dataset using the method that we proposed. We focus on the extraction of forest and vegetation information and focus on the impact of a large number of unlabeled samples on the precision of semantic segmentation, we combine water, surface, buildings, cars, and background into one category and named others, and we call the changed dataset the improved ISPRS Vaihingen dataset. The experimental results show that the proposed method can effectively improve the semantic segmentation accuracy of high-scoring remote sensing images with limited samples than common deep semi-supervised learning.

Keywords: convolutional neural network; high-resolution remote sensing images; semi-supervised; semantic segmentation

check for updates

Citation: Li, L.; Zhang, W.; Zhang, X.; Emam, M.; Jing, W. Semi-Supervised Remote Sensing Image Semantic Segmentation Method Based on Deep Learning. *Electronics* **2023**, *12*, 348. https://doi.org/10.3390/electronics12020348

Academic Editor: Chiman Kwan

Received: 10 December 2022
Revised: 31 December 2022
Accepted: 4 January 2023
Published: 9 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Semantic segmentation algorithms can be used to assign a label category to each pixel of an image. In remote sensing, the semantic segmentation of remote sensing images is usually called image classification, and it has many applications in this area, which can realize the recognition of different ground objects. This has important application prospects in precision agriculture [1,2], environmental monitoring [3], urban planning [4,5], and other fields. In recent years, semantic segmentation methods have been extensively studied in the field of remote sensing and computer vision. Due to the wide-spread availability of deep learning methods, more algorithms using deep convolutional neural networks (DCNNs) for semantic segmentation have been proposed [6].

The DCNN model extracts features from massive labeled data samples and obtains accurate representations of categories through complex operations. In order to use DCNNs for the semantic segmentation of remote sensing images, a large amount of labeled data samples are first required, and the training process is performed on these labeled data samples. Then, the pre-trained network is used for the semantic segmentation tasks. These two steps are necessary [7]. However, it is difficult and costly to obtain a large amount of labeled data. This is because remote sensing images are different from natural images. Furthermore, large intra-class and small inter-class differences require specialized technicians to accurately label them. Another reason is that with the development of remote sensing image technology, the speed of data acquisition is getting faster, and the amount of

data is getting larger, while the production volume of labeled data is far less than the data volume of remote sensing images. The lack of labeled data samples hinders the application of deep semantic segmentation models in remote sensing image processing to a certain extent. The labeled data samples in semantic segmentation are too small to find suitable DCNN parameters for the network model in pre-training, and the result is overfitting [8].

Aiming at remote sensing semantic segmentation tasks with limited labeled data samples. We propose a self-training semi-supervised semantic segmentation method on remote sensing images. This method uses the probability threshold value predicted by the pseudo label to avoid all pseudo label data being added to the training data sample dataset. Our method can avoid the problem of bad classification decisions. The whole network framework is divided into two parts. One part is to use the UNet network model and DeepLabV3 network model to determine high-reliability pseudo-labels through thresholds. The second part is to use DeepLabV3 to complete self-training semi-supervised semantic segmentation. This method can effectively improve the poor effect of training full convolutional networks under a small amount of label data and indirectly reduces the cost of label drawing.

2. Related Work

2.1. Semantic Segmentation Methods for Remote Sensing Images

For semantic segmentation methods of remote sensing images without the help of deep learning tools, the traditional practice was performed by designing the features manually [9–11] and using many machine learning algorithms to achieve the automated classification of remote sensing images, such as random forest (RF) [12] and support vector machine (SVM) [13]. Among them, RF and SVM can achieve better classification results in a smaller range of classification tasks. SVM is disadvantageous in the face of multiple classifications. The RF classifier can adapt to more complex application scenarios with better generalization, but it easily falls into overfitting because of splitting too many binary tree branches. Traditional machine learning is more sensitive to features, and the classification performance of traditional machine learning methods is efficiently improved through feature analysis to filter out advantageous feature combinations. However, feature engineering often relies on human prior knowledge, and in complex remote sensing classification tasks, the separability of data cannot be guaranteed, while the limited labeled data samples with labels can lead to poor generalization of the model [14].

With the development of deep learning tools, convolutional neural networks (CNNs) were proposed to implement handwritten recognition systems by automatically extracting features from deeper levels of the target instead of extracting features manually [15,16]. Convolutional neural networks are initially applied to the scene classification of remote sensing images, i.e., the input remote sensing image divided into certain dimensions has only one classification result, which is not suitable for the semantic segmentation task, where each pixel has to be attributed to a specific class. In 2015, the classical fully convolutional network (FCN) [17]was proposed to achieve the classification of each pixel category in remote sensing images. After the emergence of the semantic segmentation network, the FCN, SegNet [18], Pyramid Scene Parsing Network (PSPNet) [19], RefineNet [20], U-net++ [21], Deeplab [22–24], and Auto Deeplab [25] network models were successively introduced for semantic segmentation tasks for remote sensing images using methods such as: null convolution and jump-connected layers to improve the classification accuracy.

2.2. Semi-Supervised Learning

Convolutional neural networks can achieve a high classification accuracy due to the sufficient numbers of labeled samples and the great computational power they have, but a large amount of labeled data samples is difficult to obtain in some practical applications. Semi-supervised learning is a decision rule method for supervised learning from a small amount of labeled data samples and a large amount of unlabeled data samples, which can bring higher accuracy without increasing the number of data samples. The general idea of

the semi-supervised method of deep learning is to predict a large amount of unlabeled data samples into pseudo-labeled data, add the pseudo-labeled data samples to the training data samples, prevent the overfitting problem of limited data samples by increasing the amount of data samples, effectively improve the generalization ability of the model, and improve the accuracy of information extraction. Therefore, in the context of artificial intelligence technology, the semi-supervised method has become an effective method for improving classification accuracy when there are limited labeled data samples [26].

A series of studies on semantic segmentation methods based on semi-supervised deep learning methods have been carried out. Most of these semi-supervised learning methods are introduced by using RGB color images, such as: the CIFAR-10, SVHN, Pascal VOC, and CityScapess datasets. Many methods are integrated with semi-supervised learning methods, such as: model compression, migration learning, and collaborative training. Teacher–student models in model compression such as: CutMix-Seg [27], PseudoSeg [28], and CPS [29], where the student model continuously learns the pseudo-label generated by the teacher model, have achieved great success, but there are still more incorrect pseudo-labels misleading the learning of the student model, requiring multiple pseudo-labels for the same image, which is time- and memory-consuming. The idea of migration learning is to use a pre-training neural network with large-scale general data as the basis and then fine-tune it on small-scale data samples in a specific domain to train a neural network model with good performance [24]. Further, tuning performed on a large-scale semantic segmentation dataset (COCO, SUN), followed by semantic segmentation on the target segmentation samples using the strategy of pre-training and then fine-tuning to improve the classification accuracy, mostly using the public dataset for design and tuning, and if the types of test dataset to be segmented are more than the types of the trained dataset, the migration learning approach cannot achieve better segmentation accuracy. The above-mentioned methods are experimentally carried out on RGB images, with few targets included in the public dataset and with fewer specific applications on remote sensing images, which are different from natural images in: the boundaries between targets are not obvious, the size of similar targets varies in a wide range, the variance between classes is small, and the variance within classes is too large. All of these differences make the direct use of practical models on natural images unsuitable for remote sensing images. Some studies have attempted to use remote sensing images as data from generative models, transfer learning, and other networks for efficient feature extraction to seek the problem of training deep learning networks due to the lack of sufficient samples [30]. Meanwhile, other studies have tried to combine the discriminative model FCN with the generative model GAN and [31] constructed a pixel-level remote sensing image road extraction model based on GAN. In [32], the authors proposed a building extraction algorithm for high-resolution and used the principal component analysis (PCA) initialization filter method to complete convolutional neural network initialization with less labeled data to achieve high-accuracy extraction of road information. The above methods have made attempts in the semantic segmentation of remote sensing images, but most of them are based on binary semantic segmentation with a small number of classifications and a large distinction between classes, and they also have the problems of a more complicated training process, large computation, and large memory occupation. Based on the above-mentioned problems, it is necessary to focus on the algorithms and models based on semi-supervised semantic segmentation.

3. Methods

3.1. Forest Vegetation Extraction Method Based on Semi-Supervised Learning

The self-training semi-supervised learning method for information extraction from remote sensing images is usually divided into five steps: The first step involves training a model as $F(x)$ on the labeled data samples and obtaining the network model structure of $F(x)$ by pre-training. The second step uses the network semantic segmentation model $F(x)$ in the first step to predict the unlabeled data samples and obtain the predicted labels of the unlabeled data samples, also called pseudo-labels. The third step uses the labeled

and pseudo-labeled samples as training data samples to train the semi-supervised learning semantic segmentation model $F(x)$. The fourth step validates the obtained network model $F(x)$ on the validation set and evaluates the network performance through evaluation metrics. The fifth step is to predict and evaluate the network performance of the network model on the test data samples. The overall process is shown in Figure 1. The method uses the trained network semantic segmentation model $F(x)$ with labeled data to predict the data without labels, and the limited amount of labeled data samples results in some of the pseudo-labels predicted with $F(x)$ being correct and others being wrong. The wrong labels have to be involved in the subsequent process as training data samples in the semantic segmentation network $F(x)$, which affects the accuracy of the model.

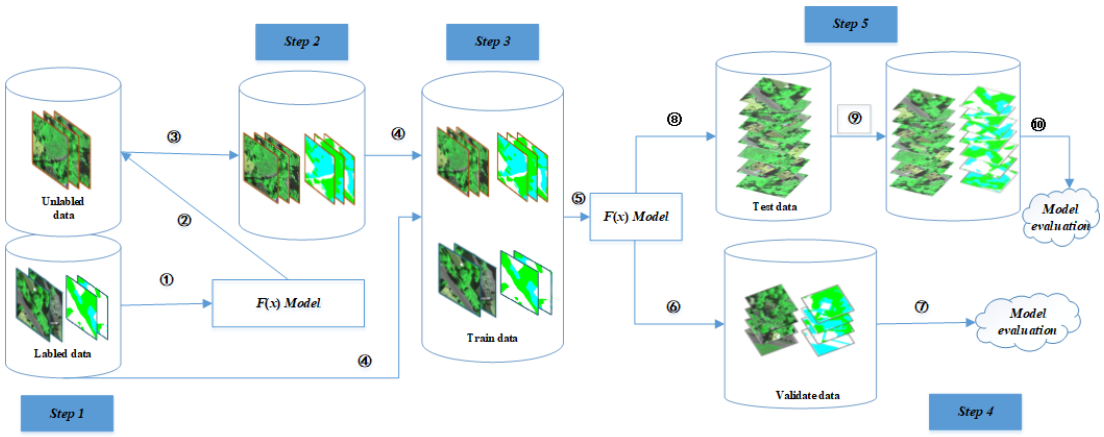


Figure 1. Semi-supervised learning method flow: a self-training semi-supervised learning method for object extraction.

To avoid the participation of incorrectly predicted pseudo-labeled samples in model training, it is crucial to keep the incorrectly predicted pseudo-labels from joining the training set. Inspired by the idea of integrated learning, a semi-supervised learning method based on self-training was designed. This method consists of two network models. The method implementation process is shown in Figure 2. First, two independent networks are trained using labeled datasets, and two network structure models with parameter values are obtained. Then, the two network models predict the unlabeled data set and obtained the pseudo-label data; the pseudo-labels are obtained by predicting the data without labels. The pseudo-label data should be filtered; two models are used to predict the pixel types in the same data. The result is the accumulation of the number of pixels of the same type. If the percentage of pixels of the same types in the total number of pixels was greater than the threshold, which was set as highly reliable pseudo-label data. High-reliability pseudo-labeled data and labeled data were used as the data of the training set, and the network model was retrained. This method can effectively avoid the error propagation problem caused by adding false labels to the training set. Finally, the precision of semantic segmentation is analyzed through the test set.

The threshold setting and model selection are more important in the process of pseudo-label screening. Two methods are used in this study as follows:

Threshold method: The selection of a threshold is to set a threshold. Pseudo-label data that are greater than the set threshold are considered to have high confidence as the training data for retraining. The selection of the threshold is more important.

Common training method: Two different network models are used to predict the same unlabeled data. The network model used is required to be independent, and there is diversity and complementarity between independent network models. If the prediction of the same pixel is the same, the certainty of the pixel category is high. Through joint screening, the confidence of the pseudo-label samples can be improved.

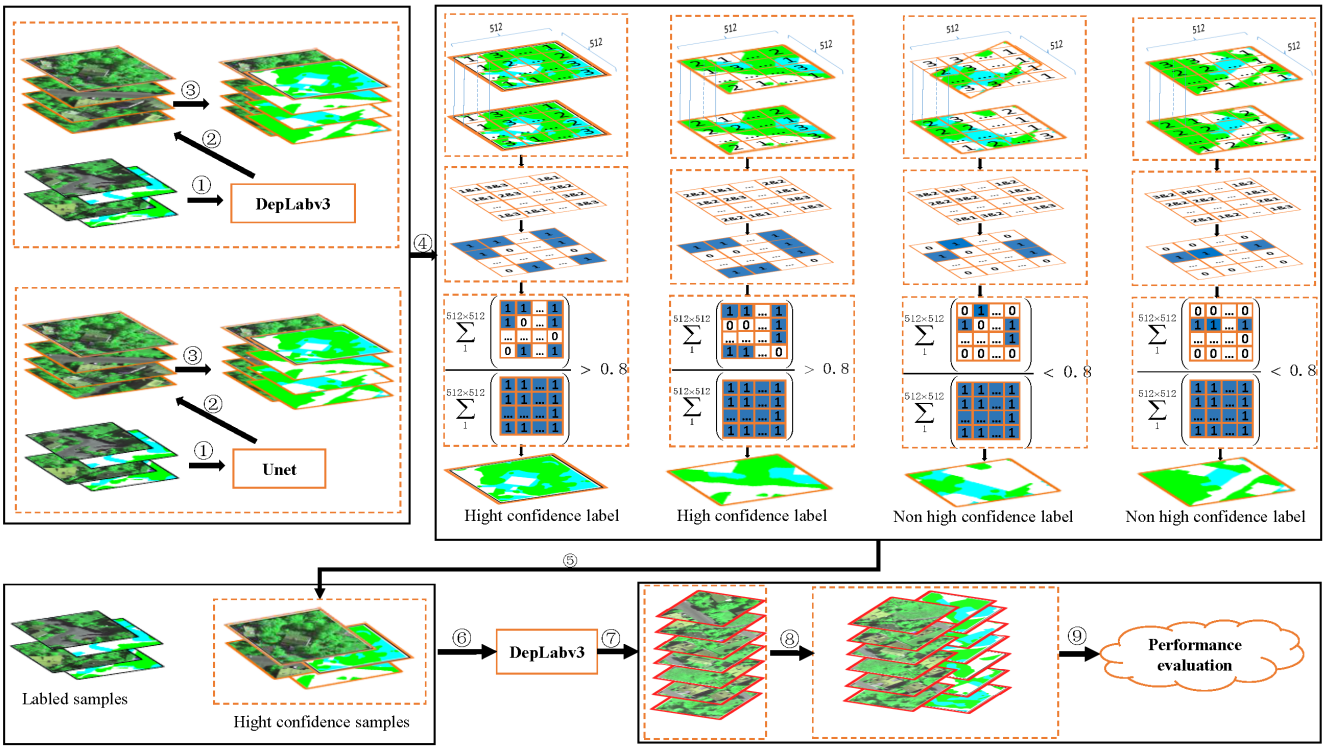


Figure 2. Process of information extraction based on the self-training semi-supervised method.

The pseudo-label samples with high confidence are composed of two stages. In the first stage, we use the common training principle to obtain the pseudo labels under the two network models. Then, we carry out normalization operation, random rotation, and other data reprocessing processes for each 512 × 512 image block and then train the network models UNet and DeepLabv3, respectively, through labeled sample data to obtain model parameters. Moreover, we use two network models to predict labels for unlabeled data to obtain pseudo-label data. In the second stage, different thresholds are used to obtain highly reliable pseudo-label data samples. For the pseudo-label data obtained in the first stage, we set different thresholds and calculate the percentage of the number of pixels with the same prediction for the same segmented image block in each network to the number of pixels in the entire image block. If it is greater than the threshold, the sample is a reliable pseudo-label sample, which can be put into the labeled data samples as the training sample of the training with the DeepLabv3 network. On the contrary, the false tags below the threshold do not participate in the training of the final network model as samples with a low confidence rate.

To facilitate the algorithm description of unlabeled samples with high confidence, the labeled samples and unlabeled samples are mathematically symbolized as follows: The dataset with labels is represented by $D_l = \{\{(X, Y)|(x_1, y_1), \cdots, (x_N, y_N)\}$ where (x_i, y_i) represents the i-th image and labeled pair, and N represents the number of labeled samples used. The dataset without labels is usually denoted by $D_u = \{\{X^{'}|x_1^{'}, \cdots, x_M^{'}\}$, where M represents the number of unlabeled data. The pseudo-label obtained from the unlabeled samples predicted by the network model is denoted by $Y^{'} = \{\{y_1^{'}, \cdots, y_M^{'}\}$, and the pseudo-labeled samples are denoted by $D_w = \{\{(X', Y')|(x_1^{'}, y_1^{'}), \cdots, (x_M^{'}, y_M^{'})\}$ where (x_i, y_i), where $x_i^{'}$ represents the i-th unlabeled sample in $(x_i^{'}, y_i^{'})$, $y_i^{'}$ refers to the pseudo-labeled samples generated by the network model, the number of pixels in each sample is $P = p \times p$, and the number of labeled samples is much smaller than the number of pseudo-labeled samples as $N << M$. The pseudo-labels filtered by the confidence algorithm are called high-confidence pseudo-labels in this paper. Let W be the number

of all predicted pseudo-labeled samples obtained, and the number of pseudo-labels is W, where $W < M$. The confidence Algorithm 1 is described as follows:

Algorithm 1 A high-confidence pseudo-label generation algorithm

Input: Threshold accuracy, integrated model, pseudo-labeled samples $D_{w1} = \{\{(X', Y')|(x_1^{'}, y_1^{'}), \cdots, (x_M^{'}, y_M^{'})\}$, False label sample $D_{w2} = \{\{(X', Y')|(x_1^{'}, y_1^{'}), \cdots, (x_M^{'}, y_M^{'})\}$, of which represents the pseudo-labeled data set generated by the UNet network model, D_{w2} represents the pseudo-labeled data set generated by the DeepLabv3 network model, P represents the number of pixels contained in each sample.

Output: Pseudo-label sample set with high confidence W

Select a sample set of pseudo-labels (like D_w1, D_w2)
for $i = 1$ to M
for $i = 1$ to P
{count=0,input the prediction type for each pixel in x_i with two net models
if prediction type is same in two net models **then**
    count=count+1
    if count/P > Threshold accuracy **then**
        put $(x_1^{'}, y_1^{'})$ into W;
    else return to the upper loop}
else return to the upper loop

3.2. Model Pre-Training

The highly reliable pseudo-labeled data samples are obtained, and the highly reliable pseudo-labeled and labeled data samples are used to train the integrated network. The labeled data samples and high-confidence pseudo-labeled data samples are used as training sets to train the DeepLabv3 convolutional network model. To improve the semantic segmentation accuracy of the network model, pre-training is used to initialize the network model parameters. The number of pre-training rounds is set to 20.

4. Experiment

4.1. The Dataset

The standard ISPRS Vaihingen dataset contains six kinds of ground objects, including impervious water surfaces, buildings, low vegetation, trees, cars, and backgrounds. We focus on the extraction of forest and vegetation information. Therefore, the impervious water surface, buildings, cars, and background are set into one category and named others. The improved ISPRS Vaihingen dataset includes three categories: forest, low vegetation, and others. In the label, forests are represented by green, low vegetation is represented by light blue, and others are represented by white. This dataset contains 33 remote-sensing images of different sizes. Each remote-sensing image is extracted from a larger top-level orthophoto image with a resolution of 9 cm. It consists of three bands: near-infrared, red, and green. The file format is an 8-bit TIFF file. In the improved ISPRS Vaihingen dataset, some remote sensing images and tag data are shown in Figure 3. The display color of remote sensing images is changed to match the visual perception of green forest vegetation. The 33 remote sensing images are cropped into 512 × 512 pixel size sample data, and a total of 459 sample datasets are obtained.

We use 20% of the improved ISPRS Vaihingen dataset as the test set and use 20 samples from the remaining data as the labeled data samples, accounting for 4.4% of the total data. We put them into the integrated network models UNet and DeepLabv3 for the training of network model parameters. The corresponding number of the remaining samples is $459 - 20 = 439$; we take out 347 from 439 as the unlabeled data, accounting for 75.6% of the total data, and the remaining 92 as the test data samples. In order to explore the impact of different counts of labeled data samples on semantic segmentation accuracy, the method of extracting 40, 60, and all labeled data is the same as that of extracting 20 labeled data. The

corresponding labeled samples account for 8.7%, 13.1%, and 17.4% of the total data, and the corresponding numbers of unlabeled data are 399, 379, and 359.

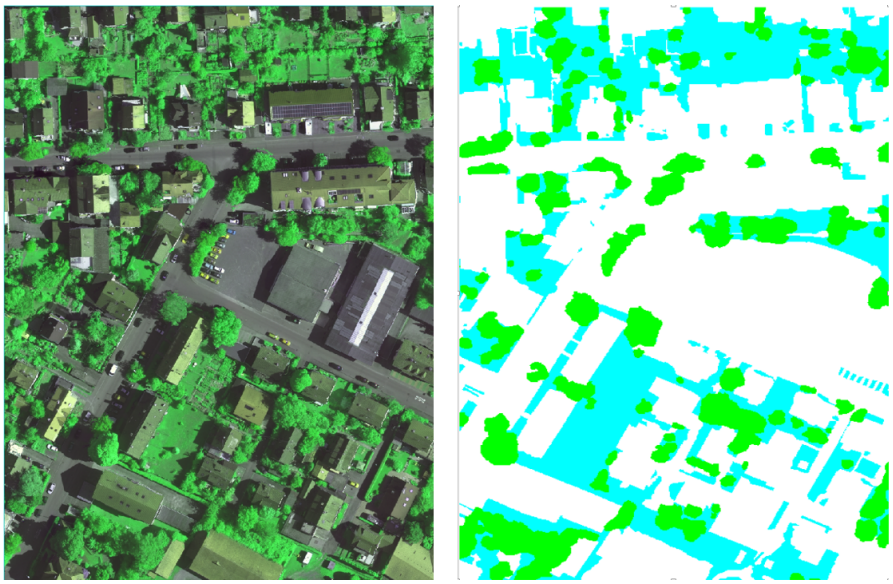


Figure 3. Improved Vaihingen dataset.

In semantic segmentation, most of the evaluation indicators are based on accuracy evaluation, which is often calculated on the basis of a confusion matrix. The main evaluation indicators usually used are: accuracy, recall, precision, balanced F1 score, and average intersection/union ratio (MIoU). In this paper, the F1 score, overall accuracy, and MIoU are used to evaluate the performance of the proposed method. The confusion matrix is shown in Table 1.

Table 1. Confusion matrix.

		True Value	
		Positive	**Negative**
Estimatevalue	Positive	True Positive (TP)	False Positive(FP)
	Negative	False Negative(FN)	True Negative (TN)

The confusion matrix is composed of rows and columns. The diagonal value in the matrix represents that the tag value is the same as the predicted value; it is expressed by P_{ii}. In the matrix, it is true positive (TP) and true negative (TN). The label marked as class i is incorrectly predicted as class j, which is represented by P_{ij}, and it is false positive (FP) and false negative (FN) in the matrix.

4.1.1. Accuracy

Accuracy represents the percentage of correctly classified pixels. The calculation formula is as shown in Equation (1). When multiple classifications are used, then the average of the accuracy of each category is used to obtain the average accuracy. The calculation formulas are as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

4.1.2. Recall

The proportion of true positive samples in the samples with the wrong prediction is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

4.1.3. Precision

The proportion of true positive samples in true positive and false positive samples is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

4.1.4. Balanced F1 Score

$$F1 = 2 \times (\frac{Precision \cdot Recall}{Precision + Recall}) \tag{4}$$

4.1.5. MIoU

The average intersection/union ratio (IoU) is the standard measure for dividing tasks, representing the average intersection/union ratio of two sets. The calculation formula is as follows: where N is the number of categories:

$$IoU = \frac{TP}{TP + FP + FN}, MIoU = \frac{\sum_{i=1}^{N} IoU}{N} \tag{5}$$

4.1.6. Params

The parameter quantity is mainly used to describe the size of the model, which is equivalent to the space complexity of the algorithm. The calculation of parameters of the convolutional layer and the full connection layer are as follows:

$$Params = C_o \times (C_i \times K_w \times K_h + 1) \tag{6}$$

where C_o represents the number of input channels, C_i represents the number of output channels, and $K_w \times K_h$ is the size of the convolution kernel; that is, the parameter quantity of a feature. If the convolution kernel has a bias, then 1 will be added, and if there is no bias, then it will not be added. If the convolution kernel is square, then $K_w = K_h = K$.

The formula for calculating the parameters of the full connection layer is:

$$Params = (I + 1) \times O \tag{7}$$

where I is the dimension of the input vector, which can also be called the size of the input feature map, and O is the dimension of the output vector.

4.1.7. FLOPs

In depth learning, computation is used to describe the length of model execution time, which is similar to the time complexity of the algorithm. The time complexity is divided into the time complexity of the convolution layer, the time complexity of the pooling layer, and the time complexity of the full connection layer. With the time complexity of the convolution layer as the main body, it can be judged by the model execution time. FLOPs of convolutional layer:

$$FLOPs = [C_i \times K_w \times K_h + (C_i \times K_w \times K_{h-1}) + 1] \times W \times H \times C_o \tag{8}$$

where $C_i \times K_w \times K_h$ is the calculation amount of a convolution, $(C_i \times K_w \times K_{h-1})$ is the calculation amount of a convolutional addition operation, 1 is the bias, and W and H are the length and width of the feature map.

FLOPs of pooling layer:

$$FLOPs = H \times W \times C_i \tag{9}$$

where H and W are the height and width of the feature map, and C_i is the number of input channels.

FLOPs of the full connection layer:

$$FLOPs = [I + (I - 1) + 1] \times O = 2 \times I \times O \tag{10}$$

where I is a one-time multiplication, $I - 1$ is a one-time addition, and 1 is the addition of a bias.

4.2. Remote Sensing Image Data Enhancement

In order to obtain better experimental results and accelerate the convergence of neural network training, the remote sensing image dataset is pre-processed by numerical normalization, data enhancement, and random disorder of the training dataset.

4.2.1. Numerical Normalization

Numerical normalization is performed on the data in the training sample, validation, and test sets by using the following formula of the normalization operation.

$$norm = \frac{x_i - min(x)}{max(x) - min(x)} \tag{11}$$

where x_i represents the value of each pixel of sample i, and $min(x)$ and $max(x)$ represent the maximum and minimum values of the pixel, respectively. The data type of the remote sensing image used in the experiment is an 8-bit unsigned integer type, so the upper limit of the value of each pixel is 255. It can be known from the formula that the pixel value of the training sample after the normalization operation ranged from 0 to 1. The purpose of this transformation is to prevent the gradient of the activation function from being too small and to speed up the convergence of the network model.

4.2.2. Data Enhancement

Data augmentation is the process of generating additional data by rotating, transforming, scaling, and randomly intercepting images based on the original data, and data augmentation does not incorporate additional labeled data. Data augmentation enables a limited amount of labeled data to generate more data through changes, improves the number and diversity of training set samples, and avoids overfitting. Additionally, the network model has better strong robustness and generalization ability.

In this paper, the following data enhancement process is carried out independently for training samples of 512 × 512 pixel size within the training set: geometric transformation and color transformation. The training data are flipped horizontally with 50% probability, while the other 50% are flipped vertically. The color transformation is performed on the training samples to increase or decrease the brightness, contrast, saturation, and hue of the color components by the amount of the original values.

4.2.3. Random Order of the Training Set Data

The training set data are obtained by cropping a remote sensing image. Cropped data usually have a certain distribution pattern, and data with a certain pattern repeatedly read by each epoch during the model training learn the order of the sample data, which is not conducive to the training of a robust network model, and the data will be disordered so that they are randomly distributed, which can reflect the commonality of the samples after learning. In the process of random data disordering, the correspondence between data and labels should be ensured; if this correspondence is not ensured, the data and labels will incorrectly correspond during the training process. In this paper, an array is used to establish a one-to-one correspondence between the labels and data, and the cards are

shuffled with the shuffle method to complete the process of random disorder. Before each epoch of training is started, the training dataset undergoes randomly disordered processing to make the selection of training sample data more random.

4.3. Experimental Environment and Parameters

The integrated network models adopted in this paper are DeepLabv3 and UNet. In the training process, SGD is used to optimize the network structure parameters. The initial learning rate is 0.0005, the value of momentum is set to 0.9, and the weight is attenuated. The value of the decal is set to 0.99, and the number of training rounds is set to 60. Due to the small sample data, the early stop strategy is used to prevent the network from overfitting.

5. Discussion

On the improved ISPRS Vaihingen dataset, our network model is trained by using different amounts of labeled data to evaluate its performance. For the convenience of description, the different percentages of labeled data taken out are denoted by A, B, and C, where A represents 4.4% of the labeled sample data, the quantity is 20, B represents 8.7% of the labeled sample data, the quantity is 40, and C represents 13.1% of the labeled sample data, the quantity is 60. Different percentages corresponded to different amounts of labeled data. The total amount of data is 459. The test set has 92 images; they come from $459 \times 20\%$. The 459 subtracted labeled set are the unlabeled set, so the count of the unlabeled sets A, B, and C are 347, 327, and 307, respectively. The integrated network structure proposed in this paper consists of two networks, one is the UNet network model, and the other is the DeepLabv3 network model.

With the data at the same scale, three network model comparison experiments could be conducted, UNet, DeepLabv3, and ours. UNet and DeepLabv3 used semi-supervised methods for forest vegetation information extraction and one for the self-training semi-supervised learning network model proposed in our study. For the convenience of description, UNet_A represents the UNet network model for extracting forest vegetation information using the semi-supervised learning method under the A dataset, DeepLabv3_A represents the DeepLabv3_A network model for extracting forest vegetation information using the semi-supervised method under the A dataset alone, and Ours_A represents the integrated network used for forest vegetation information extraction using the method proposed in this paper under the A dataset. The model performance compared two aspects, on the one hand, comparing different ratios of strongly labeled data volume on semantic segmentation results, and on the other hand, comparing the classification accuracy and other performance metrics of different models at the same ratio.

Figure 4 represents the extraction results using UNet, DeepLabv3, and the integrated network model proposed in this paper with different numbers of labels (based on the prediction results of the test samples). Visually, the extraction of forest vegetation using the network structure model introduced in this paper is better than that of a single network model using the self-learning semi-supervised method. The results as a whole show that the segmentation effect of each network model gradually improved as the amount of strongly labeled data increased. When the amount of strongly labeled data is small, the extraction effect of all network models is not good, and there are a large number of mis-segmented regions. As the training sample data increased, the segmentation errors became less and less, the semantic segmentation effect became better and better from visual observation, and the information extraction effect was the best if the amount of labels reached 60.

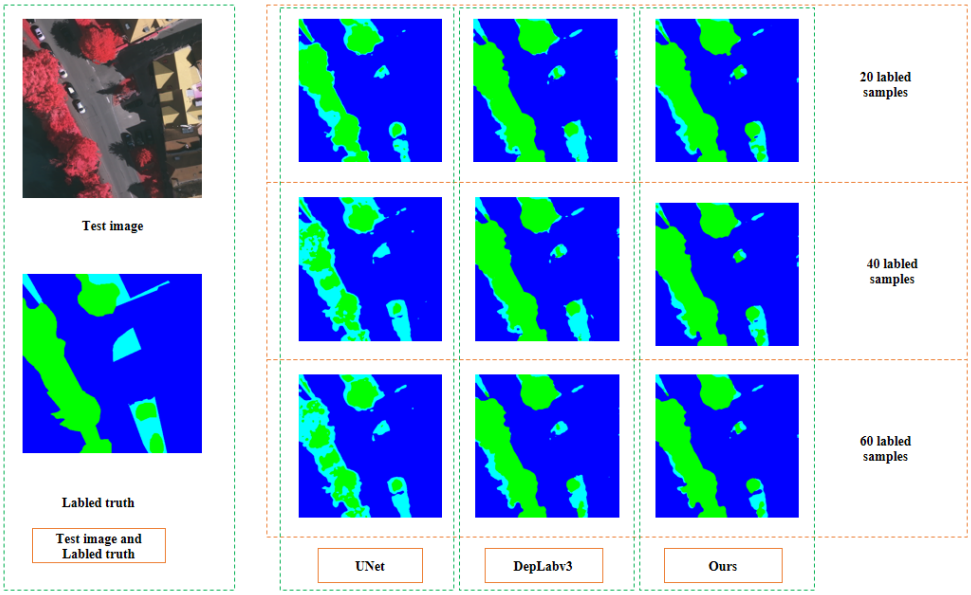


Figure 4. Extraction results of different proportions of labeled forest vegetation information.

Table 2 describes the classification accuracy, overall accuracy, average F1, and MIoU values of the different network models for forest and low vegetation with different numbers of labels. The data shown in Table 2 proves that the proposed method is optimal for the same proportion of labeled samples. This indicates that the proposed method can extract pseudo-labels with high confidence and reliability from the pseudo-labels by the confidence algorithm, prevent the wrong labeled samples from being added to the training set, and improve the accuracy of the information extraction.

Table 2. Experimental results of different numbers of labels in the improved ISPRS Vaihingen dataset.

Number of Labeled Data	Network Model	Grass (%)	Forest (%)	F1 Score (%)	Accuracy (%)	MIoU (%)	Precision (%)	Recall (%)
20	U-Net$_A$	72.12	36.58	68.43	68.40	55.87	69.37	86.40
	DeepLabV3$_A$	80.56	62.46	78.97	79.10	67.00	79.30	78.97
	Ours$_A$	82.14	61.33	79.45	79.81	68.24	80.21	79.46
40	U-Net$_B$	55.82	67.99	72.81	72.62	59.64	75.56	72.81
	DeepLabV3$_B$	81.98	62.06	79.53	79.81	68.19	80.11	79.53
	Ours$_B$	82.85	64.26	80.53	80.67	69.27	80.81	80.53
60	U-Net$_C$	63.71	62.44	73.94	74.38	61.82	76.25	73.94
	DeepLabV3$_C$	83.35	65.19	81.42	82.15	71.25	82.95	81.43
	Ours$_C$	84.29	67.16	82.58	83.46	72.96	84.42	82.59
367	Ours	86.89	70.70	84.24	84.39	73.99	84.64	84.24

The single-class classification accuracy, average F1, overall accuracy, IoU, precision, and recall of the three network models for forest vegetation at different ratios improve with the increase in the number of labeled samples, indicating that by increasing the number of training samples, the network model can learn more semantic features and the network performance is significantly improved. The proposed method is compared by using 20, 40, and 60 labeled samples. Based on the comparison of 20 labeled data, the F1 values of 40 and 60 labeled data increase by 0.03% and 0.02%, respectively. The overall accuracy improved by 0.01% and 0.04%, and MIoU improved by 0.01% and 0.03%. Using all 367 labeled samples to train the network structure for forest vegetation information extraction achieved a difference in the F1 score of approximately 0.05%, a difference in overall classification accuracy of approximately 0.05%, and a difference in MIoU of approximately 0.04%.

In Figure 5, we compare the index values of semantic segmentation under different quantities of labeled data samples. In Figure 5, the (a–c) curves show that the proportion of labeled data samples is 8.7% (20), 13.1% (40), and 17.4 (60), respectively. We use the segmentation network models UNet, DeepLabV3, and Ours. The overall rule is that as the number of labeled data samples increases, the three index values F1 score, accuracy, and MIoU are gradually increasing. The red color curve represents the Ours model, the blue color curve represents the DeepLabV3 model, and the gray color represents the UNet network model. It can be seen from Figure 5a,b that the Ours model is better than the DeepLabV3 model when the number of labels is 40 and 60. The F1 score and accuracy values of the DeepLabV3 model when the number of labels is 20 are slightly higher than the Ours model. The MIoU of the Ours model in Figure 5c is higher than the other two models. The reason is analyzed. The Ours model selects the false labels with high confidence by setting a threshold to avoid the impact of incorrect labels on model decisions. When there are 20 labeled samples, Ours' model F1 score and accuracy are lower than DeepLabV3 because the number of 20 labels is too small. After threshold screening, the remaining high-reliability labels are too few. As a result, UNet and DeepLabV3 use the traditional semi-supervised calculation process. It is predicted that all labels will participate in the training, so the MIoU is low. The indicator values of DeepLabV3 and Ours' model are not much different. This is because we also use DeepLabV3 as the backbone network of our method. We only use the threshold algorithm to select reliable pseudo labels first and use DeepLabV3 with high segmentation accuracy when retraining.

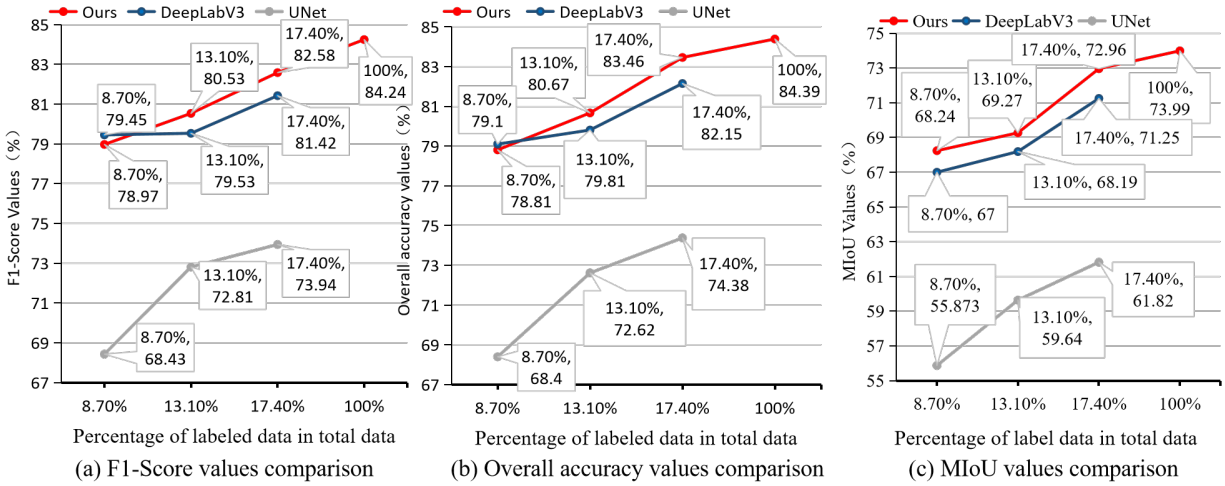


Figure 5. Performance comparison of the labeled data in different percentages.

Table 3 calculates the Params, Flops, and MemoryUsage of the method proposed in this paper and compares it with the traditional semi-supervision of popular backbone networks (UNet, DeepLabV3) without threshold settings. The experimental results are shown in Table 3. The experimental results show that the parameters Flops and Memory Usage of the model used in this paper are slightly larger than those of the DeepLabV3 network model, which are 72.41 M, 265.11 G, and 27.2%, respectively, but their MIoU values are higher than 1.1%. For hardware devices with high accuracy requirements and large video memory, we can consider using the network model proposed in this paper. In essence, the network model proposed in this paper also uses DeepLabV3 with high accuracy as the backbone model. Only the pseudo labels generated by UNet and DeepLabV3 are used for screening, and those exceeding 0.8 are put into the training set. Then, when the model is retrained, the DeepLabV3 network with high accuracy is used to complete pre-training, prediction, and performance comparison. Although the parameters and Flops of the UNet network model are smaller than those of the method proposed in this paper, its MIoU value is less than 9% of that of the method proposed in this paper, and the video memory occupancy is less than 2.4%. For such heavyweight networks as UNet and DeepLabV3, the

method proposed by us has certain advantages from the perspective of accuracy under the condition of large computation and parameter quantities.

Table 3. Parameters and Flops of the model.

Network Framework	Params (M)	Flops (G)	MemoryUsage (%)	MIoU (60 Labeled Data)
UNet	51.38	130.70	8083/32510 = 24.8%	73.94
DeeplabV3	46.72	199.76	8396/32510 = 25.8%	81.43
Ours	72.41	265.11	8860/32510 = 27.2%	82.59

Table 4 compares the time of our method under different quantities of labeled samples. It can be seen from Table 4 that the more labeled data samples there are, the longer it takes to use. Under 20, 40, and 60 labeled data samples, the UNet network model takes 8.3, 9.37, and 12.33 min. The time the DeeplabV3 model takes 10.2, 11.4, and 12.53 min, and the time of our proposed method is 25.28, 25.36, and 27.01 min. By longitudinal comparison, with the same number of labeled data, the time of our proposed method is the longest, followed by DeepLabV3, and the time of UNet is the least. As the number of labeled data samples increases, the time of each network increases slowly, and the time is only 30 min for 367 labeled data samples. Therefore, from the perspective of the execution time of the algorithm, the algorithm proposed in this paper has no advantages. It can be optimized with the network in subsequent research to reduce the running time and improve its time utilization.

Table 4. The comparison of algorithm execution time.

Method	Algorithm Execution Time (Minutes)			
Numbers labeled data (percentage)	20 (8.7%)	40 (13.1%)	60 (17.4%)	367 (78%)
UNet	8.30	9.37	12.33	-
DeeplabV3	10.20	11.40	12.53	-
Ours	25.28	25.36	27.01	30.52

The experiments show that the forest information extraction method based on semi-supervised learning proposed in this paper can achieve information extraction effects similar to those with a large number of labels with a small number of labeled samples. Overall, the proposed method is valuable for studying the semantic segmentation of remote sensing images with limited samples under deep learning. Additionally, the proposed method provides an effective way to carry out the information extraction of remote sensing images under limited sample conditions.

6. Conclusions

We present a simple but effective self-training semi-supervised semantic segmentation approach based on deep learning for remote sensing images. The proposed method imposes the consistency between two network models with different structures. We have shown that a self-training semi-supervised method with two network models can be used to assist the training of end-to-end semantic segmentation frameworks when there are not enough labeled image data. Comparing with traditional semi-supervised semantic segmentation techniques, our proposed method prevented a single net model from overfitting during the training stage. In future work, we will consider the amount of sample information to be measured based on differences while ensuring the accuracy of the unlabeled data samples. We will select high-confidence samples with more effective information to participate in the training of the model and further improve the semantic segmentation accuracy.

Author Contributions: Conceptualization, L.L. and W.J.; methodology, L.L., W.Z., X.Z. and E.M.; software, L.L. and W.Z.; validation, L.L., W.Z. and X.Z.; formal analysis, L.L.; investigation, L.L., W.Z., X.Z. and M.E.; resources, W.J.; data curation, L.L. and W.J.; writing—original draft preparation, L.L.; writing—review and editing, L.L., W.Z., X.Z. and E.M.; visualization, L.L.; supervision, W.J.; funding acquisition, W.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (32171777, 32271865), Science and Technology Program of Inner Mongolia Autonomous Region (2022YFSJ0037).

Data Availability Statement: We utilized a public 2D semantic labeling dataset, Vaihingen, provided by the International Society for Photogrammetry and Remote Sensing (ISPRS). The Vaihingen dataset is freely available at https://www.isprs.org/ accessed on 10 November 2022.

Acknowledgments: The authors want to thank the International Society for Photogrammetry and Remote Sensing for providing the public datasets used as the experimental data in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shanmugapriya, P.; Rathika, S.; Ramesh, T.; Janaki, P. Applications of remote sensing in agriculture—A Review. *Int. J. Curr. Microbiol. Appl. Sci.* **2019**, *8*, 2270–2283. [CrossRef]
2. Weiss, M.; Jacob, F.; Duveiller, G. Remote sensing for agricultural applications: A meta-review. *Remote Sens. Environ.* **2020**, *236*, 111402. [CrossRef]
3. Zhang, C.; Liu, C.; Wang, Y.; Si, F.; Zhou, H.; Zhao, M.; Su, W.; Zhang, W.; Chan, K.L.; Liu, X. Preflight Evaluation of the Performance of the Chinese Environmental Trace Gas Monitoring Instrument (EMI) by Spectral Analyses of Nitrogen Dioxide. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3323–3332. [CrossRef]
4. Rogge, D.; Rivard, B.; Segl, K.; Grant, B.; Feng, J. Mapping of NiCu–PGE ore hosting ultramafic rocks using airborne and simulated EnMAP hyperspectral imagery, Nunavik, Canada. *Remote Sens. Environ.* **2014**, *152*, 302–317. [CrossRef]
5. Xiao, Y.; Zhan, Q. A review of remote sensing applications in urban planning and management in China. In Proceedings of the 2009 Joint Urban Remote Sensing Event, Shanghai, China, 20-22 May 2009; pp. 1–5.
6. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Rodríguez, J.G. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857. Available online: https://arxiv.org/abs/1704.06857 (accessed on 10 November 2022).
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
8. Philipp, G.; Carbonell, J.G. The Nonlinearity Coefficient—Predicting Overfitting in Deep Neural Networks. *arXiv* **2018**, arXiv:1806.00179.
9. Bruzzone, L.; Chi, M.; Marconcini, M. A novel transductive SVM for semisupervised classification of remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 3363–3373. [CrossRef]
10. Sheykhmousa, M.; Mahdianpari, M.; Ghanbari, H.; Mohammadimanesh, F.; Ghamisi, P.; Homayouni, S. Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 6308–6325. [CrossRef]
11. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [CrossRef]
12. Vapnik, V. The support vector method of function estimation. In *Nonlinear Modeling*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 55–85.
13. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
14. Hong, D.; Yokoya, N.; Ge, N.; Chanussot, J.; Zhu, X.X. Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification. *ISPRS J. Photogramm. Remote Sens.* **2019**, *147*, 193–205. [CrossRef]
15. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
16. Zhong, L.; Hu, L.; Zhou, H. Deep learning based multi-temporal crop classification. *Remote Sens. Environ.* **2019**, *221*, 430–443. [CrossRef]
17. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the CVPR, Boston, MA, USA, 7–12 June 2015.
18. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
19. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. *arXiv* **2017**, arxiv:1612.01105.
20. Lin, G.; Milan, A.; Shen, C.; Reid, I. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. *arXiv* **2016**, arXiv:1611.06612.

21. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–11.
22. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
23. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef]
24. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv* **2018**, arXiv:1802.02611.
25. Liu, C.; Chen, L.C.; Schroff, F.; Adam, H.; Hua, W.; Yuille, A.; Fei-Fei, L. Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation. *arXiv* **2019**, arXiv:1901.02985.
26. Zhu, X.; Goldberg, A.B. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2009**, *3*, 1–130.
27. French, G.; Laine, S.; Aila, T.; Mackiewicz, M.; Finlayson, G. Semi-supervised semantic segmentation needs strong, varied perturbations. *arXiv* **2019**, arXiv:1906.01916.
28. Zou, Y.; Zhang, Z.; Zhang, H.; Li, C.L.; Bian, X.; Huang, J.B.; Pfister, T. Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv* **2020**, arXiv:2010.09713.
29. Chen, X.; Yuan, Y.; Zeng, G.; Wang, J. Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision. *arXiv* **2021**, arXiv:2106.01226.
30. Tan, K.; Wang, X.; Du, P. Research progress of the remote sensing classification combining deep learning and semi-supervised learning. *J. Image Graph.* **2019**, *24*, 1823–1841.
31. Shi, Q.; Liu, X.; Li, X. Road detection from remote sensing images by generative adversarial networks. *IEEE Access* **2017**, *6*, 25486–25494. [CrossRef]
32. Rongshuang, F.; Yang, C.; Qiheng, X.; Jingxue, W. A high-resolution remote sensing image building extraction method based on deep learning. *Acta Geod. Cartogr. Sin.* **2019**, *48*, 34.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.