remote sensing

MDPI

Article

Semi-Supervised Adversarial Semantic Segmentation Network Using Transformer and Multiscale Convolution for High-Resolution Remote Sensing Imagery

Yalan Zheng [1,2,3,4], Mengyuan Yang [1,2,3,4], Min Wang [1,2,3,4,*], Xiaojun Qian [5], Rui Yang [1,2,3,4], Xin Zhang [6] and Wen Dong [6]

[1] Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing 210023, China; 181301026@njnu.edu.cn (Y.Z.); ymy2020@njnu.edu.cn (M.Y.); 211301022@njnu.edu.cn (R.Y.)
[2] School of Geography, Nanjing Normal University, Nanjing 210023, China
[3] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China
[4] State Key Laboratory Cultivation Base of Geographical Environment Evolution (Jiangsu Province), Nanjing 210023, China
[5] School of Artificial Intelligence, Nanjing Normal University, Nanjing 210097, China; 05160@njnu.edu.cn
[6] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100101, China; zhangxin@radi.ac.cn (X.Z.); dongwen01@radi.ac.cn (W.D.)
* Correspondence: sysj0918@njnu.edu.cn

check for updates

Citation: Zheng, Y.; Yang, M.; Wang, M.; Qian, X.; Yang, R.; Zhang, X.; Dong, W. Semi-Supervised Adversarial Semantic Segmentation Network Using Transformer and Multiscale Convolution for High-Resolution Remote Sensing Imagery. *Remote Sens.* **2022**, *14*, 1786. https://doi.org/10.3390/rs14081786

Academic Editor: KWONG Tak Wu Sam

Received: 2 March 2022
Accepted: 6 April 2022
Published: 7 April 2022

Abstract: Semantic segmentation is a crucial approach for remote sensing interpretation. High-precision semantic segmentation results are obtained at the cost of manually collecting massive pixelwise annotations. Remote sensing imagery contains complex and variable ground objects and obtaining abundant manual annotations is expensive and arduous. The semi-supervised learning (SSL) strategy can enhance the generalization capability of a model with a small number of labeled samples. In this study, a novel semi-supervised adversarial semantic segmentation network is developed for remote sensing information extraction. A multiscale input convolution module (MICM) is designed to extract sufficient local features, while a Transformer module (TM) is applied for long-range dependency modeling. These modules are integrated to construct a segmentation network with a double-branch encoder. Additionally, a double-branch discriminator network with different convolution kernel sizes is proposed. The segmentation network and discriminator network are jointly trained under the semi-supervised adversarial learning (SSAL) framework to improve its segmentation accuracy in cases with small amounts of labeled data. Taking building extraction as a case study, experiments on three datasets with different resolutions are conducted to validate the proposed network. Semi-supervised semantic segmentation models, in which DeepLabv2, the pyramid scene parsing network (PSPNet), UNet and TransUNet are taken as backbone networks, are utilized for performance comparisons. The results suggest that the approach effectively improves the accuracy of semantic segmentation. The F1 and mean intersection over union (mIoU) accuracy measures are improved by 0.82–11.83% and 0.74–7.5%, respectively, over those of other methods.

Keywords: semantic segmentation; semi-supervised learning; transformer; adversarial learning; remote sensing; building extraction

1. Introduction

Massive quantities of high-resolution remote sensing data are collected every day, along with the progress of sensor technology, which creates great challenges to fast and accurate remote sensing imagery information acquisition. Recently, convolutional neural networks (CNNs) have realized excellent presentation on remote sensing imagery interpretation, with their powerful feature representation capability [1,2]. Semantic segmentation techniques represented by fully convolutional networks (FCNs) [3] can achieve accurate pixelwise image classification with sufficient training data, which has become the mainstream technology in the information extraction field and is widely used for remote sensing imagery object extraction, including buildings, roads, and water bodies [4–6].

Classical semantic segmentation networks, such as the pyramid scene parsing network (PSPNet) [7], DeepLabs [8] and dual attention network (DANet) [9], are trained in a fully supervised mode, which relies on massive manual annotations. Remote sensing imagery is characterized by multisource, multitemporal and complex scenes and acquiring adequate pixelwise annotations is extremely expensive. Although some datasets have been established for remote sensing semantic segmentation, such as the Gaofen Image Dataset (GID) [10], the EVLab-Semantic Segmentation (EVLab-SS) Dataset [11], and the International Society for Photogrammetry and Remote Sensing (ISPRS) Potsdam datasets [12], the quantity of training data for semantic segmentation is still small, considering the complexity of remote sensing information extraction tasks. The existing datasets have difficulty in covering different regions and image types simultaneously, which seriously affects the generalization capability of models. Therefore, many existing approaches rely on semi-supervised training schemes to reduce annotation requirements [13,14]. Research on using unlabeled samples to assist model training and improving the accuracy of object extraction with a small quantity of annotated data, namely, semi-supervised learning (SSL) strategies, is of great significance.

SSL can automatically utilize unlabeled samples to enhance the generalization ability of learners, without interacting with the outside world. End-to-end semi-supervised deep learning methods include proxy-label methods [15,16], consistency regularization [17,18], hybrid methods [19,20], and SSL methods combined with generative adversarial networks (GANs) [21]. GAN-based SSL methods, namely semi-supervised adversarial learning (SSAL) techniques, have become popular in recent years and have been applied for remote sensing tasks, involving image segmentation and image interpretation [22,23]. Figure 1 shows a typical SSAL framework for image semantic segmentation [24]. The generator in an initial GAN framework [25] is replaced by a segmentation network, which inputs labeled and unlabeled data and outputs the corresponding prediction maps. The discriminator network inputs the prediction maps and ground-truth maps and outputs confidence maps, which are taken as supervisory signals for the unlabeled data to guide the SSL process. Some studies [24] have shown that this framework enables segmentation networks to learn higher-order structural information without postprocessing, thereby improving the generalization ability of the networks.

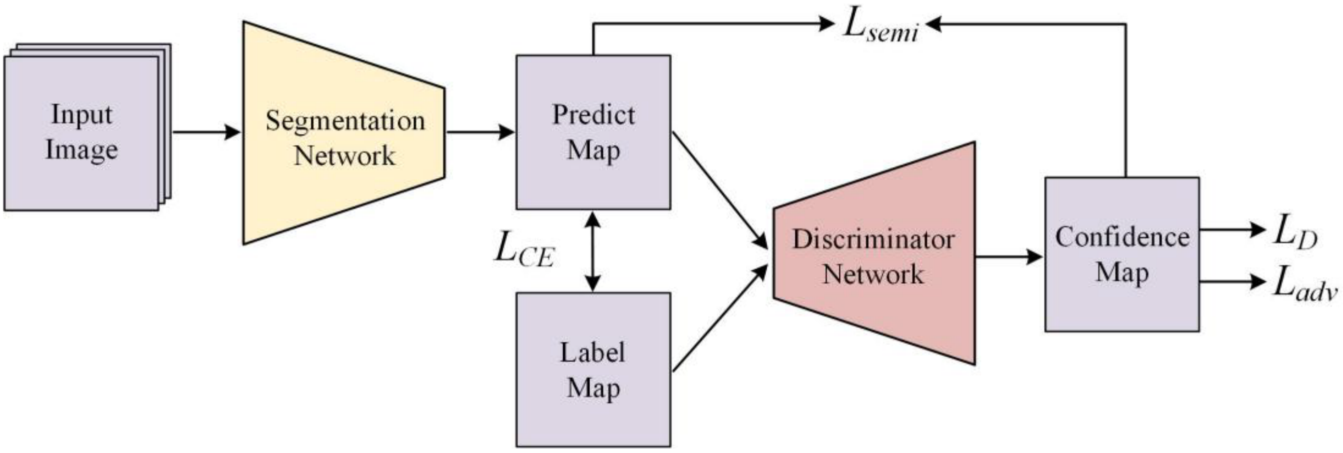

Figure 1. A typical SSAL framework, where L_{CE}, L_D, L_{adv} and L_{semi} respectively represent cross-entropy loss, discriminator loss, adversarial loss, and semi-supervised loss, respectively.

FCNs are commonly used to construct segmentation networks and discriminator networks under the SSAL framework. FCNs have powerful feature extraction capabilities.

However, restricted by the given receptive fields, convolution operations have difficulty acquiring global contextual information [26]. To overcome this limitation, some multiscale modules [7,8] have been proposed to improve the feature extraction capability of the resulting models. In addition, utilizing deep networks with complex components [27] and integrating attention modules into FCN architectures, such as DANet [9] and the squeeze-and-excitation network (SENet) [28], can provide effective global context. However, these approaches cannot avoid the loss of details when the resolutions of feature maps are gradually reduced during the encoding phase.

The Transformer first appeared in machine translation tasks and has recently raised much concern in the computer vision field [29–32]. Transformer layers [33], which contain stacked multi-head self-attention (MSA) and multilayer perceptron (MLP) blocks, can capture global contextual information and the long-range dependencies between objects. In complex remote sensing scenes, acquiring contextual long-range dependencies is important for accurate object recognition and extraction. Methods combining convolutions with a Transformer can acquire both the local feature and the global contextual relationship simultaneously. Some works have shown that this combination effectively improves image segmentation accuracy [26,34]. However, such studies are rare in semi-supervised remote sensing image segmentation.

In this article, we develop a novel semi-supervised adversarial semantic segmentation approach for remote sensing information extraction that combines the advantages of both convolution and Transformer, called TRANet. The main contributions include the following:

- A multiscale input convolution module (MICM) and an improved strip-max pooling (SMP) structure are provided. The MICM adopts multiscale downsampling and skip connections to capture information of different input scales, while maintaining the spatial details of objects in complex remote sensing scenes. The SMP preserves both the global and horizontal/vertical information during feature extraction, thereby reducing the information loss when the resolutions of the feature maps are gradually reduced.
- TRANet is developed with two subnetworks. The segmentation network is characterized by a double-branch encoder, which integrates the Transformer module (TM) and the MICM. The discriminator network is designed by using a parallel convolution architecture with different kernel sizes. Two subnetworks are trained under the SSAL framework. TRANet can extract local features and long-range contextual information simultaneously and improve generalization capability with the assistance of unlabeled data.
- Taking building extraction as a case study, experiments on the WHU Building Dataset (WBD) [35], Massachusetts Building Dataset (MBD) [36] and GID [10] are carried out to validate TRANet. DeepLabv2, PSPNet, UNet and TransUNet are used as segmentation networks for a performance comparison under the same SSAL scheme. The results demonstrate that TRANet improves segmentation accuracy compared to other approaches when only a few labeled samples are available.

The remainder of this article is arranged as follows. Section 2 introduces some related works. The design of the proposed approach is detailed in Section 3. The experimental setup and results are illustrated in Section 4. Section 5 discusses ablation experiments and parameter selections. Section 6 summarizes this article.

2. Related Work

2.1. Semi-Supervised Semantic Segmentation

Many existing methods rely on the SSL scheme to reduce the workload of manual annotation [37,38]. Currently, end-to-end SSL methods can be roughly divided into four categories (1) Proxy-label methods. Such methods use trained models with labeled data to produce pseudo-labels for unlabeled data; examples include pseudo-label [15] and co-training [16]. Their training depends on experience. (2) Consistency regularization. These approaches assume that if noise is applied to samples, the predictions for noisy and non-noisy samples should be as consistent as possible, such as the temporal ensembling [17] and mean teacher methods [18]. They require high robustness to perturbations to achieve improved generalization ability. (3) Hybrid methods. These techniques, such as MixMatch [19] and FixMatch [20], integrate the aforementioned two SSL methods into one framework and have complex model structures. (4) SSL methods combined with GANs [21]. Such methods use the discriminator to facilitate the training of the generator, thereby improving the performance of the resulting models.

SSL methods combined with GANs have been widely applied in semantic segmentation tasks and have achieved good performance. Souly et al. [39] used a GAN generator to create pseudosamples and used a discriminator to classify the pixels into different semantic categories. Four datasets were used to verify the developed method. Hung et al. [24] replaced the generator in a GAN framework with the DeepLabv2 model and designed a fully convolutional discriminator. They utilized the confidence maps generated by the discriminator as the supervisory signals for the unlabeled data to improve the segmentation accuracy under adversarial training. Zhang et al. [40] utilized a segmentation network with two self-attention modules to learn the spatial semantic relationship. They simultaneously used a discriminator containing spectral normalization to improve the training performance. Sun et al. [41] designed a segmentation network with a channel-weighted multiscale feature module and a discriminator network integrating a boundary attention module and residual blocks. Their method alleviated the boundary blur of objects and obtained improved segmentation accuracy on remote sensing datasets.

2.2. Convolution Neural Network and Variants

FCN-based architectures are used to construct both the segmentation and discriminator networks in the classical semi-supervised adversarial semantic segmentation framework. CNN is a hierarchical data representation method that gradually abstracts features with rich semantic information from shallow to deep. FCNs [3], which are extended on the basis of CNNs, contain encoder-decoder structures and replace the fully connected layers of CNNs with convolution layers for image segmentation. FCNs can automatically obtain precise local features and abstract high-level features via end-to-end training, and they have strong feature representation ability for specific tasks.

Deep learning-based semantic segmentation networks are mostly implemented with FCNs. However, restricted by the receptive fields, the features captured by the convolution layers fail to effectively learn long-range dependency information. To overcome this limitation, multiscale modules, such as the atrous convolution module [7] and spatial pyramid pooling [8], use convolution or pooling operations with different scales to obtain features with different receptive fields, thereby enhancing the feature representation ability of the resulting model. In addition, simply increasing the depths of networks [27], acquiring multiscale image characteristics, and integrating attention modules into FCN architectures can provide effective global context. For instance, Luo et al. [42] utilized two uniform residual networks with five levels in the encoder to process input images and auxiliary feature data. They also added the channel attention mechanism into the decoder for remote sensing image feature selection. Huang et al. [43] used a channel-wise attention mechanism to refine coarse labels of different scales and fused features of different levels via an attention-based module. Their method reduced the feature differences and improved the segmentation accuracy in remote sensing datasets. However, the attention modules are usually placed at the top of the employed convolution architecture, which restricts attention learning to high-level features. Such strategies still cannot prevent the loss of details when the resolutions of feature maps are gradually reduced.

2.3. Transformer

The vision Transformer (ViT) [29] was the first work to apply a pure Transformer with self-attention to image classification. ViT divides the input image into a series of image patches for sequence-to-sequence prediction and has achieved state-of-the-art performance on the ImageNet dataset. Context modeling is extremely important for semantic segmentation. The Transformer can capture global contextual information via self-attention, which compensates for the deficiency of convolution operations. Therefore, some scholars have studied combining Transformers with CNNs to improve semantic segmentation accuracy. Zheng et al. [26] proposed a segmentation model with a Transformer-alone encoder, which replaced the stacked convolution layers with a pure Transformer to extract features and combined it with a convolution-based decoder for image segmentation. Chen et al. [44] inserted a Transformer into the top of the encoder in UNet to extract global information and then upsampled the features by a convolution-based decoder to obtain precise segmentation results. However, the aforementioned methods are applied to natural scenes and medical images in a fully supervised training mode. Few studies have used the Transformer to segment high-resolution remote sensing images containing complex objects. Furthermore, few studies have focused on constructing semi-supervised segmentation networks by using Transformers.

The proposed TRANet is mainly characterized by its double-branch encoder segmentation network. The unique MICM enables the network to acquire features of different input scales and maintain spatial information. Furthermore, the long-range modeling advantages of the Transformer compensate for the deficiency regarding the limited receptive fields of convolution operations. Relying on the SSAL framework, TRANet uses the confidence map generated by the unique double-branch discriminator network to guide the training of unlabeled data and further refines the segmentation network, thereby achieving increased image segmentation accuracy.

3. Methodology

3.1. Algorithm Overview

The semi-supervised adversarial semantic segmentation task is expressed as follows. Given $(m + n)$ images with sizes of $H \times W \times C$ and corresponding labels as inputs:

$$\begin{aligned} X &= \{x_{l1}, x_{l2}, \cdots, x_{lm}; x_{u1}, x_{u2}, \cdots, x_{un}\} \\ Y &= \{y_{l1}, y_{l2}, \cdots, y_{lm}\}, \end{aligned} \tag{1}$$

where x_{lm} and x_{un} denote m labeled images x_l and n unlabeled images x_u, respectively. Generally, $n \gg m$; that is, unlabeled data are far more abundant than labeled data. y_{lm} is the binary label map corresponding to x_{lm}, which contains a target value of 1 and a background value of 0. The segmentation network generates prediction maps by training with the labeled and unlabeled data. The discriminator network distinguishes the approximation degree between segmented results and sample labels and optimizes the segmentation model during adversarial training.

Figure 2 illustrates the TRANet graphically. The segmentation network comprises a classical encoder-decoder structure, and the discriminator network includes double-branch convolution structures with different kernel sizes. The two networks are combined for image segmentation under the SSAL framework (Figure 1).

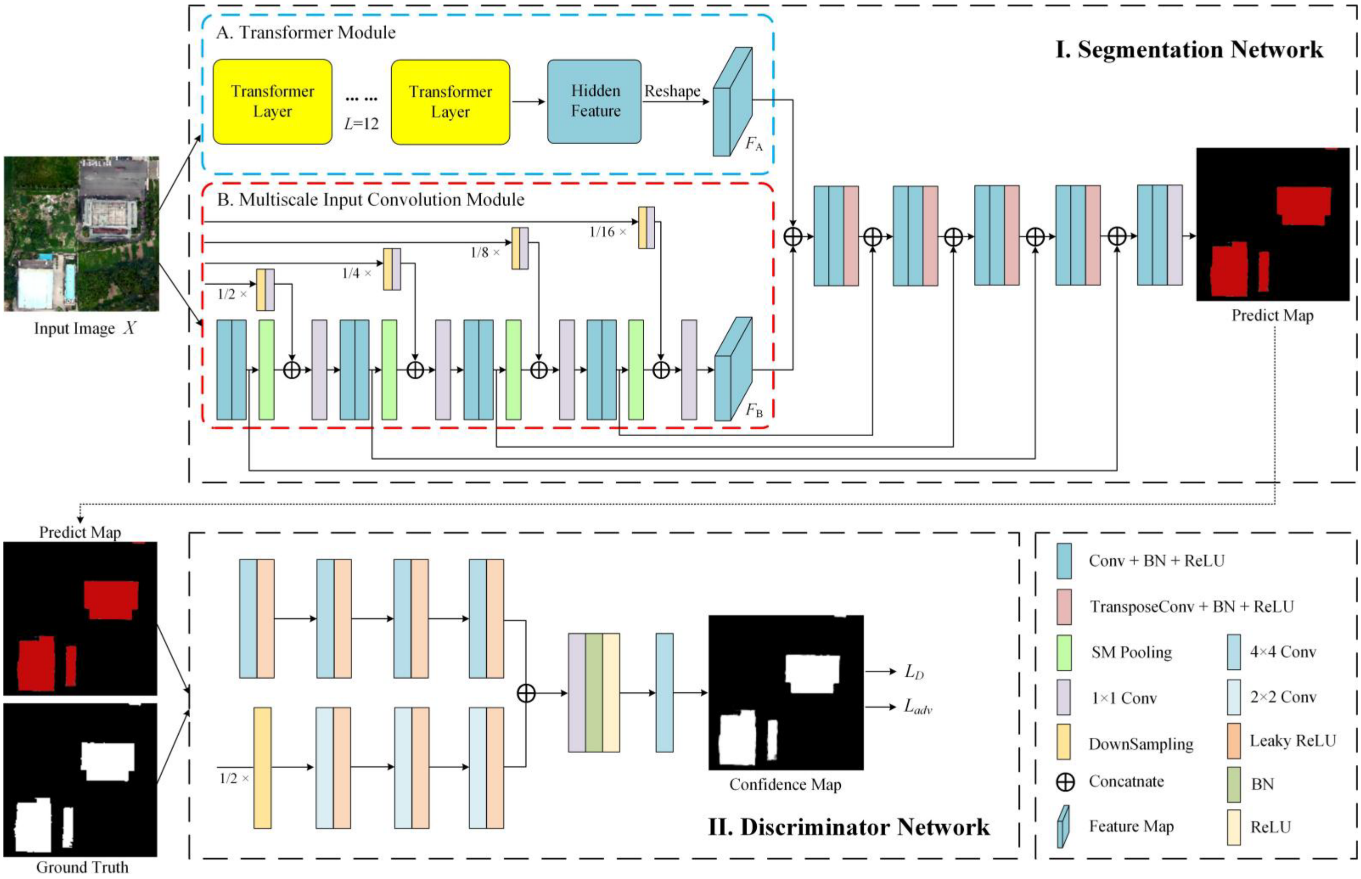

Figure 2. Architecture of TRANet.

3.2. Segmentation Network

As shown in part I of Figure 2, the encoder of the segmentation network contains a TM and an MICM. The TM acquires the global contextual features F_A by self-attention. The MICM obtains the spatial information of multiscale input images and extracts local features F_B through convolution and pooling operations. The joint feature F is obtained by Equation (2):

$$F = F_A \oplus F_B, \tag{2}$$

where $\oplus$ denotes the feature concatenation operation.

3.2.1. Transformer Module

The TM serializes the input images and captures global contextual information by using self-attention, which maintains the complete object features and alleviates the detail loss while gradually reducing the resolutions of the feature maps. The standard Transformer [33] receives a 1D sequence as input. As displayed in Figure 3, to handle a 2D image [29], we divide the input $X \in \mathrm{R}^{H \times W \times C}$ into a series of image patches $X_p \in \mathrm{R}^{N \times (P \times P \times C)}$ and then flatten them into a sequence, where (H, W) indicates the size of the input images, $N = H \times W/P^2$ indicates the patch number, C indicates the channel number, and P represents the length and width of each patch, which is set as 16 in our study.

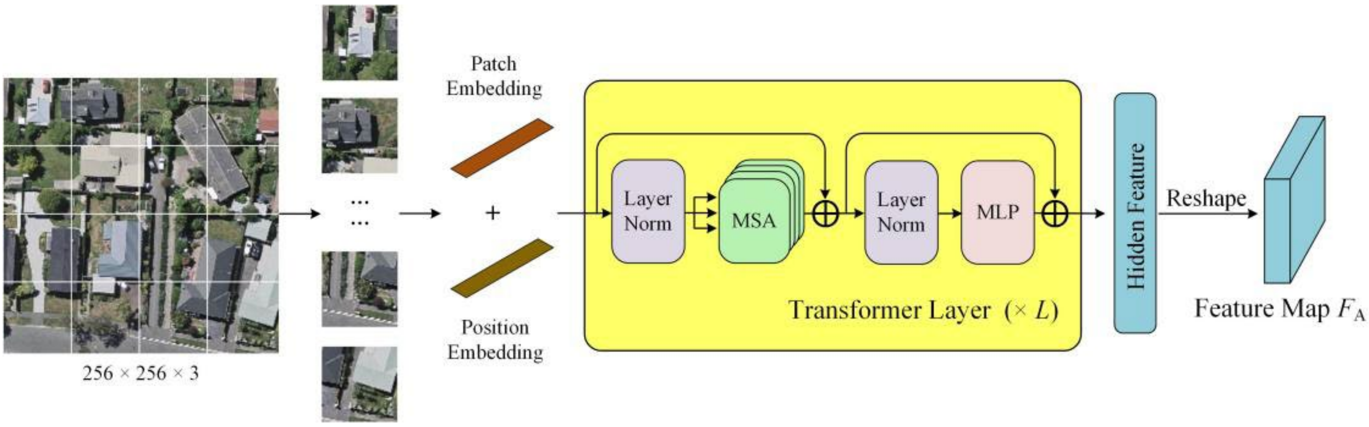

Figure 3. Transformer module.

Each vector patch is mapped to D dimensions with a learnable linear projection, resulting in a patch embedding. Then a 1D position embedding is added to this patch embedding to reserve the associated location information, as displayed in Equation (3):

$$z_0 = [X_p^1\mathrm{E}; X_p^2\mathrm{E}\cdots; X_p^N\mathrm{E}] + \mathrm{E}_{pos}, \mathrm{E} \in \mathrm{R}^{(P^2\cdot C)\times D}, \mathrm{E}_{pos} \in \mathrm{R}^{(N+1)\times D}, \tag{3}$$

where E and E_{pos} denote linear projection functions of the patch embedding and position embedding, respectively, and X_p^N denotes the N-th image patch.

Subsequently, the resulting embedding sequences are input into the Transformer layers. Each layer is composed of stacked MSA and MLP blocks. Layer normalization (LN) is used before each block, and residual connections are applied after each block [29]. The hidden feature representations are obtained by Equations (4) and (5):

$$z'_l = \mathrm{MSA}(\mathrm{LN}(z_{l-1})) + z_{l-1}, l = 1\ldots L, \tag{4}$$

$$z_l = \mathrm{MLP}(\mathrm{LN}(z'_l)) + z'_l, l = 1\ldots L, \tag{5}$$

where z_l represents the l-th encoder feature. A hidden feature representation of size $(H \times W/P^2) \times D$ is obtained by processing the L Transformer layers and reshaping to $(H/P) \times (W/P) \times D$, resulting in the middle feature F_A. In this study, D is set to 768, and the TM module contains 12 Transformer layers and 8 heads in each MSA layer. Section 5 analyses and discusses the parameter selection.

3.2.2. Multiscale Input Convolution Module

The MICM consists of four submodules, each of which has the same double-branch architecture (Figure 4). Taking X as an input, the lower branch extracts features δ_k by using two convolution layers, each of which contains a batch normalization (BN) layer and a rectified linear unit (ReLU) activation function.

$$\delta_k = g(\delta_{k-1}),\ k = \{1,2,3,4\}, \tag{6}$$

where $g(\cdot)$ denotes the double convolution operations and δ_k denotes the convolution feature of the k-th submodule when $k = 1$, $\delta_0 = X$. Then, the SMP is employed for feature abstraction and dimensionality reduction.

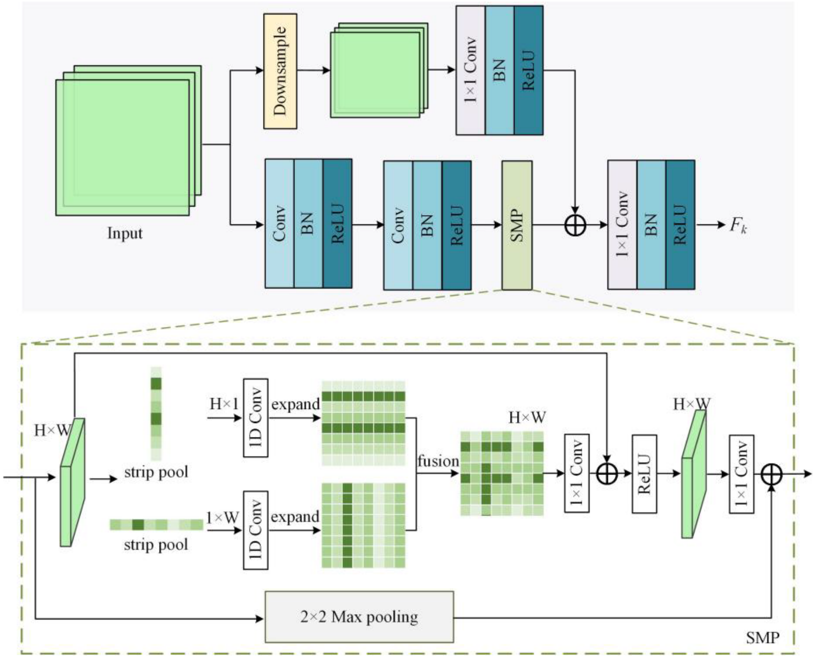

Figure 4. The MICM submodule and SMP architecture.

In this article, SMP is used to replace the max pooling operations of classical networks. Max pooling probes information within square windows, which limits the flexibility in capturing anisotropic context features. Strip pooling [45] resolves this problem well. The given convolution feature δ_k is fed into a horizontal and vertical strip pooling layer simultaneously, resulting in two 1D features $\delta^h_k \in \mathrm{R}^{C\times H}$ and $\delta^v_k \in \mathrm{R}^{C\times W}$:

$$\delta^h_{k\ i} = \frac{1}{W}\sum\nolimits_{0\le j<W}\delta_{k(i,j)}, \tag{7}$$

$$\delta^v_{k\ j} = \frac{1}{H}\sum\nolimits_{0\le i<H}\delta_{k(i,j)}, \tag{8}$$

Subsequently, δ^h_k and δ^v_k are converted into feature matrices with sizes of $H \times W$ via a 1D convolution. Then, the feature map δ'_k of the SMP structure in the k-th submodule is obtained by Equation (9):

$$\delta'_k = \mathrm{MP}(\delta_k) \oplus f^{st=2}(\mathrm{ReLU}(\delta_k \oplus f^{st=1}(\delta^h_{k\ i} + \delta^v_{k\ j}))), k = \{1,2,3,4\}, \tag{9}$$

where $\mathrm{MP}(\cdot)$ denotes a max pooling, $f^{st}(\cdot)$ denotes a 1×1 convolution with a stride size of st, and $\oplus$ represents the feature concatenation operation.

The upper branch downsamples the input and reshapes the feature dimensions to make them consistent with δ'_k. The resulting feature maps are connected with δ'_k, and subsequently a 1×1 convolution is utilized to acquire the subfeature F_k:

$$F_k = f(d^{s_k}(F_{k-1}) \oplus \delta'_k), k = \{1,2,3,4\}, \tag{10}$$

where F_k denotes the intermediate feature of the k-th submodule when $k = 1,\ F_0 = X$, $d(\cdot)$ denotes the downsampling operation, $s = \left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\right\}$ is the downsampling parameter, $f(\cdot)$ denotes a 1×1 convolution, and $\oplus$ represents the feature concatenation operation. The sizes of the four intermediate feature maps are {1282,642,322,162} pixels, and the numbers of channels are {128,256,512,1024}. Finally, the convolution feature F_{B} with a size of $16 \times 16 \times 1024$ is obtained via two convolution layers.

3.2.3. Decoder

The decoder takes the joint feature F, which concatenates the outputs of the TM and MICM, as the input for feature restoration (Figure 2). Two convolution layers are used to reshape the feature dimensions to 16 × 16 × 1024. The resulting feature is restored to the same dimension as the input image by Equation (11):

$$\gamma_k = \mathrm{ReLU}(\mathrm{BN}(\mathrm{TransposeConv}(\gamma_{k-1}))), k = \{1, 2, 3, 4\}, \quad (11)$$

where γ_k denotes the feature map of the k-th upsampling step, when $k = 1$, $\gamma_0 = F$, and $\mathrm{TransposeConv}(\cdot)$ denotes the transposed convolution layer. Four skip connections [46] are adopted to combine the convolution features in the MICM with the upsampled feature maps. Such an operation effectively alleviates the loss of features over successive convolution and pooling operations.

$$\widetilde{\gamma}_k = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}(\gamma_k \oplus \delta_k))), k = \{1, 2, 3, 4\}, \quad (12)$$

where $\widetilde{\gamma}_k$ denotes the feature map of the k-th double convolution, and the numbers of feature channels are {512,256,128,64}. Finally, the feature maps with 2 channels are acquired via a 1 × 1 convolution, and these maps are fed into the sigmoid layer to obtain the prediction result R.

3.3. Discriminator Network

An FCN-based discriminator network is designed; it contains a double-branch structure with different convolution kernel sizes. More information about different receptive fields can be obtained by multiscale inputs and convolution kernels with different sizes. The discriminator network receives the segmentation result R or ground-truth maps as input, as shown in part II of Figure 2. Features are extracted from the upper and lower branches (Equations (13) and (14)):

$$F_k{}^U = \mathrm{LeakyReLU}(\mathrm{Conv}_{ke=4}^{st=2}(F_{k-1}^U)), k = \{1, 2, 3, 4\}, \quad (13)$$

$$F_k{}^D = \mathrm{LeakyReLU}(\mathrm{Conv}_{ke=2}^{st=2}(d^s(R)_{k-1})), k = \{2, 3, 4\}, \quad (14)$$

where F_k^U and F_k^D denote the features obtained by the k-th convolution in the upper and lower branches, respectively. When k = 1, $F_0^U = R$, $\mathrm{Conv}_{ke}^{st}(\cdot)$ represents a convolution with strides of st and kernel sizes of ke, $\mathrm{LeakyReLU}(\cdot)$ denotes the leaky ReLU activation function, and $d^s(\cdot)$ denotes the downsampling operation with a parameter $s = 1/2$. The numbers of channels in the resulting four feature maps are {64,128,256,512}. Subsequently, the feature maps generated by the two branches are concatenated and fed into a 1 × 1 convolution and a classification layer. Last, the confidence map is acquired via a sigmoid operation, in which each pixel represents the approximation degree of the pixels in the segmented map with respect to the sample label. This map is utilized as a supervisory signal for unlabeled data.

3.4. Loss Function

The segmentation network and discriminator network are trained jointly via labeled samples. When inputting unlabeled samples, the discriminator network generates confidence maps to supervise the training of the segmentation network in a self-taught mechanism. The discriminator network is optimized by minimizing the binary cross-entropy loss L_D:

$$L_D = -\sum_{i,j}((1-y)\log(1 - O_{(i,j)}^R) + y \log O_{(i,j)}^Y), i \in H, j \in W, \quad (15)$$

where $O_{(i,j)}^R$ and $O_{(i,j)}^Y$ represent confidence maps for the prediction maps R and ground-truth labels Y, respectively, (i, j) denotes pixel locations, and y represents the label of each pixel.

The multitask loss in [24] is optimized to train the segmentation network:

$$L_{Seg} = L_{CE} + \lambda_{adv} L_{adv} + \lambda_{semi} L_{semi}, \tag{16}$$

where L_{CE}, L_{adv} and L_{semi} respectively indicate the cross-entropy loss, adversarial loss, and semi-supervised loss, and λ_{adv} and λ_{semi} are weights utilized for adjusting L_{Seg}. In this study, λ_{adv} is respectively set to 0.01 and 0.001 while using labeled and unlabeled samples. λ_{semi} is equal to 0.1. Taking C as the number of categories, L_{CE} is obtained by Equation (17):

$$L_{CE} = -\sum_{i,j} \sum_{c \in C} Y_{(i,j,c)} \log(R_{(i,j,c)}), i \in H, j \in W \tag{17}$$

The adversarial loss and semi-supervised loss are shown in Equations (18) and (19), respectively:

$$L_{adv} = -\sum_{i,j} \log O^{R}_{(i,j)}, \tag{18}$$

$$L_{semi} = \begin{cases} -\sum_{i,j,c} Y^{u}_{c} \log R^{u}_{(i,j,c)}, & \text{if } O_{(i,j)} \geq \tau \\ 0, & \text{otherwise} \end{cases}, \tag{19}$$

where $R^{u}_{(i,j,c)}$ denotes the class c prediction results of the unlabeled data at location (i, j), Y^{u}_{c} denotes the pseudo-label of the class c of unlabeled data, $O_{(i,j)}$ represents the confidence map, and τ is a threshold value of 0.2.

4. Results

4.1. Datasets

Three open-source remote sensing datasets with different spatial resolutions, including the WBD [35], MBD [36] and GID [10], were used for method verification. We clipped all images and labels into 256 × 256 image patches for model training and classification. Some building examples contained in the three datasets are shown in Figure 5. The labels were uniformly processed into binary images with a target value of 1 and a background value of 0.

- WBD: This building dataset consists of 8189 aerial image tiles and contains 187,000 buildings with diverse usages, sizes and colors in Christchurch, New Zealand. The spatial resolution is 0.3 m. After cropping without overlap, 15,256 image patches were selected and randomly split into 14,256 patches for training and 1000 patches for testing.
- MBD: The MBD is a large dataset for building segmentation that consists of 151 aerial images of the Boston area with 1500 × 1500 pixels. The spatial resolution is 1 m. A total of 11,384 image patches containing buildings with 256 × 256 pixels were chosen after cropping. These patches were further randomly divided into 10,384 patches for training and 1000 patches for testing.
- GID: This land-use dataset contains 5 land-use categories and 150 Gaofen-2 satellite images, obtained from more than 60 different cities in China. The spatial resolution is 4 m. We extracted the building class and constructed a dataset containing 13,671 image patches for our experiments, among which 12,175 patches were used for training and 1496 were used for testing.

4.2. Experimental Procedure

4.2.1. Method Implementation

Several well-known semantic segmentation networks, i.e., DeepLabv2 [8], PSPNet [7], UNet [46], and TransUNet [44], with combinations of Transformer and convolution, were used for method comparisons under the SSAL framework. ResNet-101 was used as the backbone for DeepLabv2 and PSPNet. The numbers of Transformer layers and attention heads in TransUNet are set to 12 [44]. To validate the proposed method, we randomly sampled 1/8, 1/4 and 1/2 of images as labeled data and the remainder as unlabeled data. The quantities of labeled data are displayed in Table 1.

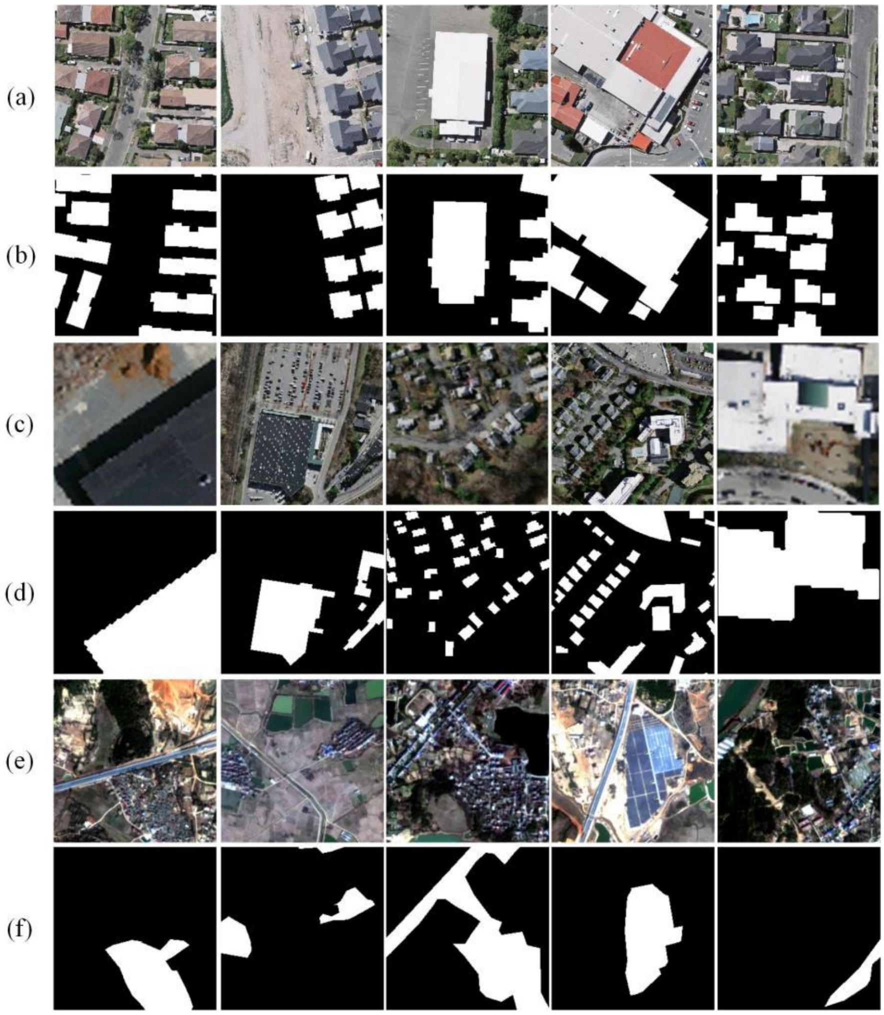

Figure 5. Different buildings in the three datasets: (**a**,**b**) are WBD building images and corresponding labels, (**c**,**d**) are MBD building images and corresponding labels, and (**e**,**f**) are GID building images and corresponding labels, respectively.

Table 1. Amounts of labeled data.

Datasets	**Labeled Data Amount**			
	1/8	**1/4**	**1/2**	**Full**
WBD	1782	3564	7128	14,256
MBD	1298	2596	5192	10,384
GID	1522	3044	6088	12,671

All models were implemented with Python 3.6 and PyTorch 1.2.0, which were powered by a 24-GB NVIDIA GeForce RTX 3090 GPU. The segmentation network was optimized using the stochastic gradient descent approach. The original learning rate was 2.5×10^{-4} and was declined via polynomial decay with a power of 0.9. The Adam optimizer [47], where the learning rate is 1×10^{-4}, was utilized to optimize the discriminator network. All networks were trained over 80 K iterations and the batch size was 4. Adopting the same strategy used in [24], we started SSL after training 5000 iterations with labeled samples to avoid the model being influenced by the original noisy masks and predictions.

4.2.2. Method Evaluation Measures

Four assessment indices, precision, recall, F1 and mean intersection over union (mIoU), were utilized to evaluate the different methods. Equation (20) gives the definitions of these metrics:

$$\begin{aligned} &\text{Precision} = \frac{TP}{TP+FP} \\ &\text{Recall} = \frac{TP}{TP+FN} \\ &\text{F1} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \\ &\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP}{TP+FP+FN} \end{aligned}, \qquad (20)$$

where TP indicates the quantity of building pixels correctly categorized, FP indicates the quantity of nonbuilding pixels categorized as buildings, FN indicates the quantity of building pixels incorrectly categorized as nonbuildings, and C is the quantity of categories. The F1 and mIoU metrics were utilized to comprehensively assess the model performance.

4.3. Experimental Results and Analysis

All the networks were trained on the WBD, MBD and GID using different quantities of labeled samples under the SSAL framework. The test sets did not participate in the model training and were used for evaluating and comparing the method performance.

4.3.1. Quantitative Analyses

Tables 2–4 show the building extraction accuracies achieved on the three datasets. In general, adding the quantity of labeled samples increases the accuracy measures of each approach. The F1 and mIoU measures of the proposed TRANet were the best on the three datasets, and this finding was consistent with the subsequent visualization analysis.

Table 2. Building extraction accuracies obtained with different quantities of labeled data on the WBD. The highest accuracy is displayed in bold.

Method	Labeled Data Amount							
	1/8				1/4			
	Recall	Precision	F1	mIoU	Recall	Precision	F1	mIoU
DeepLabv2	0.8965	0.8713	0.8837	0.8714	0.9187	0.8586	0.8876	0.8759
PSPNet	0.8834	0.8267	0.8541	0.8429	0.8886	0.8301	0.8583	0.8470
UNet	0.9293	0.9284	0.9288	0.9182	0.9421	0.9352	0.9387	0.9290
TransUNet	0.9193	0.9202	0.9197	0.9084	0.9362	0.9282	0.9322	0.9219
TRANet	**0.9364**	**0.9301**	**0.9332**	**0.9230**	**0.9495**	**0.9346**	**0.9420**	**0.9327**
Method	1/2				Full			
	Recall	Precision	F1	mIoU	Recall	Precision	F1	mIoU
DeepLabv2	0.8973	0.8924	0.8949	0.8824	0.9204	0.8831	0.9013	0.8895
PSPNet	0.9002	0.8220	0.8593	0.8483	0.9020	0.8294	0.8642	0.8529
UNet	0.9512	0.9394	0.9453	0.9364	0.9554	0.9408	0.9480	0.9394
TransUNet	0.9457	0.9317	0.9387	0.9290	0.9496	0.9337	0.9416	0.9323
TRANet	**0.9547**	**0.9402**	**0.9474**	**0.9387**	**0.9571**	**0.9421**	**0.9495**	**0.9411**

Table 3. Building extraction accuracies obtained with different quantities of labeled data on the MBD. The highest accuracy is displayed in bold.

Method	**Labeled Data Amount**							
	1/8				**1/4**			
	Recall	**Precision**	**F1**	**mIoU**	**Recall**	**Precision**	**F1**	**mIoU**
DeepLabv2	0.7706	0.4964	0.6038	0.6704	0.7032	0.5799	0.6356	0.6856
PSPNet	0.7296	0.5224	0.6088	0.6714	0.7576	0.4923	0.5968	0.6659
UNet	0.7490	**0.6819**	0.7139	0.7380	0.7752	0.6943	0.7325	0.7523
TransUNet	0.7252	0.6437	0.6820	0.7156	0.7630	0.6852	0.7220	0.7443
TRANet	**0.7839**	0.6693	**0.7221**	**0.7454**	**0.7785**	**0.7178**	**0.7469**	**0.7627**
Method	**1/2**				**Full**			
	Recall	**Precision**	**F1**	**mIoU**	**Recall**	**Precision**	**F1**	**mIoU**
DeepLabv2	0.7398	0.5526	0.6326	0.6858	0.7292	0.6312	0.6766	0.7124
PSPNet	0.7590	0.5062	0.6073	0.6720	0.7623	0.5060	0.6083	0.6726
UNet	**0.7988**	0.7225	0.7588	0.7723	0.8127	0.7402	0.7748	0.7848
TransUNet	0.7926	0.7001	0.7435	0.7608	0.8047	0.7180	0.7589	0.7726
TRANet	0.7987	**0.7355**	**0.7658**	**0.7775**	**0.8160**	**0.7482**	**0.7806**	**0.7894**

Table 4. Building extraction accuracies obtained with different quantities of labeled data on the GID. The highest accuracy is displayed in bold.

Method	**Labeled Data Amount**							
	1/8				**1/4**			
	Recall	**Precision**	**F1**	**mIoU**	**Recall**	**Precision**	**F1**	**mIoU**
DeepLabv2	**0.8560**	0.6946	0.7669	0.7679	**0.8281**	0.7381	0.7805	0.7773
PSPNet	0.8003	0.6553	0.7205	0.7302	0.8064	0.6701	0.7320	0.7388
UNet	0.7647	0.7460	0.7552	0.7535	0.7731	0.7442	0.7583	0.7565
TransUNet	0.7904	0.7534	0.7715	0.7679	0.7538	0.7711	0.7624	0.7582
TRANet	0.7659	**0.7939**	**0.7797**	**0.7728**	0.7765	**0.8052**	**0.7905**	**0.7823**
Method	**1/2**				**Full**			
	Recall	**Precision**	**F1**	**mIoU**	**Recall**	**Precision**	**F1**	**mIoU**
DeepLabv2	0.8288	0.7533	0.7893	0.7844	0.8358	0.7507	0.7910	0.7862
PSPNet	0.7850	0.7122	0.7468	0.7486	0.8268	0.6851	0.7493	0.7530
UNet	0.8154	0.7326	0.7718	0.7697	0.8326	0.7532	0.7909	0.7860
TransUNet	0.8368	0.7519	0.7921	0.7872	0.8240	0.7687	0.7954	0.7892
TRANet	**0.8406**	**0.7597**	**0.7981**	**0.7923**	**0.8433**	**0.7720**	**0.8061**	**0.7991**

As shown in Table 2, the building extraction accuracies of all methods on the WBD were higher than 90%, except DeepLabv2 and PSPNet. PSPNet performed worst among all the models. When trained with fully labeled data, the four measures yielded by TRANet increased by 5.51%, 11.27%, 8.53% and 8.82%, compared with those of PSPNet. The UNet model performed the second best. With only 1/8 of the labeled data, UNet's F1 and mIoU values were 92.88% and 91.82%, respectively, which were 0.5% lower than those of TRANet. The accuracy of TransUNet was slightly lower than that of UNet. The Transformer structure is added only at the top of the TransUNet encoder, resulting in limited global information. TRANet, which combines the Transformer and convolution, performed the best on the WBD.

Table 3 lists the accuracy measures produced by the different methods on the MBD. The accuracies of all models were lower than 80%. With 1/8 of the labeled data, the F1 and mIoU measures of TRANet were 72.21% and 74.54%, respectively, which were 5% lower than those obtained using fully labeled data. However, this method still performed the best. TRANet's F1 and mIoU increased by approximately 0.82%~11.83% and 0.74%~7.5%, respectively, compared with those of other methods. The UNet model performed suboptimally. The F1 and mIoU measures of TransUNet were 3.19% and 2.24% lower than those of UNet, respectively, under 1/8 of the labeled data. The performances of DeepLabv2 and PSPNet were poor, and all the F1 and mIoU values were lower than 70%. The DeepLabv2 model performed slightly better than PSPNet.

On the GID, as shown in Table 4, TransUNet, using the Transformer structure, achieved better building extraction accuracy than UNet. When trained with 1/8 labeled samples, TransUNet's F1 and mIoU values were 1.63% and 1.44% better than those of UNet, respectively. DeepLabv2 performed better than PSPNet and UNet. When trained with fully labeled data, DeepLabv2's F1 and mIoU were 1.51% and 1.29% less than those of TRANet, respectively. TRANet performed the best. The four measures of TRANet, when training with 1/2 labeled data, decreased by 0.27%, 1.23%, 0.8%, and 0.68% relative to the metrics obtained when training with fully labeled data, where TRANet achieved an accuracy similar to that of using fully supervised training.

4.3.2. Qualitative Analyses

The semantic segmentation results obtained when training with 1/8 labeled samples under the SSAL framework were used for visual analysis. Figures 6–8 show the representative building regions derived with the three datasets.

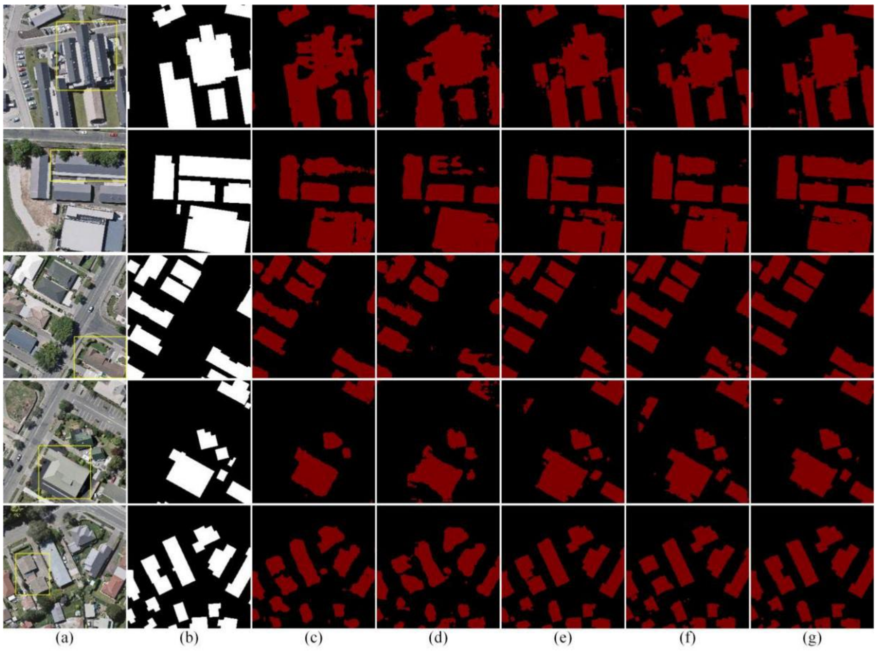

Figure 6. Typical building extraction results on the WBD. (**a**) Images. (**b**) Labels. (**c**) DeepLabv2. (**d**) PSPNet. (**e**) UNet. (**f**) TransUNet. (**g**) TRANet. Yellow boxes identify partial differences between the different methods.

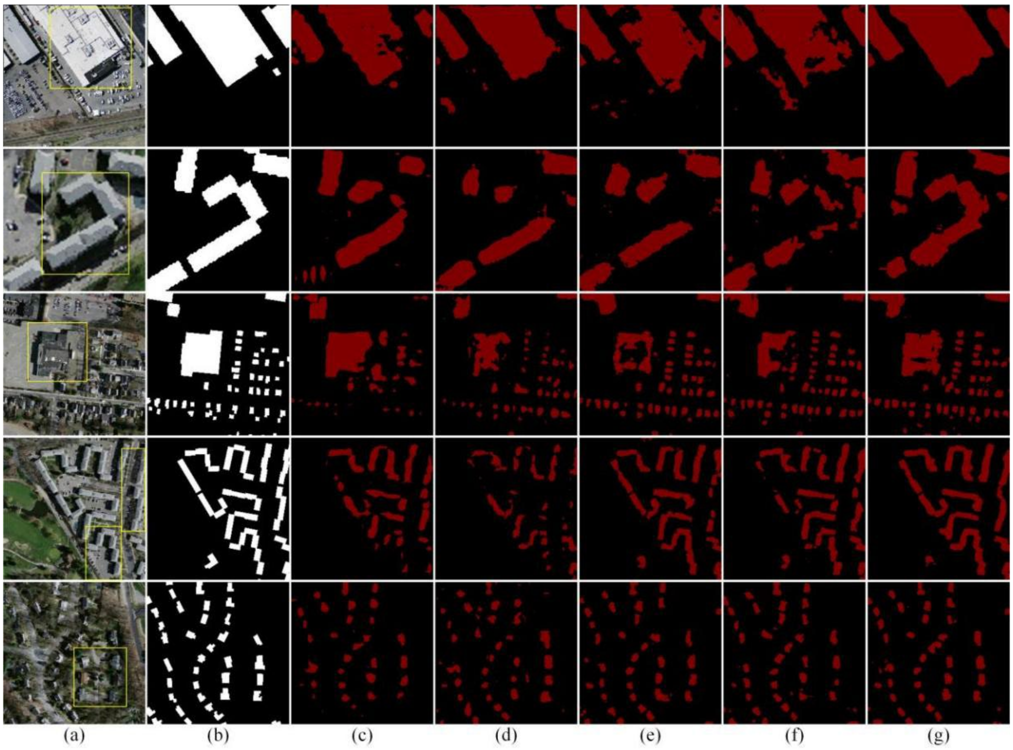

Figure 7. Typical building extraction results on the MBD. (**a**) Images. (**b**) Labels. (**c**) DeepLabv2. (**d**) PSPNet. (**e**) UNet. (**f**) TransUNet. (**g**) TRANet. Yellow boxes identify partial differences between the different methods.

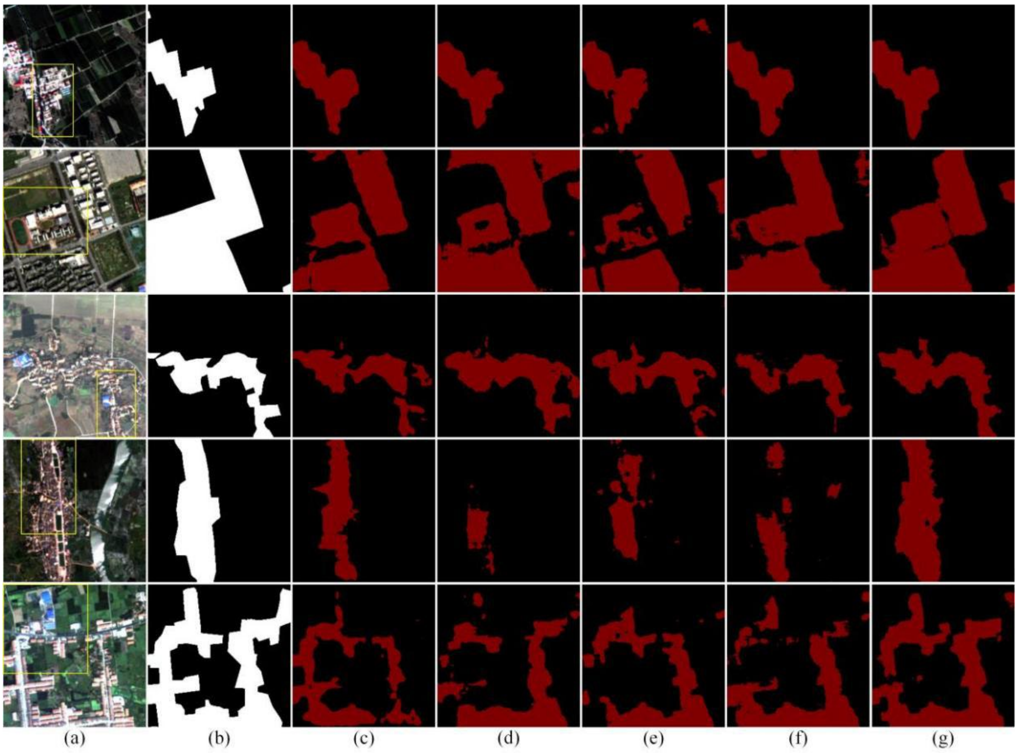

Figure 8. Typical building extraction results on the GID. (**a**) Images. (**b**) Labels. (**c**) DeepLabv2. (**d**) PSPNet. (**e**) UNet. (**f**) TransUNet. (**g**) TRANet. Yellow boxes identify partial differences between the different methods.

The WBD has high resolution and good image quality. Figure 6c,d show that the results obtained by DeepLabv2 and PSPNet exhibited many missed extractions and falsely extracted areas, and obvious distortions were present on the edges of buildings, especially in subregions 1 and 2. The extraction results of UNet and TransUNet had fewer missed extractions (subregions 1 and 2) and falsely extracted areas (subregion 4). TRANet extracted more complete building surfaces in subregions 2–5, and the details were closer to the reference labels.

The resolution of the MBD is 1 m. Many buildings with small areas are represented by only a few to more than a dozen pixels in the corresponding images; this situation brings difficulties to the fine extraction of buildings. As shown in Figure 7c,d, all results obtained by PSPNet and DeepLabv2 had large numbers of missed extractions, and the extracted buildings had irregular shapes and fuzzy boundaries. UNet extracted more complete small buildings with clear boundaries, as shown in Figure 7e, but obvious losses existed in the large buildings of subregion 3. In addition, the strip buildings in subregions 1, 2, and 4 were extracted incompletely. TRANet extracted complete buildings, especially in subregions 4 and 5 of Figure 7g, and the boundaries of small buildings and the surfaces of strip buildings demonstrated the better performance of this method, although small, missed extractions existed in subregions 2 and 3.

The GID has good image quality but relatively low resolution. Multiple complex objects, i.e., water bodies, roads, farmland, bare land, etc., are contained in one image. Buildings have irregular edges and are mostly distributed in pieces, which are easily mixed with other types of objects. Such a situation increases the difficulty of building extraction. Overall, all extraction results had missed extractions and falsely extracted areas. The falsely extracted areas in the results obtained by DeepLabv2, PSPNet, UNet and TransUNet were smaller, as shown in Figure 8c–f, but there were more missed extractions in subregions 2, 4 and 5. TRANet extracted more complete buildings than other models.

Based on the aforementioned quantitative and qualitative analyses, the proposed TRANet performed the best. TRANet uses the Transformer to obtain global contextual information and the MICM to extract local multiscale features simultaneously. The proposed SMP structure is designed to retain horizontal and vertical features, which alleviates the loss of details over continuous convolution operations. All these designs facilitate improvements in the building extraction accuracy.

5. Discussion

We performed four groups of ablation experiments to validate the performance of the designed double-branch segmentation network, the MICM, the SMP, and the discriminator network. The double-branch encoder is the core of TRANet, and it was verified by semi-supervised experiments with the WBD, MBD and GID under different amounts of labeled data, to fully illustrate the advantages of the Transformer combined with convolution. For the other three groups, 7128 labeled samples and 7128 unlabeled samples from the WBD were selected for the ablation experiments.

5.1. Comparison between Single/Double-Branch Encoder Structures

The encoder of the TRANet segmentation network contains a parallel TM and MICM, and it was verified via module replacement, along with the fixed decoder and discriminator network under the SSAL framework. Table 5 shows that the accuracies were low when the TM was used alone as the encoder, among which the F1 and mIoU were approximately 8.11~18.96% and 8.44~12.69% less than those obtained by the encoder using the MICM alone, respectively. The Transformer focuses on context modeling during the encoding phase and ignores the detailed localization of low-level features, which is hardly restored by upsampling. Convolution operations can extract rich low-level features. Combining the Transformer with convolution facilitates the improvement in the segmentation accuracy. The F1 and mIoU increased by approximately 0.13~19.44% and 0.14~13.09%, respectively, over the results obtained by using the single encoder. Therefore, TRANet utilizes the advantages of the Transformer and convolution to extract robust features, thereby improving semantic segmentation accuracy.

Table 5. Building extraction accuracies with single/double-branch encoders. The highest accuracy is displayed in bold.

Dataset	Encoder	Labeled Data Amount											
		1/8				1/4				1/2			
		Recall	Precision	F1	mIoU	Recall	Precision	F1	mIoU	Recall	Precision	F1	mIoU
WBD	TM	0.8258	0.8205	0.8231	0.8125	0.8599	0.8465	0.8532	0.8411	0.8801	0.8505	0.8650	0.8529
	MICM	0.9355	0.9293	0.9324	0.9221	0.9364	**0.9396**	0.9380	0.9282	**0.9562**	0.9361	0.9461	0.9373
	TM+MICM	**0.9364**	**0.9301**	**0.9332**	**0.9230**	**0.9495**	0.9346	**0.9420**	**0.9327**	0.9547	**0.9402**	**0.9474**	**0.9387**
MBD	TM	0.5461	0.5187	0.5321	0.6169	0.6161	0.4859	0.5433	0.6291	0.6579	0.5050	0.5714	0.6466
	MICM	0.7477	0.6688	0.7060	0.7327	**0.7809**	0.7103	0.7439	0.7607	0.7892	0.7348	0.761	0.7735
	TM+MICM	**0.7839**	**0.6693**	**0.7221**	**0.7454**	**0.7785**	0.7178	**0.7469**	**0.7627**	**0.7987**	**0.7355**	**0.7658**	**0.7775**
GID	TM	0.7228	0.5878	0.6483	0.6762	0.7534	0.6269	0.6843	0.7019	0.7630	0.6299	0.6901	0.7065
	MICM	0.7584	0.7734	0.7658	0.7613	**0.7815**	0.7702	0.7758	0.7708	0.8039	**0.7787**	0.7911	0.7846
	TM+MICM	**0.7659**	**0.7939**	**0.7797**	**0.7728**	0.7765	**0.8052**	**0.7905**	**0.7823**	**0.8406**	0.7597	**0.7981**	**0.7923**

5.2. Comparison among Different Pooling Modules

The proposed SMP was verified by module replacement along with the fixed decoder and discriminator network under the SSAL framework. One set of experiments used a single-branch encoder, containing four simple "convolution-pooling" architectures, where the pooling layer was successively replaced by max pooling, strip pooling [45], and the SMP structure. These corresponding alternates were represented by CNN_MP, CNN_SP, and CNN_SMP. Another set of experiments used a double-branch encoder combining the TM and the aforementioned "convolution-pooling" architectures, which were represented by TM+CNN_MP, TM+CNN_SP, and TM+CNN_SMP. The achieved accuracy measures are listed in Table 6. The single- or double-branch encoders using the SMP performed the best when compared with those using other pooling structures, thereby proving the proposed SMP structure.

Table 6. Accuracy assessment of TRANet in terms of building extraction with different pooling modules. The highest accuracy is displayed in bold.

Method	Recall	Precision	F1	mIoU
CNN_MP	0.9476	0.9360	0.9418	0.9325
CNN_SP	0.9502	0.9346	0.9424	0.9332
CNN_SMP	**0.9532**	**0.9391**	**0.9461**	**0.9373**
TM+CNN_MP	0.9518	0.9398	0.9458	0.9369
TM+CNN_SP	0.9453	0.9366	0.9409	0.9315
TM+CNN_SMP	**0.9547**	**0.9402**	**0.9474**	**0.9387**

5.3. Comparison among Different Multiscale Modules

The MICM was verified by module replacement along with the fixed decoder and discriminator network under the SSAL framework. One set of experiments used a single-branch encoder, containing four simple "convolution-pooling" architectures and added atrous spatial pyramid pooling (ASPP) [8], selective kernel (SK) [48], and MICM modules to the encoder, which were represented by CNN, CNN+ASPP, CNN+SK, and CNN+MICM, respectively. Another set of experiments used the aforementioned double-branch encoder with different multiscale modules, which were represented by TM+CNN, TM+CNN+ASPP, TM+CNN+SK, and TM+CNN+MICM. Table 7 shows that the methods using multiscale modules achieved higher accuracy than those that did not utilize multiscale modules. Both the single- and double-branch encoders using the MICM performed better than those using other multiscale modules. The MICM captures multiscale input maps before feature extraction, which reduces the loss of details caused by continuous convolution operations with limited receptive fields.

Table 7. Building extraction accuracies with different multiscale modules. The highest accuracy is displayed in bold.

Method	Recall	Precision	F1	mIoU
CNN	0.9476	0.9360	0.9418	0.9325
CNN+ASPP	0.9515	0.9357	0.9435	0.9344
CNN+SK	0.9546	**0.9379**	0.9462	0.9374
CNN+MICM	**0.9559**	**0.9379**	**0.9468**	**0.9381**
TM+CNN	0.9518	0.9398	0.9458	0.9369
TM+CNN+ASPP	0.9540	0.9377	0.9458	0.9370
TM+CNN+SK	0.9539	0.9391	0.9464	0.9377
TM+CNN+MICM	**0.9547**	**0.9402**	**0.9474**	**0.9387**

5.4. Comparison among Different Discriminator Networks

The discriminator network in [24] and that proposed in this paper (represented by an additional *), along with five segmentation networks, including DeepLabv2, PSPNet, UNet, TransUNet and TRANet, were utilized for model training under the SSAL framework. Table 8 presents the achieved accuracy measures. The developed discriminator network facilitated the same segmentation network to obtain higher segmentation accuracy. This strategy was effective for all five segmentation networks. The proposed discriminator network can capture more information with different receptive fields by utilizing multiscale inputs and convolutions with different kernel sizes.

Table 8. Building extraction accuracies with different discriminator networks. The highest accuracy is displayed in bold.

Method	Recall	Precision	F1	mIoU	Method	Recall	Precision	F1	mIoU
DeepLabv2	0.9042	0.8564	0.8797	0.8677	DeepLabv2 *	0.8973	0.8924	0.8949	0.8824
PSPNet	0.8738	0.8283	0.8504	0.8391	PSPNet *	0.9002	0.8220	0.8593	0.8483
UNet	0.9415	0.9329	0.9372	0.9274	UNet *	0.9512	0.9394	0.9453	0.9364
TransUNet	0.9451	0.9302	0.9376	0.9279	TransUNet *	0.9457	0.9317	0.9387	0.9290
TRANet	0.9504	0.9386	0.9445	0.9354	TRANet *	**0.9547**	**0.9402**	**0.9474**	**0.9387**

5.5. Model Parameter Discussions

Two important parameters in the TM of TRANet, the number of Transformer layers and number of heads, are represented by layer_num and head_num, respectively. We used 7128 labeled data and 7128 unlabeled data from the WBD for semi-supervised training, with different parameter settings, and analyzed the network performance. When the influence of layer_num was analyzed, head_num was fixed to 8, and layer_num was set to {4,8,12,16,20}. When the influence of head_num was analyzed, layer_num was fixed to 12, and head_num was set to {2,4,8,12,16}. Tables 9 and 10 show that the highest accuracy was obtained when layer_num was 12 and head_num was 8. Therefore, this set of values was used in all experiments in this study.

Table 9. Building extraction accuracies under different layer_num settings when head_num = 8. The highest accuracy is displayed in bold.

layer_num	Recall	Precision	F1	mIoU
4	0.9477	0.9236	0.9355	0.9257
8	0.9502	0.9282	0.9391	0.9296
12	**0.9547**	**0.9402**	**0.9474**	**0.9387**
16	0.9443	0.9361	0.9402	0.9307
20	0.9453	0.9278	0.9365	0.9267

Table 10. Building extraction accuracies under different head_num settings when layer_num = 12. The highest accuracy is displayed in bold.

head_num	Recall	Precision	F1	mIoU
2	0.9461	0.9336	0.9398	0.9303
4	0.9467	0.9347	0.9407	0.9313
8	**0.9547**	0.9402	**0.9474**	**0.9387**
12	0.9456	0.9327	0.9391	0.9295
16	0.9328	**0.9407**	0.9367	0.9268

6. Conclusions

In this article, we designed a novel semi-supervised adversarial semantic segmentation network for object extraction, from high-resolution remote sensing imagery, which leverages both the local feature extraction advantages of CNNs and the global context modeling abilities of the Transformer. Experimental results on three datasets with different spatial resolutions show that TRANet significantly increases the building extraction accuracies and makes the acquired segmentation results close to those obtained via fully supervised learning when a small number of labeled data are available. Future works will further fuse the multilevel features of the Transformer and CNNs to obtain more refined object information, thereby enhancing the performance of the segmentation network and applying it to segmentation tasks involving other objects in high-resolution remote sensing imagery.

Author Contributions: Conceptualization, Y.Z., M.W., X.Q., X.Z. and W.D.; methodology, Y.Z. and M.W.; validation, Y.Z. and M.Y.; formal analysis, Y.Z. and M.W.; data curation, Y.Z. and R.Y.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z., M.Y. and M.W.; supervision, Y.Z. and M.W.; visualization, Y.Z.; funding acquisition, M.W. and Y.Z.; project administration, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (grant number 2021YFB3901300), the National Natural Science Foundation of China (grant numbers 42071301, 41671341), the Jiangsu Province Water Conservancy Science and Technology Project (grant number 2021064), the Chongqing Agricultural Industry Digital Map Project (grant number 21C00346), and the Postgraduate Research and Practice Innovation Program of Jiangsu Province (KYCX21_1349).

Data Availability Statement: The data provided in this work are available from the corresponding authors.

Acknowledgments: We would like to sincerely thank the editors and reviewers for their time.

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

References

1. Kang, J.; Wang, Z.; Zhu, R.; Sun, X.; Fernandez-Beltran, R.; Plaza, A. PiCoCo: Pixelwise Contrast and Consistency Learning for Semisupervised Building Footprint Segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 10548–10559. [CrossRef]
2. Su, Y.; Cheng, J.; Bai, H.; Liu, H.; He, C. Semantic Segmentation of Very-High-Resolution Remote Sensing Images via Deep Multi-Feature Learning. *Remote Sens.* **2022**, *14*, 533. [CrossRef]
3. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 640–651. [CrossRef]
4. Alshehhi, R.; Marpu, P.R.; Woon, W.L.; Mura, M.D. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* **2017**, *130*, 139–149. [CrossRef]
5. Li, Y.; Lu, H.; Liu, Q.; Zhang, Y.; Liu, X. SSDBN: A Single-Side Dual-Branch Network with Encoder–Decoder for Building Extraction. *Remote Sens.* **2022**, *14*, 768. [CrossRef]
6. Kang, J.; Guan, H.; Peng, D.; Chen, Z. Multi-scale context extractor network for water-body extraction from high-resolution optical remotely sensed images. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *103*, 102499. [CrossRef]
7. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239. [CrossRef]
8. 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.* **2018**, *40*, 834–848. [CrossRef]

9. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, NY, USA, 15–20 June 2019; pp. 3141–3149. [CrossRef]
10. Tong, X.; Xia, G.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-Cover Classification with High-Resolution Remote Sensing Images Using Transferable Deep Models. *arXiv* **2019**, arXiv:1807.05713. Available online: https://arxiv.org/abs/1807.05713 (accessed on 20 November 2019). [CrossRef]
11. Zhang, M.; Hu, X.; Zhao, L.; Lv, Y.; Luo, M. Learning dual multi-scale manifold ranking for semantic segmentation of high-resolution images. *Remote Sens.* **2017**, *9*, 500. [CrossRef]
12. Gerke, M.; Rottensteiner, F.; Wegner, J.D.; Sohn, G. ISPRS Semantic Labeling Contest. 2014. Available online: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx (accessed on 7 September 2014).
13. Kemker, R.; Luu, R.; Kanan, C. Low-shot learning for the semantic segmentation of remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6214–6223. [CrossRef]
14. Wambugu, N.; Chen, Y.; Xiao, Z.; Tan, K.; Wei, M.; Liu, X.; Li, J. Hyperspectral image classification on insufficient-sample and feature learning using deep neural networks: A review. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *105*, 102603. [CrossRef]
15. Lee, D.H. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013.
16. Qiao, S.; Shen, W.; Zhang, Z.; Wang, B.; Yuille, A. Deep Co-Training for Semi-Supervised Image Recognition. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 142–159. [CrossRef]
17. Laine, S.; Aila, T. Temporal ensembling for semisupervised learning. *arXiv* **2017**, arXiv:1610.02242. Available online: https://arxiv.org/abs/1610.02242 (accessed on 15 March 2017).
18. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semisupervised deep learning results. *arXiv* **2017**, arXiv:1703.01780. Available online: https://arxiv.org/abs/1703.01780 (accessed on 6 March 2017).
19. Berthelot, D.; Carlini, N.; Goodfellow, I.; Oliver, A.; Papernot, N.; Raffel, C. MixMatch: A holistic approach to semi-supervised learning. *arXiv* **2019**, arXiv:1905.02249. Available online: https://arxiv.org/abs/1905.02249 (accessed on 23 October 2019).
20. Sohn, K.; Berthelot, D.; Li, C.; Zhang, Z.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Zhang, H.; Raffel, C. FixMatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv* **2020**, arXiv:2001.07685v2. Available online: https://arxiv.org/abs/2001.07685v2 (accessed on 25 November 2020).
21. Odena, A. Semi-supervised learning with generative adversarial networks. *arXiv* **2016**, arXiv:1606.01583.
22. Wang, L.; Sun, Y.; Wang, Z. CCS-GAN: A semi-supervised generative adversarial network for image classification. *Vis. Comput.* **2021**, *4*, 1–13. [CrossRef]
23. Luc, P.; Couprie, C.; Chintala, S.; Verbeek, J. Semantic segmentation using adversarial networks. *arXiv* **2016**, arXiv:1611.08408. Available online: https://arxiv.org/abs/1611.08408 (accessed on 25 November 2016).
24. Hung, W.C.; Tsai, Y.H.; Liou, Y.T.; Lin, Y.Y.; Yang, M.H. Adversarial learning for semi-supervised semantic segmentation. *arXiv* **2018**, arXiv:1802.07934. Available online: https://arxiv.org/abs/1802.07934 (accessed on 24 July 2018).
25. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *arXiv* **2014**, arXiv:1406.2661. [CrossRef]
26. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Zhang, L. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv* **2020**, arXiv:2012.15840. Available online: https://arxiv.org/abs/2012.15840 (accessed on 31 December 2020).
27. Chen, Z.; Wang, C.; Li, J.; Fan, W.; Du, J.; Zhong, B. Adaboost-like End-to-End multiple lightweight U-nets for road extraction from optical remote sensing images. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *100*, 2341. [CrossRef]
28. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
29. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
30. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision transformer using shifted windows. *arXiv* **2021**, arXiv:2103.14030. [CrossRef]
31. Yang, F.; Yang, H.; Fu, J.; Lu, H.; Guo, B. Learning texture transformer network for image super-resolution. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Seattle, DC, USA, 13–19 June 2020; pp. 5790–5799. [CrossRef]
32. Wang, Z.; Zhao, J.; Zhang, R.; Li, Z.; Lin, Q.; Wang, X. UATNet: U-Shape Attention-Based Transformer Net for Meteorological Satellite Cloud Recognition. *Remote Sens.* **2022**, *14*, 104. [CrossRef]
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, NY, USA, 4–9 December 2017; pp. 6000–6010. [CrossRef]
34. Zhang, Y.; Liu, H.; Hu, Q. TransFuse: Fusing transformers and cnns for medical image segmentation. *arXiv* **2021**, arXiv:2102.08005. [CrossRef]
35. Ji, S.; Wei, S.; Lu, M. Fully convolutional networks for multi-source building extraction from an open aerial and satellite imagery dataset. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 574–586. [CrossRef]

36. Mnih, V. Machine Learning for Aerial Image Labeling. Ph.D. Dissertation, Department Computer Science, Toronto University, Toronto, ON, Canada, 2013.
37. Mittal, S.; Tatarchenko, M.; Brox, T. Semi-supervised semantic segmentation with high- and low-level consistency. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1369–1379. [CrossRef]
38. He, Y.; Wang, J.; Liao, C.; Shan, B.; Zhou, X. ClassHyPer: ClassMix-Based Hybrid Perturbations for Deep Semi-Supervised Semantic Segmentation of Remote Sensing Imagery. *Remote Sens.* **2022**, *14*, 879. [CrossRef]
39. Souly, N.; Spampinato, C.; Shah, M. Semi Supervised Semantic Segmentation Using Generative Adversarial Network. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5689–5697. [CrossRef]
40. Zhang, J.; Li, Z.; Zhang, C.; Ma, H. Robust Adversarial Learning for Semi-Supervised Semantic Segmentation. In Proceedings of the IEEE International Conference on Image Processing, Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 728–732. [CrossRef]
41. Sun, X.; Shi, A.; Huang, H.; Mayer, H. BAS4Net: Boundary-aware semi-supervised semantic segmentation network for very high resolution remote sensing images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 5398–5413. [CrossRef]
42. Luo, H.; Chen, C.; Fang, L.; Zhu, X.; Lu, L. High-resolution aerial images semantic segmentation using deep fully convolutional network with channel attention mechanism. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3492–3507. [CrossRef]
43. Huang, J.; Zhang, X.; Sun, Y.; Xin, Q. Attention-guided label refinement network for semantic segmentation of very high resolution aerial orthoimages. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4490–4503. [CrossRef]
44. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, L.A.; Zhou, Y. TransUNet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306.
45. Hou, Q.; Zhang, L.; Cheng, M.; Feng, J. Strip Pooling: Rethinking Spatial Pooling for Scene Parsing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4002–4011. [CrossRef]
46. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241. [CrossRef]
47. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. Available online: https://arxiv.org/abs/1412.6980 (accessed on 22 December 2014).
48. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective Kernel Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, NY, USA, 15–20 June 2019; pp. 510–519. [CrossRef]