remote sensing

MDPI

Article

MSST-Net: A Multi-Scale Adaptive Network for Building Extraction from Remote Sensing Images Based on Swin Transformer

Wei Yuan [1,2,*] and Wenbo Xu [3]

[1] School of Architecture and Civil Engineering, Chengdu University, Chengdu 610106, China
[2] Key Laboratory of Pattern Recognition and Intelligent Information Processing, Institutions of Higher Education of Sichuan Province, Chengdu University, Chengdu 610106, China
[3] School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China; xuwenbo@uestc.edu.cn
* Correspondence: yuanwei@cdu.edu.cn

Abstract: The segmentation of remote sensing images by deep learning technology is the main method for remote sensing image interpretation. However, the segmentation model based on a convolutional neural network cannot capture the global features very well. A transformer, whose self-attention mechanism can supply each pixel with a global feature, makes up for the deficiency of the convolutional neural network. Therefore, a multi-scale adaptive segmentation network model (MSST-Net) based on a Swin Transformer is proposed in this paper. Firstly, a Swin Transformer is used as the backbone to encode the input image. Then, the feature maps of different levels are decoded separately. Thirdly, the convolution is used for fusion, so that the network can automatically learn the weight of the decoding results of each level. Finally, we adjust the channels to obtain the final prediction map by using the convolution with a kernel of 1×1. By comparing this with other segmentation network models on a WHU building data set, the evaluation metrics, mIoU, F1-score and accuracy are all improved. The network model proposed in this paper is a multi-scale adaptive network model that pays more attention to the global features for remote sensing segmentation.

Keywords: deep learning; remote sensing; transformer; semantic segmentation; multi-scale adaptive

Citation: Yuan, W.; Xu, W. MSST-Net: A Multi-Scale Adaptive Network for Building Extraction from Remote Sensing Images Based on Swin Transformer. *Remote Sens.* **2021**, *13*, 4743. https://doi.org/10.3390/rs13234743

Academic Editors: Mercedes E. Paoletti and Juan M. Haut

Received: 27 September 2021
Accepted: 22 November 2021
Published: 23 November 2021

1. Introduction

The traditional semantic segmentation of remote sensing images mainly uses a certain algorithm to transform the spectral feature of the image, set a threshold, and then classify pixels with similar values into one category [1–3]. However, different objects may have similar spectral values, the same objects may have different spectral values, and the search for the optimal threshold lacks theoretical support. Therefore, the classification effect of such methods is not acceptable. After machine learning technology was proposed, some scholars used random forest [4] and support vector machine [5] algorithms for the semantic segmentation of remote sensing images, and achieved certain results. However, the input features of machine learning methods are extracted manually, which is labor-intensive.

Deep learning, which can extract the features of an image automatically, has gradually replaced traditional machine learning methods. The LeNet5 [6] proposed by Lecun et al. uses a convolutional neural network to replace partially full connections, which reduces the number of parameters and speeds up the solution. The VGGNet [7] model proposed by Simonyan increases the depth of the network and achieves better performance, but it has a fully connected layer, so it can only be used for image classification. The FCN [8] proposed by Long removes the fully connected layer, adopts the convolution layers completely, and outputs a map with the same size of input image through up sampling. This is an actual pixel-wise network of semantic segmentation. However, the accuracy of segmentation is not high; because this method has only one layer for upsampling to the input image

size, much information is lost in the decoding. Unlike the FCN, the UNet [9] proposed by Ranneberger up samples step by step, and integrates the downsampling feature of the same scale in each step, so the accuracy of segmentation is significantly improved. SegNet [10], proposed by Badrinarayanan, is similar to UNet. The idea is to remember the index of the maximum pool during down sampling, only recovering the data of the index position during up sampling, and fill the rest with zero. In the same year, Zhao proposed a pyramid pooling module named PSPNet [11], which improves the ability of the network to capture global feature information by aggregating the contexts of different regions. Zhou et al. [12] proposed Unet++, which integrates different scales' features. DeepLabV1 [13] used atrous convolution to increase the receptive field. DeepLabV2 [14] proposed an atrous spatial pyramid pooling (ASPP), and DeepLabV3 [15] further improved the atrous spatial pyramid pooling.

1.1. Research on Segmentation of Remote Sensing Image by Deep Learning

Many scholars have applied deep learning technology to remote sensing image segmentation. Yuan [16] designed a simple structure network of semantic segmentation that integrates activation from multiple layers, and introduced the signed distance function of building boundaries as part of the loss function. Bischke [17] introduced a new multi-task loss, which leverages multiple output representations of the segmentation mask; the loss function requires the network to pay more attention to the pixels near the boundary. Zhong et al. [18] extracted roads and buildings using the FCN network from areal images of the Massachusetts road data set [19], and good performance was achieved. Panboonyuen et al. [20] improved the effect of road segmentation by using the SegNet network and ELU activation function. Wei et al. [21] improved the accuracy of road extraction by enhancing the weights of pixels near the label in the cross entropy. Mattyus et al. [22] proposed a variant of FCN that uses ResNet as a backbone and a fully deconvolutional decoder to extract roads with good topology in aerial images. Gao et al. [23] proposed an end-to-end network called the multiple feature pyramid network, which is similar to the RSRCNN network. They made use of the multi-level features of HRSI and designed a new loss function that focuses more on imbalanced classes. Zhang [24] proposed a deep residual UNet for road extraction, which has the advantages of both residual learning and UNet. Xu [25] proposed a GL-Dense-UNet to extract roads from aerial images. D-LinkNet [26], the best solution in Deepglobe 2018 [27], is a UNet-like network with a dilation part in the center. A method that uses weakly supervised learning for road extraction was proposed at the global scale [28]. Wu [29] proposed a method for road extraction that uses only weakly labeled data. Yuan [30] proposed a neighborhood-based loss function that gives different weights to pixels according to the consistency of prediction results between pixels and eight neighborhood pixels; it offers a significant improvement in the house extraction of remote sensing images.

1.2. Related Works

The above deep learning segmentation networks for remote sensing images are all based on a convolutional neural network. A convolutional neural network can only pay attention to a small range of neighborhood features, resulting in insufficient attention being paid to global features. A transformer [31] only makes up for the shortcomings of the convolutional neural network; its unique self-attention mechanism causes each pixel to contain the global feature. Alexey [32] proposed the ViT network model, which adapted the transformer from natural language processing to image recognition. ViT directly divides the input image into fixed-size patches and inputs them as vectors into the transformer, but it can only be used for classification tasks. Zheng [33] proposed a network called SETR, which reshapes the output of the transformer from vectors into an image. The upsampling is then carried out for decoding by deconvolution to get the segmentation result. However, the transformer has its own defect; because it is a vector operation using all pixels, operations on large-size images will be very time-consuming

and memory-consuming. The Swin Transformer [34] was designed to solve this problem. It divides the input image into several patches, and each pixel only undergoes a transformer operation with the pixels in the same patch. In this way, the number of operations is greatly reduced. In this way, there is no communication between patches, and pixels cannot capture wider ranges of information. Therefore, the Swin Transformer proposes a concept called shift window, which changes the second step patches so as to realize communication between the patches. At the same time, the Swin Transformer adopts a downsampling architecture, which reduces the amount of computation and increases the receptive field of pixels. This hierarchical architecture obtains features from different levels. However, most of the existing networks use multi-scale feature information via deep supervision, or only use the final abstract feature. As a result, multi-level features cannot be fully used. Therefore, we propose a multi-scale feature fusion network model based on the Swin Transformer. The network model can automatically learn the fusion weights of levels to realize multi-scale adaptation.

1.3. Contributions and Highlights of This Paper

The main highlights of this paper are as follows:

(a) A multi-scale adaptive network model named MSST-Net (Multi-Scale Swin Transformer) is proposed. Compared with single-scale and other CNN segmentation network models on the WHU building data set, MSST-Net is better;
(b) The Swin Transformer encoding architecture based on a transformer is applied to the semantic segmentation of remote sensing images, which is better than SETR that is also based on a transformer;
(c) We compare the segmentation results of different patch window sizes in the Swin Transformer.

The next section presents the materials and the proposed network, MSST-Net. Section 3 explains the evaluation metrics of experiments in this paper. Section 4 details the experiment and results. Finally, in Section 5, we present conclusions from this paper and the direction of future work.

2. Materials and Methods

2.1. Data Set and Preprocessing

The experimental data were based on the WHU building open data set [35], which is edited manually by the team of Professor Ji Shunping at Wuhan University. The data set includes two parts: one is an aerial image data set, the other is a satellite image data set. The satellite image data set consists of two subsets. One of them was collected from cities worldwide and from various remote sensing resources, including QuickBird, Worldview series, IKONOS, ZY-3, etc. The other satellite building sub-data set consists of 6 neighboring satellite images covering 550 km^2 in East Asia with 2.7 m ground resolution. The aerial image data set consists of more than 220,000 independent buildings extracted from aerial images with 0.075 m spatial resolution and covering 450 km^2 in Christchurch, New Zealand. The method proposed in this paper mainly aims at the segmentation problem of houses of different sizes in the remote sensing image; there were different-sized houses in both aerial images and satellite images. Therefore, as long as one data set is selected, the effectiveness of the method in this paper can be proven. The ground resolution of the aerial image was higher, and was the first choice in actual production. For the purpose of serving production, this article selected an aerial image data set. Original aerial data were downloaded from the New Zealand Land Information Services website and downsampled from 0.075 to 0.3 m ground resolution to reduce the amount of data, and then cropped into 8189 tiles with 512 $\times$ 512 pixels. In order to train and evaluate the network model, the tiles were divided into three parts. The training data set consisted of 4736 images (including 130,500 buildings), the validation data set consisted of 1036 images (including 14,500 buildings), and the test data set consisted of 2416 images (including 42,000 buildings). The format of all tiles was TIF, the number of channels was 3, and the corresponding label

was TIF format data with one channel. As shown in Figure 1, the training data set is in the blue rectangle, the validation data set is in the orange rectangle, and the test data set is in the red rectangle.

Figure 1. WHU building data set map.

In order to speed up the data loading, we used software written in Python to zip the respective training, validation, and test data sets into TFrecord, which is the official format of TensorFlow.

The data set augmentation method has been used with less training data [36]. However, few-shot learning was the development direction of the deep learning algorithm [37]. Moreover, the WHU building data set was very large, compared with other remote sensing data sets, such as ISPRS Vaihingen, and the same training and test data (no augmentation) were convenient for use by other scholars to compare their networks. Therefore, we did not use data set augmentation in this paper.

2.2. Transformer and Swin Transformer

The core concept of the transformer is the self-attention mechanism. Firstly, three trainable weight matrices Q, K, and V are set. After the input vector is multiplied by the Q matrix, the query vector is obtained; after the input vector is multiplied by the K matrix, the key vector is obtained; and after the input vector is multiplied by the V matrix, the value vector is obtained. Then, the query vector of each pixel and the key vector of all pixels are made inner products, respectively, and the result is normalized to the probability distribution using the softmax function, where probability represents the weight of the correlation between one pixel and another pixel. Finally, the weight is multiplied by the value vector of the corresponding pixel to obtain the output value. Each pixel will operate with all other pixels of the input image in self-attention, so each pixel contains the global feature. At the same time, because the matrix operation is adopted in self-attention, the computer calculation is parallel, and the efficiency of the calculation is higher than convolution.

Although the transformer pays attention to the global feature, if the image size is very large, the sum of all pixel information will contain a lot of useless information, and it will cause information redundancy and an inability to learn useful information effectively. Moreover, local features are more important for the segmentation of boundaries or small targets, and each pixel needs to be calculated with all other pixels in the image. The amount of calculation is very large, and the memory consumed is also very large. This cannot be deployed on an ordinary computer for model training. For the above reasons, the Swin Transformer was proposed. The key theory of the Swin Transformer is to divide

the image into several patches, and each pixel only performs the transformer operation in its own patch, such that pixels pay more attention to local features, and the amount of operation is greatly reduced. In order to enable pixels to capture information beyond a patch, the shift window method is used to divide patches with different boundaries in the second transformer operation. At the same time, the Swin Transformer uses a hierarchical architecture of downsampling layer by layer, which gradually increases the pixel reception field and provides multi-scale feature information.

2.3. MSST-Net Methodology

In contrast to the semantic segmentation of other images, such as medical CT images, the sizes of houses in remote sensing images are not uniform, and it is difficult to segment all houses by the feature of a single scale. As shown in Figure 2, the house within the red circle is relatively small, and the pixels where the house is located can be correctly segmented only by capturing the pixel information not far from it. If the captured range is too large, it will cause information redundancy, and this redundant information will inhibit the effective information around the house. However, the house within the green rectangle is relatively large, and it is difficult to judge the type of pixels by capturing only the local features in a small range—a bigger range is needed. Therefore, the merging of multi-scale features can improve the semantic segmentation of houses of different sizes in a remote sensing image.

Figure 2. Differently sized houses in the remote sensing image.

In this paper, we used the Swin Transformer as the encoder to automatically extract the features of the input image. The deeper the level, the more abstract the extracted features, as shown in the red dotted rectangle in Figure 3. First, the input image was divided into a 2 × 2-sized patch; each patch was reshaped into a vector, and learnable location information parameters were embedded to record the location of each patch. The stage1 module contained two transformer blocks: the first was the regular window and the second was the shift window, whose principle is shown in Figure 4, so as to ensure the information outside the patch was captured. After the stage1 module, the size of the output feature was $H/2 \times W/2 \times 64$, which means the height and width were half of the input image and the number of channels was 64. Similar to stage1, the stage2

module also contained two transformer blocks. There was a module named patch merging in the front of stage2. The function of this module, similar to pooling in CNN, was to downsample before the start of each stage to reduce the resolution and adjust the number of channels, forming a hierarchical architecture. At the same time, this also saved the cost of computation. Specifically, pixels were selected at intervals of 2 in the row and column directions. Then, they were concatenated, and the channel dimension quadrupled (because H and W were each reduced 2 times). At this time, the channel dimension was adjusted to twice the previous number through a full connection layer. Because patch merging was used in the head of stage2, the size of the output feature after the stage2 module was $H/4 \times W/4 \times 128$, which means that the height and width were one-quarter of the input image, and the number of channels changed to 128. The stage3 module had six transformer blocks. The regular windows and shift windows were arranged alternately, equivalent to three times that of stage1. Similarly, patch merging was used in the head. Therefore, after the stage3 module, the size of the output feature was $H/8 \times W/8 \times 256$, meaning that the height and width were each one-eighth of the input image, and the number of channels changed to 256. Stage4 had the same structure as stage2, so the encoded feature mapping size was $H/16 \times W/16 \times 512$, meaning the height and width were each one-sixteenth of the input image, and the number of channels changed to 512. Many networks, such as PSPNet, directly use the final output of the backbone as the encoding result, but this did not employ multi-scale information by directly using the output results of stage4 for decoding, which meant changing from the high-level feature of the input image to the prediction result. In fact, all different levels of feature maps have an impact on the segmentation results. For example, high-level features contain mainly the location information of the segmentation object, while low-level features contain more edge information of the segmentation object. In order to segment the object more accurately, we should make full use of the feature maps at all levels. There are also networks using deep supervision to take advantage of the multi-scale features of the backbone, such as UNet++, but all its losses calculated by different scale features are directly averaged as the final loss. This simple fusion method will suppress valid feature information using invalid feature information.

In order to make full use of feature information of different levels, we propose a multi-scale feature merging method. Firstly, the respective output features of stage1 to stage4 were decoded. The decoding method of upsampling the output feature to double size was performed; the result was concatenated with the same size feature of the Swin Transformer, and then two convolutions with a kernel of 3×3 were performed, halving the number of channels at the same time. After that, upsampling to double the size was performed again; the result was concatenated with the same size feature of the Swin Transformer, and then two convolutions with a kernel of 3×3 were performed, halving the number of channels at the same time. This loop continued until the size of the output result was the same as the original image, and at that time, the number of output channels was 32. Secondly, the output results from stage1 to stage4 were concatenated to obtain the output result with 128 channels. At this time, the multi-level decoding results were concatenated together. For the better merging of multi-level decoding results, we performed two convolutions with a kernel of 3×3, so that the network automatically learned the weight of the decoding results at all levels and reduced the number of channels to 32 at the same time. Finally, the number of channels was adjusted to the class number via convolution with a kernel of 1×1, so that the network could further automatically learn the merging parameters. The decoding architecture is shown by the blue dotted rectangle in Figure 3.

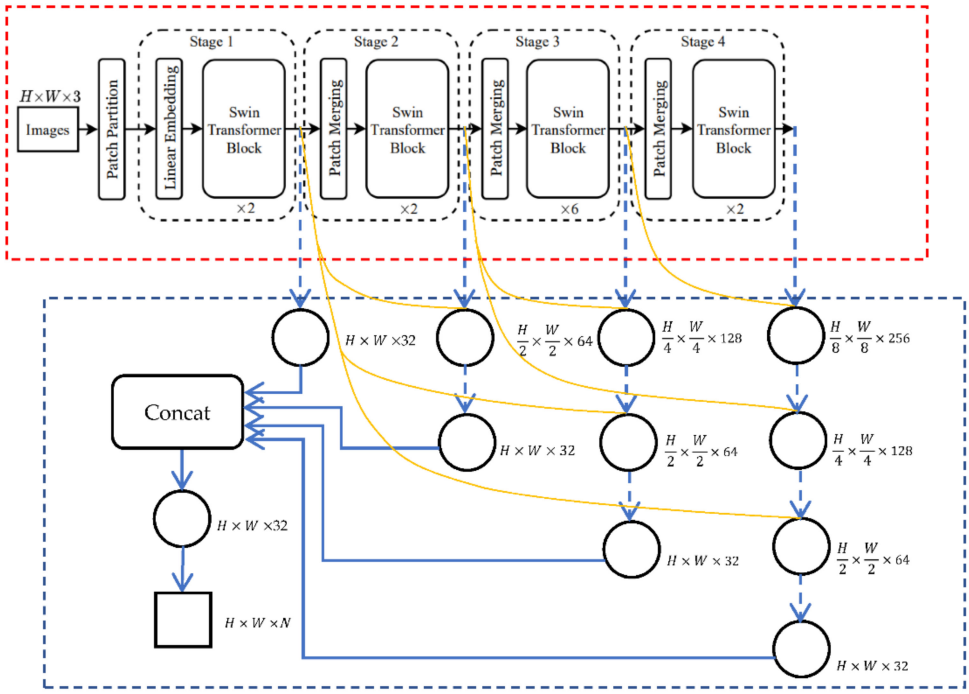

Figure 3. MSST-Net architecture. The red dotted rectangle is the encoding architecture of the Swin Transformer. The blue dotted rectangle is the decoding architecture of multi-level merging. The circle in the figure represents twice convolution with a kernel of 3×3, to the right of the circle is its output dimension, H represents the height of input image, W represents the width of the input image, the last number is the number of channels. The square represents single convolution with a kernel of 1×1, to the right of the square is its output dimension where N represents class number. The downward dashed arrow represents the deconvolution of upsampling to double size. The yellow curve represents a jump connection.

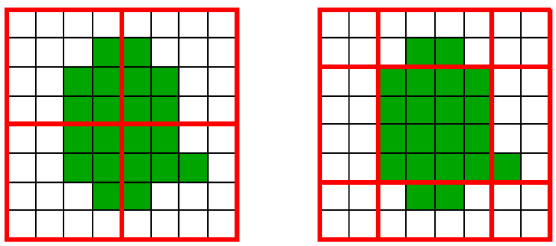

Figure 4. Regular window and shift window of Swin Transformer in the Swin Transformer block [34]. Left is the regular window, and the red windows divide the input image into four equal patches. Right is the shift window, which is formed by scrolling the regular window right and down by 2 pixels, so that the pixels divided into one patch are not the same as in the patch of the regular window.

3. Evaluation Metrics

In order to verify the advantages of our network objectively, three evaluation metrics—mean intersection over union (mIoU), accuracy, and F1-score—were used for comparative analysis.

The formula of mIoU is:

$$\text{mIoU} = \frac{1}{N+1}\sum_{i=0}^{N}\frac{TP}{TP+FN+FP} \quad (1)$$

The formula of accuracy is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

The formula of the F1-score is:

$$\text{F1} - \text{Score} = \frac{2 \times Precison \times Recall}{Precison + Recall} \tag{3}$$

where *Precision* and *recall* are

$$Precison = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

In all formulas, N represents the number of foreground classes; plus the background, the total class number is $N + 1$. In this paper, N is taken as 1. TP represents true positive, which is the number of foreground pixels correctly predicted as foreground. FP represents false positive, which is the number of background pixels mispredicted as foreground. TN represents true negative, which is the number of background pixels correctly predicted as background. FN represents false negative, which is the number of foreground pixels mispredicted as background. In Formula 1, houses and other buildings are taken as a positive class to get the evaluation metrics, and their average is the mIoU. The positive classes in Equations (2)–(5) are houses.

4. Experiment and Results

4.1. Hardware and Software

The CPU was an Intel I5-9400F, the GPU was an NVIDIA Geforce RTX 2060 super 8G, and the CUDA10.0 used was the GPU acceleration library. TensorFlow version 1.14.0 was chosen as the deep learning framework. The environment in which the code was written and run was Python 3.6.8.

The AdamOptimizer [38] is the most widely used method, so it was chosen in this paper, and the learning rate was 0.0001. L2 regularization [39] was used to prevent over fitting and improve generalization ability. The total loss is shown in Formula 6. The maximum number of epochs for the training was set to 100, and after each epoch, evaluation was performed on the validation data set. If the metrics no longer increased for ten consecutive epochs, the early stop strategy was used to end the training and avoid excessive model training.

$$\left.\begin{array}{c} \text{TotalLoss} = \text{CrossEntropy} + L2 \\ L2 = ||w||_2^2 = \sum_i |w_i^2| \end{array}\right\} \tag{6}$$

4.2. Results and Discussion

The evaluation metric results of the main segmentation networks for the test data set of the WHU building data set are shown in Table 1. From the table, it can be seen that our network based on the multi-scale Swin Transformer was state of the art in all three evaluation metrics.

Among the semantic segmentation networks based on the convolutional neural network—DeepLabV3, PSPNet, and SegNet—the best performing was the PSPNet network, the worst performing was the SegNet network, and DeepLabV3 was in the middle. The mIoU of PSPNet was 3.8% higher than SegNet, the F1-score of PSPNet was 4.3% higher than SegNet, and the accuracy of PSPNet was 1.1% higher than SegNet. Among the networks based on the transformer—MSST-Net, SETR, and Swin Transformer (only the stage4 output result in Swing Transformer was used for decoding)—MSST-Net performed best, SETR performed worst, and the Swin Transformer was in the middle. The mIoU of

MSST-Net was 5.2% higher than that of SETR, the F1-score of MSST-Net was 6.0% higher than that of SETR, and the accuracy of MSST-Net was 1.3% higher than that of SETR. In the experiment, the window sizes of MSST-Net and Swin Transformer were 8 × 8 pixels.

Table 1. Metrics of the WHU data set.

Methods	mIoU	F1-Score	Accuracy
SegNet	82.1	81.5	95.8
DeepLabV3	84.6	84.3	96.7
PSPNet	85.9	85.8	96.9
SETR	82.8	82.2	96.1
Swin Transformer (only stage4, winSize = 8)	86.7	86.7	97.1
MSST-Net (winSize = 8)	88.0	88.2	97.4

The multi-scale MSST-Net was better than the Swin Transformer using a single scale. mIoU increased by 1.3%, F1-score increased by 1.5%, and accuracy increased by 0.3%. These results show that the merging of multi-scale feature information can better segment the houses in remote sensing images.

MSST-Net, which was the best of all networks using the transformer, performed better than PSPNet, which was the best of all networks using convolution. mIoU was 2.1% higher, F1-score was 2.4% higher, and accuracy was 0.5% higher. SETR, which was the worst of all networks using the transformer, was still better than SegNet, which was the worst of all networks using convolution. mIoU was 0.7% higher, F1-score was 0.7% higher, and accuracy was 0.3% higher. Therefore, transformer-based networks showed more advantages in semantic segmentation than convolution-based networks.

It can be seen from the prediction image of the networks used on the test set that the houses in the prediction image based on the convolutional neural network had curved edges and strong connectivity, and there were few unconnected pixels (house composed of a single or few pixels), as shown in the green circle of Figure 5. Therefore, most of the pixels at the corners were predicted in error, because most house edges are straight-line, but the advantage was that there were few discrete prediction error points. This is because the convolutional neural network only paid attention to the neighborhood pixels; the receptive field of the edge pixels was too small to capture a wider range of features, and the predicted value of the edge pixels was largely affected by the neighborhood pixels' value.

Compared to the convolutional neural network, in the neural network prediction image based on the transformer, the edges of the houses had straighter lines, and the right angles were more obvious, as shown in the green circles in Figure 6. However, in the neural network prediction image based on the transformer, there were some unconnected pixels (houses composed of a single or few pixels). This was particularly obvious in the prediction images of SETR, as shown in the red circle in the SETR images in Figure 6. This is because the transformer used by SETR only captured global features. In this way, the feature information of the neighborhood was flooded, resulting in different prediction results for nearby pixels. In the MSST-Net prediction image, this kind of situation was significantly reduced, as shown in the red circle in the MSST-Net images in Figure 6. This was because MSST-Net merged the multi-scale features extracted by the Swin Transformer. It judged the class of pixels not only from the global features, but also assessed the local features of various scales, so the classification was more accurate.

There is a hyper-parameter—window size—in the Swin Transformer. This parameter is actually the range of the transformer calculation and the range of captured features. The larger the window size, the more features can be captured. In order to analyze the relationship between window size and segmentation accuracy, we conducted experiments with window sizes of 4, 8 and 16. The results are shown in Table 2. It can be seen that when the window size was 8, the metrics were the best, and when the window size was 4, the metrics were the worst. Therefore, the larger range of features was not necessarily better, but it showed a change trend of improving first and then getting worse. If the range is too

small, the network can only detect local features, and cannot capture global features. If the range is too large, the useful local feature information will be flooded by a large amount of useless information, which will have a negative impact on the segmentation accuracy.

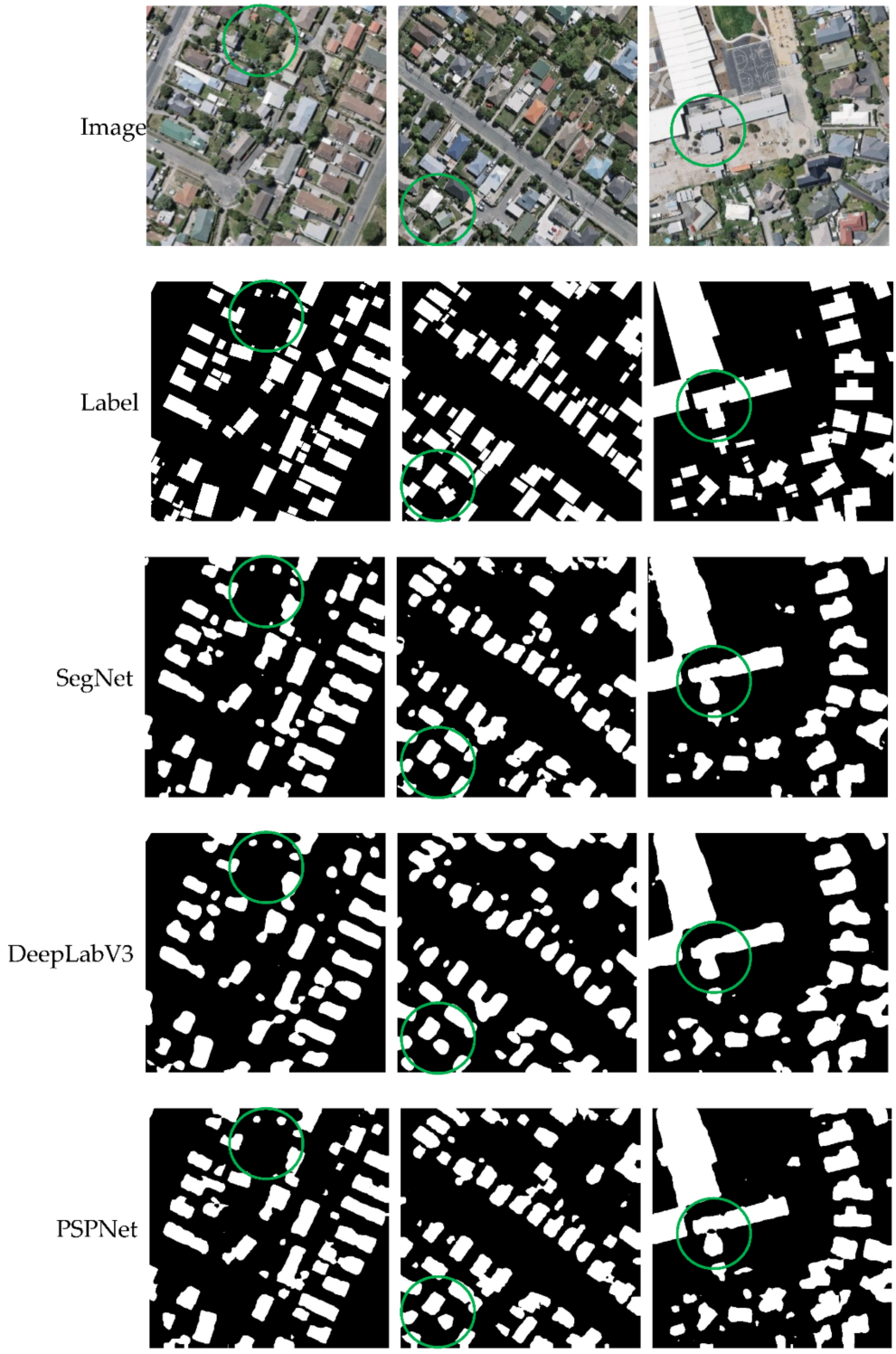

Figure 5. Comparison of prediction results by networks based on CNN.

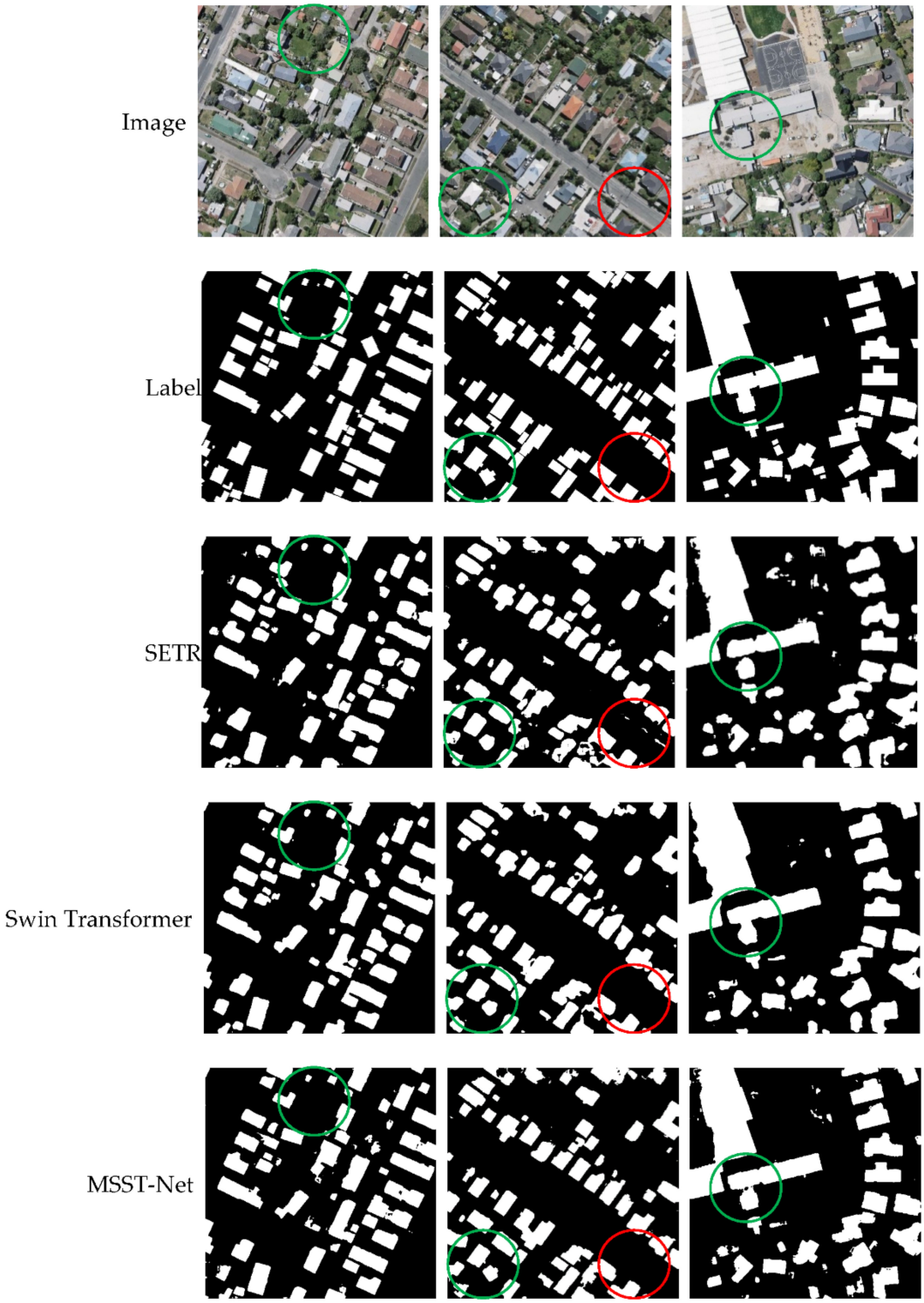

Figure 6. Comparison of prediction results by network based on transformer.

Table 2. Comparison of different window sizes of MSST-Net.

Window Size	mIoU	F1-Score	Accuracy
MSST-Net (winSize = 4)	86.0	85.9	97.0
MSST-Net (winSize = 8)	88.0	88.2	97.4
MSST-Net (winSize = 16)	87.4	87.5	97.2

In order to prove the benefits of our multi-scale fusion method, we performed an ablation experiment. Each respective stage of the MSST-Net network was used as the decoder, and all stages were used to decode in the form of deep supervision. The results are compared with the results of our proposed multi-scale fusion method in Table 3.

Table 3. Ablation test.

Window Size	mIoU	F1-Score	Accuracy
Only Stage 1	80.9	79.7	95.9
Only Stage 2	84.5	84.2	96.6
Only Stage 3	86.8	86.9	97.1
Only Stage 4	86.7	86.7	97.1
Stage1–4 with deep supervision	84.1	83.9	96.3
Stage1–4 with our method	88.0	88.2	97.4

It can be seen that from stage1 to stage4, with the increase in network depth, the evaluation metric values did not always increase, but first increased and then decreased. This shows that the deeper network was not necessarily better. When the network is shallow, the data cannot be well fitted, so the evaluation metric is poor. However, if the network is too deep, it will lead to over fitting, which will reduce the generalization of the data and worsen the evaluation metrics. Furthermore, the results of using stage1–stage4 with deep supervision were not good—they were even worse than stage 2 and only better than stage 1, which showed that although the deep supervision method made up for the poor stage, it inhibited the effect of the better stage. The results of the fusion method proposed by us are better than those for the best stage, which shows that, instead of merely finding the best stage, our multi-scale fusion method made full use of the advantages of multiple stages to achieve better results than any single stage.

5. Conclusions and Future Work

In this paper, we proposed a multi-scale adaptive semantic segmentation network called MSST-Net, with a Swin Transformer as the backbone. It was compared with SETR, the non-multi-scale Swin Transformer, and CNN-based networks DeepLabV3, PSPNet, and SegNet.

The results show that SETR, the worst network, and MSST-Net, the best network, both of which are based on a transformer, performed better than the worst and the best networks based on CNN, respectively. Moreover, our MSST-Net network achieved the best performance of all semantic segmentation networks. At the same time, we also compared different window sizes in MSST-Net. The results show that larger or smaller window sizes are not necessarily better, but with increases in window size, the accuracy becomes first better, and then worse, so an optimal window size must be set.

It can be seen that both MSST and SETR used CNN in the decoding stage. In the future, we will explore replacing CNN with a transformer in the decoding stage, so as to better capture the global information in the decoding stage and improve the segmentation accuracy of houses of different sizes in remote sensing images.

Author Contributions: W.Y. wrote the manuscript and designed the comparative experiments; W.X. supervised the study and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the open fund of Key Laboratory of Pattern Recognition and Intelligent Information Processing, Institutions of Higher Education of Sichuan Province, Chengdu University (MSSB-2021-04).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are open data sets. The data sets can be downloaded from https://gpcv.whu.edu.cn/data/building_dataset.html (accessed on 20 June 2020).

Acknowledgments: We would like to thank the anonymous reviewers for their constructive and valuable suggestions on the earlier drafts of this manuscript.

Conflicts of Interest: The authors declare that there is no conflict of interest.

References

1. Kauth, R.J.; Thomas, G.S. The tasselled-cap—A graphic description of the spectral-temporal development of agricultural crops as seen by landsat. In Proceedings of the Symposium on Machine Processing of Remotely Sensed Data, West Lafayette, IN, USA, 29 June 1976; pp. 41–51.
2. Baatz, M.; Schape, A. Multiresolution segmentation: An optimization approach for high quality multi-scale image segmentation. In Proceedings of the Beiträge Zum AGIT-Symposium, Karlsruhe, Germany, 1 January 2000; pp. 2–23.
3. Gaetano, R.; Masi, G.; Poggi, G.; Verdoliva, L.; Scarpa, G. Marker-controlled watershed-based segmentation of multiresolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 2987–3004. [CrossRef]
4. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
5. Vapnik, V.; Chervonenkis, A. A note on class of perceptron. *Autom. Remote Control* **1964**, *24*, 112–120.
6. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–14. Available online: https://arxiv.org/abs/1409.1556 (accessed on 3 July 2021).
8. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 7–12 June 2015; pp. 3431–3440.
9. Ronneberger, O.; Fischer, P.; Brox, T. Convolutional networks for biomedical image segmentation. In Proceedings of the 2015 Medical Image Computing and Computer Assisted Intervention, Piscataway, NJ, USA, 5–9 October 2015; pp. 234–241.
10. 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] [PubMed]
11. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239. [CrossRef]
12. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* **2020**, *39*, 1856–1867. [CrossRef] [PubMed]
13. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *Comput. Sci.* **2014**, 357–361. Available online: https://arxiv.org/abs/1412.7062 (accessed on 13 July 2021).
14. Chen, L.; 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] [PubMed]
15. Rethinking Atrous Convolution for Semantic Image Segmentation. 2017. Available online: https://arxiv.org/abs/1706.05587 (accessed on 13 July 2021).
16. Yuan, J. Learning building extraction in aerial scenes with convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 2793–2798. [CrossRef] [PubMed]
17. Bischke, B.; Helber, P.; Folz, J.; Borth, D.; Dengel, A. Multi-task learning for segmentation of building footprints with deep neural networks. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1480–1484. [CrossRef]
18. Zhong, Z.; Li, J.; Cui, W.; Jiang, H. Fully convolutional networks for building and road extraction: Preliminary results. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 1591–1594. [CrossRef]
19. Mnih, V. *Machine Learning for Aerial Image Labeling*; Department of Computer Science, University of Toronto: Toronto, ON, Canada, 2013.
20. Panboonyuen, T.; Vateekul, P.; Jitkajornwanich, K.; Lawawirojwong, S. An enhanced deep convolutional encoder-decoder network for road segmentation on aerial imagery. In Proceedings of the International Conference on Computing & Information Technology, Bangkok, Thailand, 6–7 July 2017; Springer: Cham, Switzerland, 2017; pp. 191–201.
21. Wei, Y.; Wang, Z.; Xu, M. Road Structure Refined CNN for Road Extraction in Aerial Image. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 709–713. [CrossRef]
22. Máttyus, G.; Luo, W.; Urtasun, R. Deeproadmapper: Extracting Road Topology from Aerial Images. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3458–3466. [CrossRef]
23. Gao, X.; Sun, X.; Zhang, Y.; Yan, M.; Xu, G.; Sun, H.; Jiao, J.; Fu, K. An End-to-End Neural Network for Road Extraction From Remote Sensing Imagery by Multiple Feature Pyramid Network. *IEEE Access* **2018**, *6*, 39401–39414. [CrossRef]
24. Zhang, Z.; Liu, Q.; Wang, Y. Road extraction by deepresidual U-Net. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 749–753. [CrossRef]

25. Xu, Y.; Xie, Z.; Feng, Y.; Chen, Z. Road extraction from high resolution remote sensing imagery using deep learning. *Remote Sens.* **2018**, *10*, 1461. [CrossRef]
26. Zhou, L.; Zhang, C.; Wu, M. D-Linknet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 192–196.
27. Deep Globe. 2018. Available online: http://deepglobe.org/ (accessed on 10 July 2021).
28. Bonafilia, D.; Gill, J.; Basu, S.; Yang, D. Buildinghighresolutionmaps for humanitarian aid and development with weakly-and semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–20 June 2019; pp. 1–9.
29. Wu, S.; Du, C.; Chen, H.; Xu, Y.; Guo, N.; Jing, N. Road extraction from very high resolution images using weakly labeled OpenStreetMap centerline. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 478. [CrossRef]
30. Yuan, W.; Xu, W. NeighborLoss: A Loss Function Considering Spatial Correlation for Semantic Segmentation of Remote Sensing Image. *IEEE Access* **2021**, *9*, 75641–75649. [CrossRef]
31. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 6000–6010.
32. 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 16x16 Words: Transformer s for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
33. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.S.; et al. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. *arXiv* **2020**, arXiv:2012.15840.
34. 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.
35. Ji, S.P.; Wei, S.Q. Building extraction via convolutional neural networks from an open remote sensing building dataset. *Acta Geod. Cartogr. Sin.* **2019**, *48*, 448–459. (In Chinese)
36. Lashgari, E.; Liang, D.; Maoz, U. Data augmentation for deep-learning-based electroencephalography. *J. Neurosci. Methods* **2020**, *346*, 108885. [CrossRef] [PubMed]
37. Liu, W.; Zhang, C.; Lin, G.; Liu, F. CRNet: Cross-reference networks for few-shot segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4164–4172. [CrossRef]
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
39. Hui, J.; Qin, Q.; Xu, W.; Sui, J. Instance segmentation of buildings from high-resolution remote sensing images with multitask learning. *Acta Sci. Nat. Univ. Pekin.* **2019**, *55*, 1067–1077.