International Journal of
Geo-Information

Article

Multi-Resolution Transformer Network for Building and Road Segmentation of Remote Sensing Image

Zhongyu Sun [1,2], Wangping Zhou [1,2,*], Chen Ding [2] and Min Xia [1,2]

[1] Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; 201913360009@nuist.edu.cn (Z.S.); xiamin@nuist.edu.cn (M.X.)

[2] Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; 20181223015@nuist.edu.cn

* Correspondence: wpzhou@nuist.edu.cn

Abstract: Extracting buildings and roads from remote sensing images is very important in the area of land cover monitoring, which is of great help to urban planning. Currently, a deep learning method is used by the majority of building and road extraction algorithms. However, for existing semantic segmentation, it has a limitation on the receptive field of high-resolution remote sensing images, which means that it can not show the long-distance scene well during pixel classification, and the image features is compressed during down-sampling, meaning that the detailed information is lost. In order to address these issues, Hybrid Multi-resolution and Transformer semantic extraction Network (HMRT) is proposed in this paper, by which a global receptive field for each pixel can be provided, a small receptive field of convolutional neural networks (CNN) can be overcome, and the ability of scene understanding can be enhanced well. Firstly, we blend the features by branches of different resolutions to keep the high-resolution and multi-resolution during down-sampling and fully retain feature information. Secondly, we introduce the Transformer sequence feature extraction network and use encoding and decoding to realize that each pixel has the global receptive field. The recall, F1, OA and MIoU of HMPR obtain 85.32%, 84.88%, 85.99% and 74.19%, respectively, in the main experiment and reach 91.29%, 90.41%, 91.32% and 84.00%, respectively, in the generalization experiment, which prove that the method proposed is better than existing methods.

Keywords: segmentation; high resolution; transformer; deep learning

check for updates

Citation: Sun, Z.; Zhou, W.; Ding, C.; Xia, M. Multi-Resolution Transformer Network for Building and Road Segmentation of Remote Sensing Image . *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 165. https://doi.org/10.3390/ijgi11030165

Academic Editors: Gloria Bordogna, Cristiano Fugazza and Wolfgang Kainz

Received: 9 January 2022
Accepted: 23 February 2022
Published: 25 February 2022

1. Introduction

Land resources have the following qualities as carriers of human existence and development: nonrenewable resources, fixed location, and imbalanced distribution [1]. With the population and economy expanding at such a rapid pace, the amount of available land resources is gradually diminishing. In modern society, buildings and roads are the basic components of urban layout, and accurately extracting buildings and roads from remote sensing satellite images helps to realize the macro-planning of the city [2]. The research methods for remote sensing images can be divided into two parts: traditional theoretical calculation methods and artificial intelligence big data analysis methods. The traditional theoretical calculation method is to extract image texture features through theoretical calculation of each pixel of the image, so as to realize remote sensing image segmentation and target extraction. Although the traditional method has reached a certain standard in terms of segmentation accuracy, it needs to manually set the calculation parameters, which consumes human resources and material resources and lacks in calculation efficiency. On the contrary, the deep learning method in artificial intelligence can complete the end-to-end remote sensing image segmentation [3,4] and realize the high-accuracy automatic image segmentation function without manual intervention [5,6]. The dividends brought by the big data era have made a qualitative leap in computing efficiency, which results in that the computing efficiency has been greatly improved compared to traditional methods. Therefore, it is of great importance to use the deep learning method in artificial intelligence to achieve semantic segmentation of remote sensing images.

In the past few years, there have been many works on the semantic segmentation of remote sensing images. For example, Yuan et al. [7] calculated the texture features and spectrum of the image using local spectral histograms, linearly combined representative features using each local spectral histograms, classified the pixel by weight estimation, and finally realized the segmentation of images. This method could greatly reduce the feature dimension of the network through subspace projection, and realize that the input dimension of the network could be selected adaptively. However, the disadvantage is that only spectral information is used in the calculation process. Li et al. [8] proposed a watershed algorithm for edge embedding markers, which was used for the segmentation of high-resolution remote sensing images. The method results in improvements in the two key steps of segmentation (one is label extraction and the other one is pixel labeling), which could improve the accuracy of edge segmentation in high-resolution images. Furthermore, this method used an edge embedding detector to extract edge information with confidence, which was usually used in situations with weak boundaries and improved the positioning accuracy of target boundaries. Although the accuracy of the segmentation boundary had been improved, there are also problems that the detailed feature information is complex and the interference factors make it difficult to obtain information. Fan et al. [9] found that these remote sensing segmentation methods rarely use prior information. As a result, he proposed a new approach based on prior information. A single-point iterative weighted fuzzy C-means clustering algorithm was used in this method, which solved the data distribution and the effect of random initialization of cluster centers on the quality of clustering. The above feature segmentation method can divide remote sensing images effectively, but there are also problems that exist, such as weak noise resistance, slow speed of segmentation, manual parameter design, etc., which cannot be used for the task of automatically segmenting large amounts of data.

Current deep learning is still under development in the area of building and road extractions. Panboonyuen et al. [10] proposed an enhanced deep convolutional encoding and decoding network for road segmentation of remote sensing images, combining ELU activation function [11] and SegNet network [12] to form an end-to-end segmentation network, and finally through optimizing indicators and removing false road objects to further improve the overall effect. However, for this method, fewer applications of continuous feature information are used when extracting roads from remote sensing images, which led to the interference map and fracture area. Aiming at the problem of loss of detailed information during the down-sampling process, Sun et al. [13] offered a new feature fusion strategy based on a full convolutional network with ultra-high resolution image segmentation, which maximized the fusion of deep-level semantic features and shallow-level detail information. Combining with this model, the effective digital surface model was proposed, and the information of high-resolution remote sensing images was extracted, which improved the accurate segmentation of the full convolutional network. However, in remote sensing image segmentation, there were problems that the scale of the target segmentation was inconsistent and the scale of information had not been mined. In order to solve the problem, Liu et al. [14] proposed a multi-channel deep convolutional neural network to alleviate the loss of spatial and scale features of segmented targets in images. Qi et al. [15] proposed a multi-scale convolution and attention mechanism based on a segmentation model. The attention mechanism, on the other hand, could only capture the local receptive field. Li et al. [16] proposed to use a two-way attention mechanism network for semantic segmentation of remote sensing images. One is focused on the spatial semantic information in the feature map, and the other is on the associated information between channels. Combining the two-way attention information could effectively improve the accuracy of segmentation. Lan et al. [17] proposed a global context road automatic segmentation neural network for road segmentation under complex background and field of view occlusion. In the network, a residual cavity convolutional network was used to provide a wide receptive field. Although with the multi-scale information of a larger receptive field, for the network, the relevance of the middle layer could not be ignored. He et al. [18] proposed a hybrid first-order and second-order attention network to enhance the relevance of feature information in the middle of the network.

In summary, for the above remote sensing image semantic segmentation methods [7–18] in deep learning, satisfactory results have been achieved. At present, in most semantic segmentation networks, down-sampling of the convolutional neural network (CNN) is used to extract features. Feature maps are compressed many times during extraction, which results in the loss of details. Up-sampling by feature maps with missing detailed features makes it difficult to restore feature maps with high-resolution and classify resolution accurately. In the process of feature extraction by deep convolutional networks, the receptive field is limited. Although a larger receptive field can be obtained by the use of hollow convolution and feature pyramids, the receptive field is still local, understanding of long-distance scenes cannot be achieved, and pixels cannot be classified precisely. To solve these problems, a Hybrid Multi-resolution and Transformer semantic extraction Network (HMRT) is proposed in this study. In general, this work has made three contributions: (1) The multi-resolution semantic extraction branch is constructed. In this structure, branches of different resolutions conduct feature fusion, which not only ensures that high and multi-resolution are kept during the down-sampling process but also ensures that feature information is retained. (2) The Transformer sequence feature extraction network is introduced. In this network, each pixel with a global receptive field is realized by the use of encoding and decoding, and in the meantime, the location information of the pixel is overlain. The small receptive field of the convolutional neural network can be overcome and the understanding of a long-distance scene can be improved. (3) Feature Channels Maximum Element is proposed to strengthen the class location information, which can effectively improve the accuracy of segmentation.

2. Methodology

In order to solve the two problems of the loss of details caused by scale compression in the down-sampling process and the lack of long-distance understanding due to the limitation of the receptive field, this paper proposes a Hybrid Multi-resolution and Transformer semantic extraction Network (HMRT). The overall framework of the HMRT is shown in Figure 1. The overall framework of the HMRT proposed in this work is divided into two parallel branches. The first branch provides the network with different resolution feature maps. The feature maps of different resolutions are divided into 2 times down-sampling, 4 times down-sampling, and 8 times down-sampling. There are 3 different stages in the down-sampling process of each resolution feature map. Each stage will map the channel of the feature map and increase the dimension, and the number of channels are 64, 128, and 256, respectively. Finally, the three feature maps of different resolutions are cross-fused. In the process of feature map cross-fusion, the feature maps of different scales are sampled and restored to the input image size, so that the second branch can use these feature maps directly when the features are merged. The second branch mainly uses a combination of convolutional neural networks and transformers to extract semantic information from the global receptive field of the feature map. First, it extracts the local feature map information of the input image through the convolutional neural network and obtains the 16 times down-sampled feature map. Next, unlike the current semantic segmentation network, it uses the Transformer method to continue to encode and decode the 16 times down-sampled feature map. The advantage of the Transformer encoding and decoding method is to make the entire feature map perform a global receptive field, which overcomes the limitation of the receptive field caused by the small convolution kernel of the convolutional neural network. In addition, the position information of each pixel in the feature map is introduced in the encoding process, so that each pixel adds semantic information in the position dimension. In the decoding process, the category high-dimensional mapping matrix is used as the query matrix, and the key-value matrix and the numerical matrix are from the output of the Transformer encoding layer. The structure diagram of the HMRT is shown in Figure 1, and there are two different dimensional transformations in the network layer. The dimension of the 3-dimensional feature map indicates the quantity of channels, the height and width of the feature map. In the 2-dimensional feature map, *N* represents the number of categories, and *D* represents the hidden layer mapping dimension of the Transformer encoding and decoding network. The upper half of the figure is the multi-resolution semantic extraction branch, and the lower half is the Transformer semantic extraction branch.

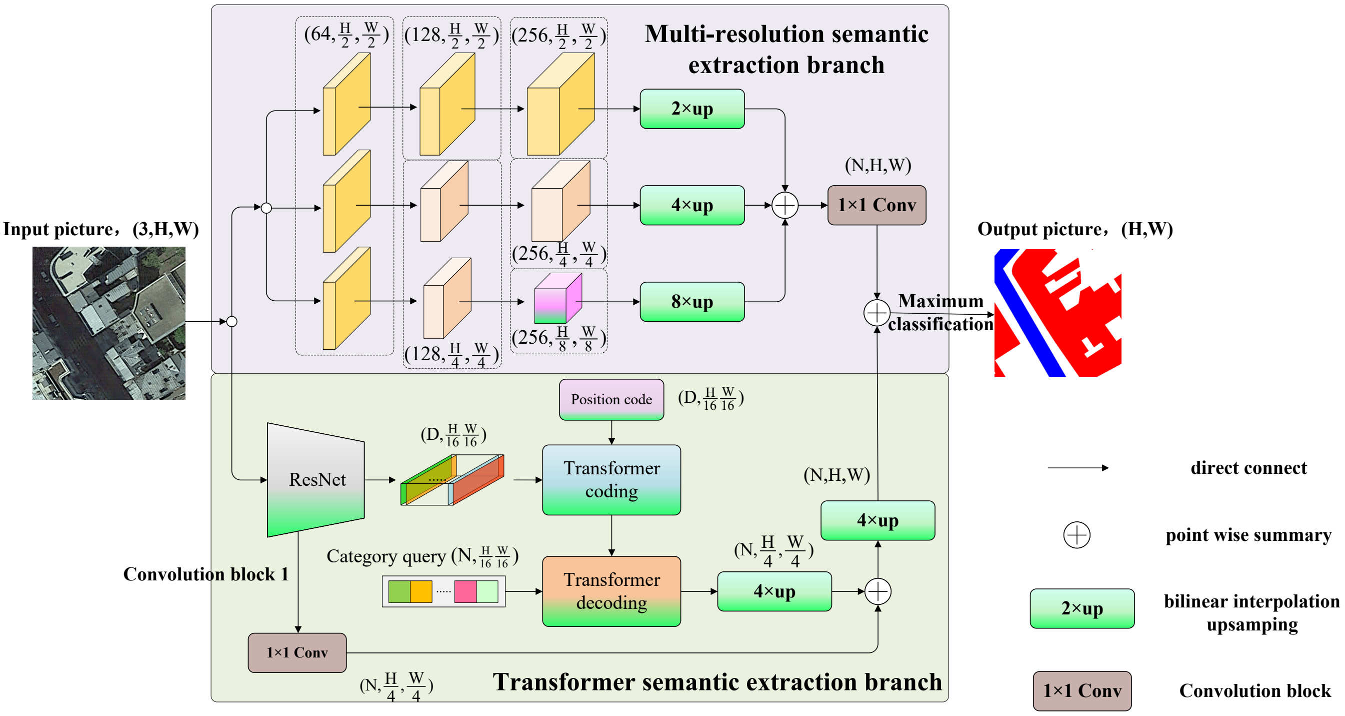

Multi-resolution semantic extraction branch
$(64,\frac{H}{2},\frac{W}{2})$
$(128,\frac{H}{2},\frac{W}{2})$
$(256,\frac{H}{2},\frac{W}{2})$
$(256,\frac{H}{4},\frac{W}{4})$
$(256,\frac{H}{8},\frac{W}{8})$
$(128,\frac{H}{4},\frac{W}{4})$
2×up
4×up
8×up
(N,H,W)
1×1 Conv
Input picture，(3,H,W)
Output picture，(H,W)
Maximum classification
Position code
$(D,\frac{H}{16}\frac{W}{16})$
ResNet
$(D,\frac{H}{16}\frac{W}{16})$
Transformer coding
(N,H,W)
4×up
Convolution block 1
Category query $(N,\frac{H}{16}\frac{W}{16})$
Transformer decoding
$(N,\frac{H}{4},\frac{W}{4})$
4×up
1×1 Conv
$(N,\frac{H}{4},\frac{W}{4})$
Transformer semantic extraction branch
direct connect
point wise summary
2×up
bilinear interpolation upsamping
1×1 Conv
Convolution block

Figure 1. Hybrid multi-resolution and Transformer semantic extraction network framework.

2.1. Multi-Resolution Semantic Extraction Branch

The significance of the multi-resolution semantic extraction branch is that in the down-sampling feature extraction process of the convolutional neural network, the maximum pooling layer or the 3 × 3 convolution kernel with a sliding convolution stride of 2 is usually used for feature map length and width compression. Although rich semantic information is obtained in this way, it is inevitable that a lot of detailed information is discarded. In order to intuitively understand the differences in the feature map compression process, we show three feature compression ways, they are: (1) convolution with a 3 × 3 convolution kernel, a stride of 1, and a padding of 1. (2) Convolution with a 3 × 3 convolution kernel, a stride of 2, and a padding of 1. (3) The maximum pooling with a window of 2 × 2 size and a stride of 2. The visualization effects of the three feature compression ways are shown in Figure 2, Figure 2a shows the convolution with a stride of 1, Figure 2b shows the convolution with a stride of 2, Figure 2c shows the maximum pooling with a stride of 2. The size of the input picture demonstrated in Figure 2 is 512 × 512.

As is shown in Figure 2, the convolution with a stride of 1 is the most sufficient for feature extraction, almost retaining all the feature details in the original image, and the output feature map size is consistent with the input image specification (512 × 512). In the feature extraction process of the convolution operation with a stride of 2, the size of the output feature is 256 × 256, which is compressed to half of the original input image. When comparing Figure 2b with Figure 2a, it is not difficult to see that Figure 2b has significantly less feature information than Figure 2a. Then, compare the output result of the maximum pooling feature compression in Figure 2c with Figure 2a,b, the image in Figure 2c appears jagged and the loss of feature information is more serious than that of Figure 2b. Finally, we conclude that the semantic information expression capabilities of feature compression are as follows: the convolution with a stride of 1 is greater than the convolution with a stride of 2, and the convolution with a stride of 2 is greater than the maximum pooling operation with a stride of 2.

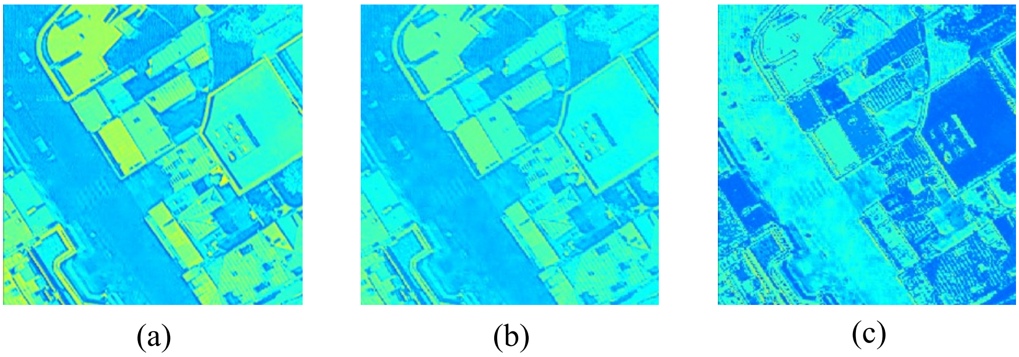


Figure 2. Different feature map compression visualization effects: (**a**) convolution with a stride of 1 (**b**) convolution with a stride of 1 (**c**) maximum pooling with a stride of 2.

At present, most of the current convolutional neural networks need to use the maximum pooling operation with a stride of 2 and use the convolution operations with a stride of 2 repeatedly in the feature extraction process, which will lead to the loss of detailed information in the feature map [19,20]. In order to overcome this difficulty, this section proposes a multi-resolution semantic extraction branch to provide rich multi-resolution feature maps for the network. The multi-resolution semantic extraction branch proposed in this section is divided into three branches. The three branches use the same input, but the input image is down-sampled at different multiples to obtain feature maps of different resolutions. The three feature maps with different resolutions are 2 times down-sampling, 4 times down-sampling and 8 times down-sampling.

The branch structure of multi-resolution semantic extraction is shown in Figure 3, the input of the whole branch is a picture of $3 \times H \times W$ size. The first branch in Figure 3 is 2 times down-sampling, consisting of a residual module with a stride of 2 and two residual modules with a stride of 1. The second branch in Figure 3 is 4 times down-sampling, consisting of two residual modules with a stride of 2 and a residual module with a stride of 1. The third branch in Figure 3 is 8 times down-sampling, consisting of three residual modules with a stride of 2. Each branch is composed of three residual modules, and the three residual modules will gradually increase the number of channel mapping during the down-sampling process. The number of channel mapping is 64, 128 and 256, respectively. After the three branches pass through the residual modules with different strides, the down-sampling feature maps of 2 times, 4 times and 8 times are obtained, respectively. Next, the feature maps need to be gathered and fused. However, the resolution of the feature maps is inconsistent, and it needs to be standardized to the same level. Therefore, the feature maps need to be up-sampled, and the multiple of the up-sampling is the inverse transform of the down-sampling multiple, which are 2 times up-sampling, 4 times up-sampling, and 8 times up-sampling, respectively. After sampling on the three branches, feature maps are restored to the size of the input image and then correspondingly added and fused in the channel dimension to obtain a feature map of $256 \times H \times W$ size. Finally, 1×1 convolution is used to map the number of channels of semantic information feature maps containing multiple resolutions to the number of categories N that can be learned by the model. As a result, the multi-resolution semantic extraction feature map ($N \times H \times W$) of category information is obtained.

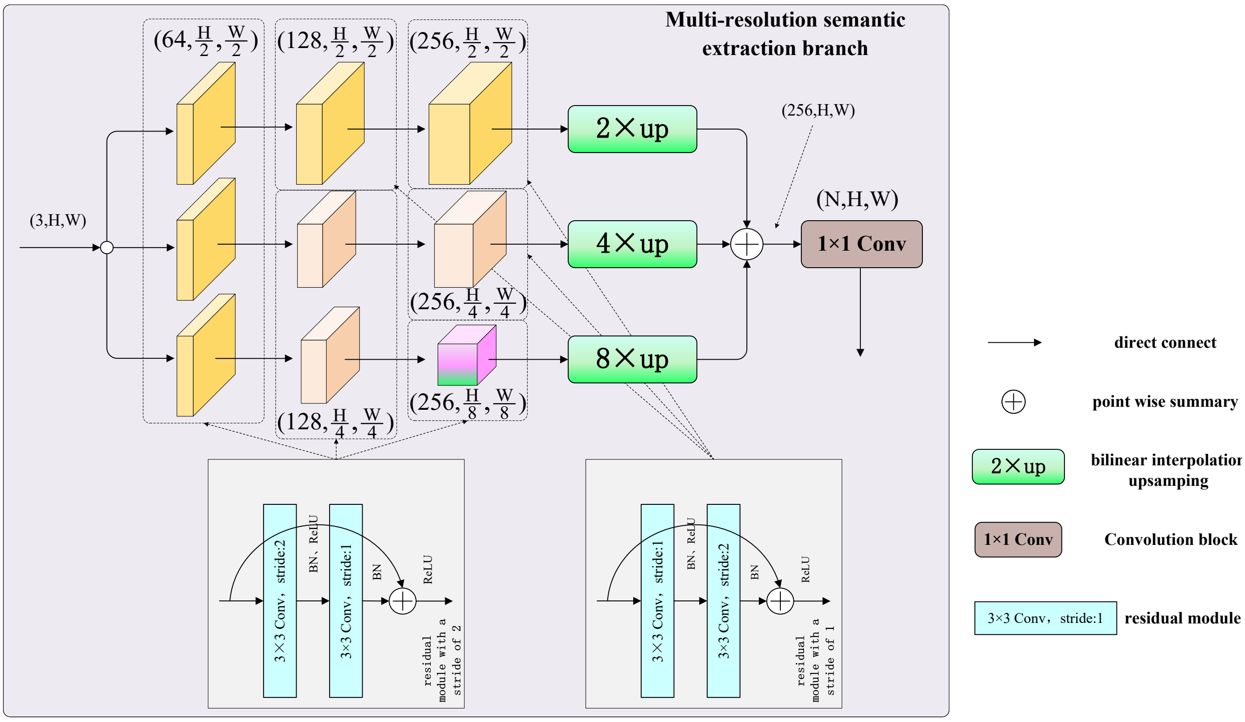


Figure 3. Structure diagram of multi-resolution semantic extraction branch.

Many excellent CNNs have appeared in recent years, including ResNet [21], VGG [22] and GoogLeNet [23]. After considering the amount of parameters and network accuracy, this work adopts the ResNet-18 basic module as the residual block in the three branches. The structure of the two residual modules in Figure 3 is shown in Figure 4. This work uses residual modules with two kinds of strides, the forward propagation of the residual module with a stride of 1 is shown in Equation (1).

$$X_{out} = \sigma(\beta(Conv_{3\times3}(\beta(\sigma(Conv_{3\times3}(X))))) + X), \tag{1}$$

where $Conv_{3\times3}$ is a 3 × 3 convolution with a stride of 1, β is BN, σ is a ReLU activation function.

The forward propagation of the residual module with a stride of 2 is shown in Equation (2).

$$X_{out} = \sigma(\beta(Conv'_{3\times3}(\beta(\sigma(Conv_{3\times3}(X))))) + X), \tag{2}$$

where $Conv_{3\times3}$ is a 3×3 convolution with a stride of 2, β is BN, σ is a ReLU activation function, $Conv'_{3\times3}$ is a 3×3 convolution with a stride of 1.

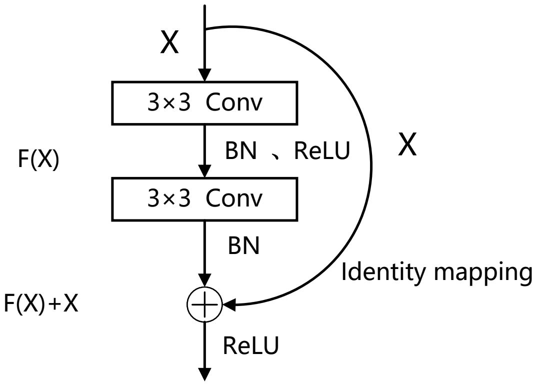


Figure 4. Structure diagram of ResNet-18 residual module.

The specific structure of the multi-resolution semantic extraction branch is shown in Table 1. The table shows the parameter settings of the six stages (input, first stage, second stage, third stage, feature convergence and output) of the branch.

Table 1. The specific structure of multi-resolution semantic extraction branch.

Stages	Number of Channels	2 Times Down-Sampling Branch	4 Times Down-Sampling Branch	8 Times Down-Sampling Branch
Input	3		3 × H × W	
The first stage	64	Convolution stride: 2	Convolution stride: 2	Convolution stride: 2
The second stage	128	Convolution stride: 1	Convolution stride: 2	Convolution stride: 2
The third stage	256	Convolution stride: 1	Convolution stride: 1	Convolution stride: 2
Feature convergence	256	2 times up-sampling	4 times up-sampling	8 times up-sampling
	N		1 × 1 convolution changes the number of channels	
Output	N		Accumulate to obtain the feature maps of N × H × W size	

2.2. Transformer Semantic Extraction Branch

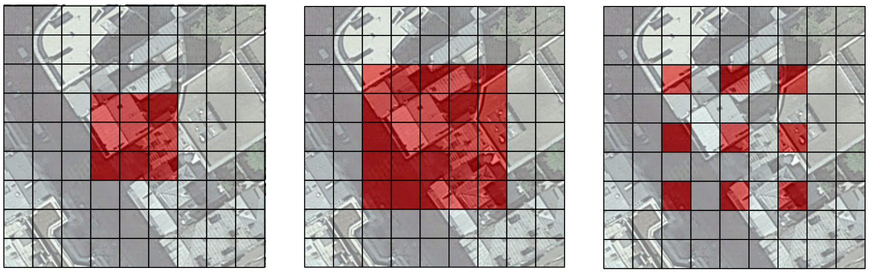
The significance of the Transformer semantic extraction branch is that the small receptive field of ordinary convolutional neural networks causes a lack of long-distance scene understanding. Although there have been many methods to improve the problem of small perception fields, such as enlarging the convolution kernel and using atrous convolution, they all have certain drawbacks. On the one hand, after enlarging the convolution kernel, the amount of model parameters and model calculations will increase, which will increase a lot of computational overhead. It is not a good choice in the case of lack of time and limited computing resources. On the other hand, although the use of atrous convolution can expand the receptive field of the original convolution kernel without adding additional calculations, the atrous convolution is filled with 0 when the convolution kernel is expanded, resulting in loss of internal details. The atrous convolution is more friendly to the extraction of large target objects and can capture the long-distance dependence of large target objects, but the advantages of small target objects are not obvious enough. Since the 0 padding used by the convolution kernel affects the continuity of the convolution kernel in the feature extraction process, small target objects are split or ignored, which affects the effectiveness of small target object extraction. The change in the size of the receptive field plays a big role in the feature extraction process. A convolution kernel with a large receptive field can extract the long-distance dependence of a large target, while a convolution kernel with a small receptive field can extract the complete features of a small target object. Taking the dimension of the convolution kernel width as an example, the derivation process of the receptive field is shown in Equation (3).

$$K' = K + (K-1) \times (d-1), \tag{3}$$

where K is the size and width of the convolution kernel, d is dilation rate, K' is the size of the receptive field.

In order to reflect the difference of the receptive fields conveniently and intuitively, this work designs different sizes of convolution kernels and different sizes of dilation rates for visualization. The comparison of the receptive fields with different dilation rates and convolution kernels is shown in Figure 5.

It can be seen that the current semantic segmentation networks have limitations in the receptive field, so this work combines the Transformer method with global receptive fields [24–28] to deeply mine the semantic information of the feature maps. On this basis, a hybrid convolutional neural network (ResNet-18) and a Transformer semantic extraction branch based on Transformer encoding and decoding are constructed.



(a) (b) (c)

Figure 5. Comparison of different void rates and convolution kernel receptive fields. (**a**) Receptive field of 3 × 3 convolution kernel; (**b**) receptive field of 5 × 5 convolution kernel; (**c**) receptive field of 5 × 5 convolution kernel with a dilation rate of 2.

2.2.1. The Overall Framework of Transformer Semantic Extraction Branch

The overall framework of the Transformer semantic extraction branch is composed of a hybrid convolutional neural network (backbone network) and Transformer encoding and decoding modules. The structure of this branch is shown in Figure 6, *D* in the figure is the mapping dimension, and *N* is the number of categories. Firstly, the backbone network adopts ResNet-18 with a sliding window to extract features. Secondly, the backbone network performs 16 times down-sampling, flattening the feature map in the width and height dimensions to obtain a feature map with a dimension of (D, H/16, W/16). Next, we add the obtained feature map to the position coding matrix of the same size and input the result into the Transformer coding module for global coding. The number of times the encoding module repeats the encoding is set to 6. Corresponding Transformer decoding is performed after encoding, and a total of 6 decoding layers are set. The query matrix of the first decoding layer is provided by the category matrix and the query matrix after the second layer is the output of the previous decoding layer. In addition, the key-value matrix and the numerical matrix are also the output of the decoding matrix of the previous layer. After Transformer encoding and decoding, the feature map with a dimension of (H, H/16, W/16) is outputted. Then, the last dimension of the feature map is flattened into two dimensions to obtain a feature map (H, H/16, W/16) with the number of channels of category N. Next, the new feature map is up-sampled 4 times to obtain the feature map with a dimension of (H, H/4, W/4). The reason for using 4 times up-sampling is to make the size of the feature map consistent with the convolution block 1 of the backbone network. Since the feature map of the convolution block 1 is used to enrich position information, the fusion of the feature map can help position restoration. Finally, the feature map needs to be restored to the input image size to achieve pixel-level classification. In the restoration process, the feature map is up-sampled 4 times to obtain the feature map.

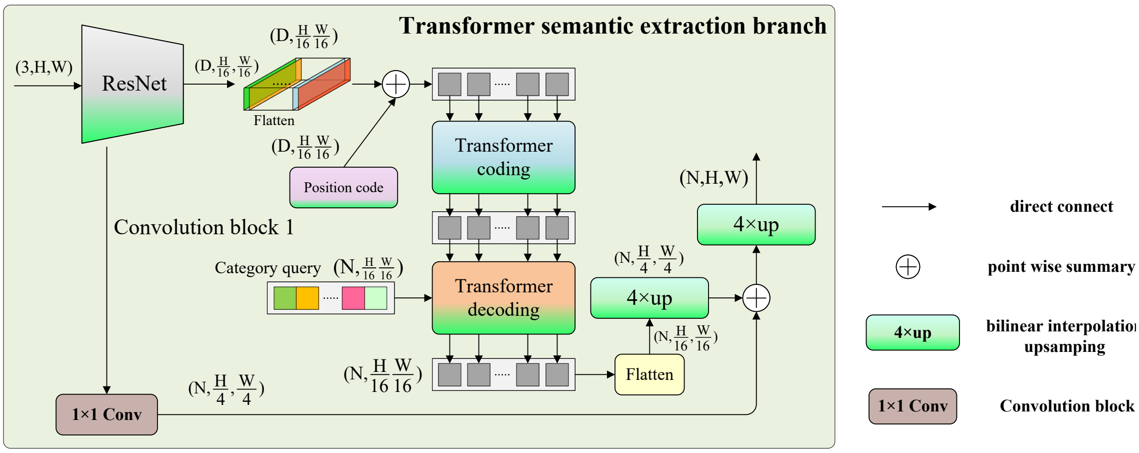

Transformer semantic extraction branch
(3,H,W)
ResNet
$(D,\frac{H}{16},\frac{W}{16})$
$(D,\frac{H}{16},\frac{W}{16})$
Flatten
$(D,\frac{H}{16},\frac{W}{16})$
Position code
Transformer coding
(N,H,W)
4×up
Convolution block 1
Category query
$(N,\frac{H}{16},\frac{W}{16})$
$(N,\frac{H}{4},\frac{W}{4})$
Transformer decoding
4×up
$(N,\frac{H}{16},\frac{W}{16})$
$(N,\frac{H}{16},\frac{W}{16})$
Flatten
$(N,\frac{H}{4},\frac{W}{4})$
1×1 Conv
direct connect
point wise summary
4×up
bilinear interpolatio upsamping
1×1 Conv
Convolution block

Figure 6. Structure diagram of Transformer semantic extraction branch.

2.2.2. The Feature Extraction of Backbone Network

Transformer semantic extraction branch proposed in this section uses a convolutional neural network as the backbone network for feature extraction. ResNet-18 is used to extract the deep semantic information of the image, but the structure of ResNet-18 used in this paper is slightly different from the structure in the original paper.

As shown in Table 2, after the picture of 512 × 512 size passes through the convolution block 1, convolution block 2, convolution block 3, and convolution block 4 that are consistent with the original paper, a feature map of 32 × 32 size can be obtained. Next, the network maintains this resolution and uses convolution to deepen the extracted features. Convolution deepening is completed by convolution block 5 and convolution block 5 is a 3 × 3 convolution with a stride of 1 and 256 channels. The final output feature map is 1/16 of the size of the input image, which is twice as large as the output size of the original backbone network. This maintains a large resolution feature map, which is conducive to the extraction of richer feature information from Transformer global features. In addition, the number of channels of the output feature map is also reduced by half. The reason is that the Transformer connected to the backbone network still has a high-dimensional mapping channel when it continues to encode and decode. The reduction in the number of channels in the backbone network can reduce a certain amount of parameters.

Table 2. The specific structure of the backbone network ResNet-18.

Modules	The Size of the Feature Map	ResNet-18
Input	512 × 512	-
convolution block 1	256 × 256	7 × 7, Number of channels: 64, Stride: 2, padding 3 × 3, Maximum pooling layer, Stride: 2
convolution block 2	128 × 128	3 × 3, Number of channels: 64, Stride: 2 3 × 3, Number of channels: 64, Stride: 1
convolution block 3	64 × 64	3 × 3, Number of channels: 128, Stride: 2 3 × 3, Number of channels: 128, Stride: 1
convolution block 4	32 × 32	3 × 3, Number of channels: 256, Stride: 2 3 × 3, Number of channels: 128, Stride: 1
convolution block 5	32 × 32	3 × 3, Number of channels: 256, Stride: 1 3 × 3, Number of channels: 128, Stride: 1

2.2.3. Transformer Encoding and Decoding

Transformer encoding and decoding was first proposed by Vaswani et al. [29] for natural language processing. This method extracts global information from the input feature map. Inspired by this innovation, this paper transplants and fine-tunes the Transformer to the semantic segmentation task to make up for the limitations of the convolutional neural network's receptive field when performing semantic segmentation. The overall structure of the improved Transformer in this paper is composed of encoding and decoding, and the encoding and decoding modules are spliced from the self-attention mechanism into a multi-head attention mechanism. The structure of the Transformer is shown in Figure 7.

Firstly, the input feature map is the feature map (D, H/16, W/16) extracted by the backbone network. Secondly, the flattening function maps the last two dimensions into a one-dimensional vector to obtain a new feature map (D, H/16, W/16). Then the position coding matrix $p \in R^{(\mathrm{D,H/16,W/16})}$ of each pixel in the feature map and the feature map are superimposed as the input of the coding layer. The coding layer first undergoes the feature normalization layer to normalize the channel dimensions, and then passes through different matrix mappings to obtain the query matrix $q \in R^{(\mathrm{H/16,W/16,D})}$, the key-value matrix $k \in R^{(\mathrm{H/16,W/16,D})}$, and the numerical matrix $v \in R^{(\mathrm{H/16,W/16,D})}$. The calculation process is shown in Equations (4)–(7). X_{input} is the input of the coding layer. Δ represents feature normalization layer. K_q is an encoding layer query matrix mapping function. K_k is the coding layer key-value matrix mapping function. K_v is the coding layer numerical matrix mapping function.

$$X = \Delta(X_{input}), \tag{4}$$

$$q = K_q(X), \tag{5}$$

$$k = K_k(X), \tag{6}$$

$$v = K_v(X), \tag{7}$$

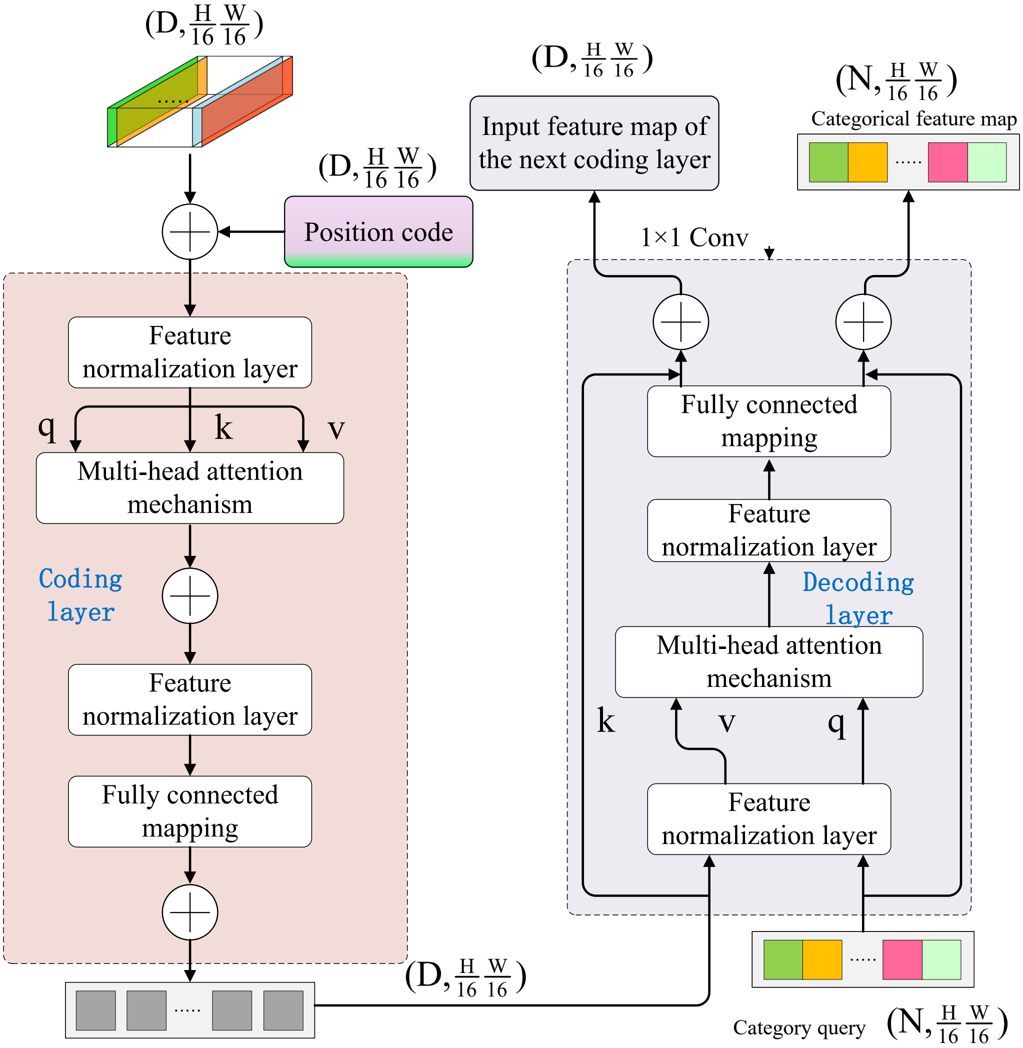


Figure 7. The structure diagram of Transformer encoding and decoding.

After obtaining the query matrix, the key-value matrix and the numerical matrix, the three matrices are input into the multi-head attention mechanism module for attention calculation. The multi-head attention mechanism module is obtained by splicing multiple self-attention mechanism modules (this paper sets the number of heads of the multi-head attention mechanism to 4). The advantage of having multi-headed attention is that feature information can be obtained from different branches to enrich semantic information, and different branches can independently extract features and then merge them, which can increase the diversity of feature extraction. After the multi-head attention mechanism module, the output feature map and the feature map before the feature normalization layer aggregate the original feature map information through jump connection. As is shown in Equation (8).

$$X_{atten} = \Gamma(q, k, v) + X_{input}, \tag{8}$$

where Γ represents multi-head attention mechanism module, X_{atten} is the output of the multi-head attention mechanism, q, k, v represent the query matrix, key-value matrix, and numeric matrix of the coding layer, respectively. X_{input} is the input of the coding layer.

Then through the feature normalization layer and the fully connected layer, the feature dimension is mapped to the high dimension. The fully connected mapping is a 4 times mapping. Finally, the original feature map before entering the full connection is also aggregated by jump connection to obtain the output feature map of the coding layer (HW,D). The calculation process is shown in Equation (9).

$$X_{encoder} = \Pi(\Delta(X_{atten})), \tag{9}$$

where $X_{encoder}$ is the output of the coding layer, Π represents fully connected mapping, Δ represents the feature normalization layer, and X_{atten} is the output of the multi-head attention mechanism.

The input of the decoding layer is composed of two parts: the output feature map of the coding layer and category query feature map created based on the number of categories. The feature layer is normalized before entering the multi-head attention mechanism of the decoding layer, and then the output feature map of the coding layer is decomposed into a key-value matrix $k' \in R^{(\mathrm{H}/16,\mathrm{W}/16,\mathrm{N})}$ and a numerical matrix $v' \in R^{(\mathrm{H}/16,\mathrm{W}/16,\mathrm{N})}$ through different matrix mappings. Among them, N is the number of categories. The calculation process is shown in Equations (10)–(12).

$$X'_{encoder} = \Delta(X_{encoder}), \tag{10}$$

$$k' = K'_k(X'_{encoder}), \tag{11}$$

$$v' = K'_v(X'_{encoder}), \tag{12}$$

where $X'_{encoder}$ is the output of the feature normalization layer, Δ represents feature normalization layer, $X_{encoder}$ is the output of the coding layer, K'_k represents key-value matrix mapping function in the decoding layer, K'_v represents the numerical matrix mapping function in the decoding layer.

The decoding layer query matrix $q' \in R^{(\mathrm{H}/16,\mathrm{W}/16,\mathrm{N})}$ is obtained according to the category initialization, and the key-value matrix $k' \in R^{(\mathrm{H}/16,\mathrm{W}/16,\mathrm{N})}$ and the numerical matrix $v' \in R^{(\mathrm{H}/16,\mathrm{W}/16,\mathrm{N})}$ are obtained through matrix mapping. Next, we input them into the multi-head attention mechanism for decoding at the same time, and the decoded result goes through the feature normalization layer and fully connected mapping. The fully connected mapping is a 4 times mapping. Then, the first feature of the decoding layer and the feature map before the feature normalization layer are correspondingly fused. Finally, the output category feature map of the decoding layer and the input feature map of the next coding layer are obtained. The calculation process is shown in Equation (13).

$$X_{decoder} = \Pi(\Delta(\Gamma(q', k', v'))) + q', \tag{13}$$

where $X_{decoder}$ is the output of the decoding layer, Π represents fully connected mapping, Δ represents the feature normalization layer, Γ represents the multi-head attention mechanism module, q', k', v' are the query matrix, key-value matrix and numerical matrix of the decoding layer

The structure of the multi-head attention mechanism module is shown in Figure 8. As shown in Figure 8, query matrix q, key-value matrix k, and numerical matrix v are inputed. The calculation process of a single attention mechanism is as follows: Firstly, the key-value matrix and the transpose of the query matrix are multiplied. Secondly, Softmax is performed at the last of the results obtained. Finally, the result of Softmax and numerical matrix are multiplied to obtain the attention result. In addition, the multi-head attention mechanism proposed in this paper splices the single-head attention mechanisms to obtain the attention mechanism information of different branches.

The decomposition process of a single self-attention mechanism module is shown in Figure 8. In the case of inputting three targets, the three targets are calculated by their corresponding query matrix mapping matrix W_{query}, key-value mapping matrix $W_{key-value}$, and numerical mapping matrix $W_{numericlvalue}$ to obtain their respective query matrix, key-value matrix, and numerical matrix $(q_1, q_2, q_3, k_1, k_2, k_3, v_1, v_2, v_3)$. Next, Softmax calculates the weights of the target itself and all targets. The calculation process of measure weights is shown in Equation (14), and the decomposition structure of the calculation process of the self-attention mechanism is shown in Figure 9.

$$Weights_i = \frac{q_i * k_i^T}{\sum_i (q_i * k_i^T)}, \tag{14}$$

where $Weights_i$ is the measure weights of the i-th goal, q_i represents the query matrix of the i-th target, k_i represents the key-value matrix of the i-th target.

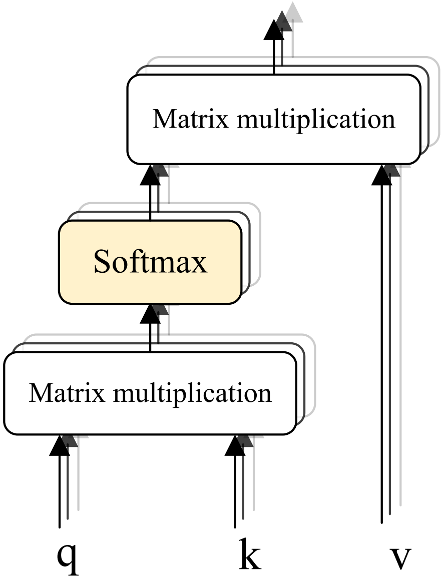


Figure 8. The structure diagram of multi-head attention mechanism.

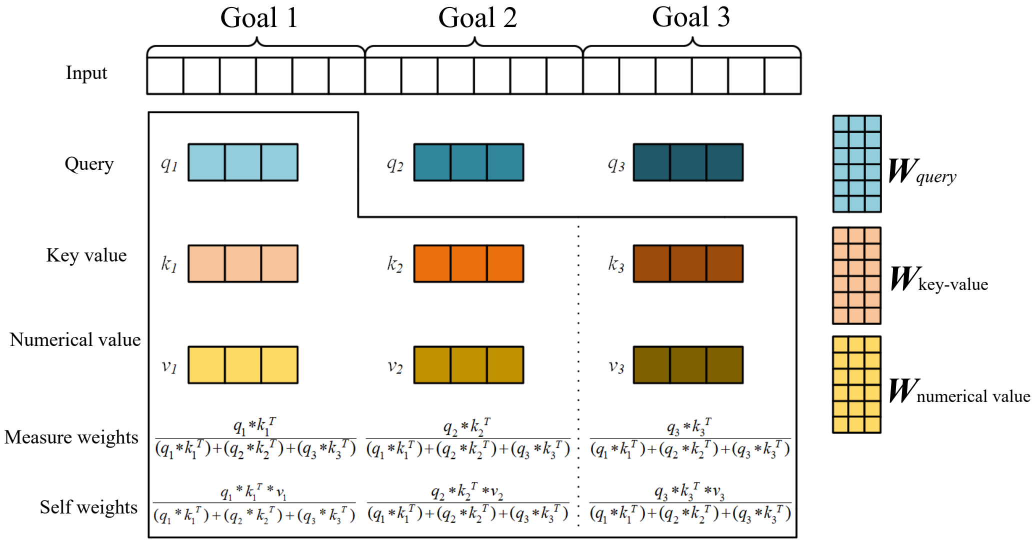

Goal 1
Goal 2
Goal 3
Input
Query
Key value
Numerical value
Measure weights
Self weights
q_1
q_2
q_3
k_1
k_2
k_3
v_1
v_2
v_3
$\frac{q_1 * k_1^T}{(q_1 * k_1^T)+(q_2 * k_2^T)+(q_3 * k_3^T)}$
$\frac{q_2 * k_2^T}{(q_1 * k_1^T)+(q_2 * k_2^T)+(q_3 * k_3^T)}$
$\frac{q_3 * k_3^T}{(q_1 * k_1^T)+(q_2 * k_2^T)+(q_3 * k_3^T)}$
$\frac{q_1 * k_1^T * v_1}{(q_1 * k_1^T)+(q_2 * k_2^T)+(q_3 * k_3^T)}$
$\frac{q_2 * k_2^T * v_2}{(q_1 * k_1^T)+(q_2 * k_2^T)+(q_3 * k_3^T)}$
$\frac{q_3 * k_3^T * v_3}{(q_1 * k_1^T)+(q_2 * k_2^T)+(q_3 * k_3^T)}$
W_{query}
$W_{\text{key-value}}$
$W_{\text{numerical value}}$

Figure 9. Decomposition structure diagram of the calculation process of the self-attention mechanism.

Finally, the self weights are obtained by multiplying the weights of each target and the corresponding numerical matrix. The calculation process is shown in Equation (15).

$$Attention_i = Weights_i * v_i, \tag{15}$$

where $Attention_i$ is the self weights (Attention information) of the i-th goal, $Weights_i$ is the measure weights of the i-th goal, v_i represents the numerical matrix of the i-th target.

3. Experiment and Result Analysis

In this chapter, experiments were carried out on the Aerial Image Segmentation Dataset (AISD) [30] and ISPRS 2D Semantic Labeling Contest (ISPRS) [31]; the HMRT model was compared with many of the best models currently available FCN-8S [32], U-Net [33], PSPNet [34] and DeeplabV3+ [35]; the overall accuracy rate (OA), recall rate (Recall), F1-Score and mean intersection over union (MIoU) are used as the quantitative analysis indicators of the experiment. The results show that HMRT is better than the comparison model in various assessment indicators.

3.1. Datasets

3.1.1. AISD Dataset

The original images of the AISD dataset were collected from OpenStreetMap online remote sensing image data, and the semantic segmentation dataset of high resolution remote sensing images were constructed by manual annotation. AISD included image data from six regions: Berlin, Chicago, Paris, Potsdam, and Zurich. This paper chose Potsdam regional data to conduct the experiment, and we named the data set as Potsdam-A. There are 24 original images and labels with an average size of 3000 × 3000 in Potsdam-A. The example of the original image and label is shown in Figure 10.

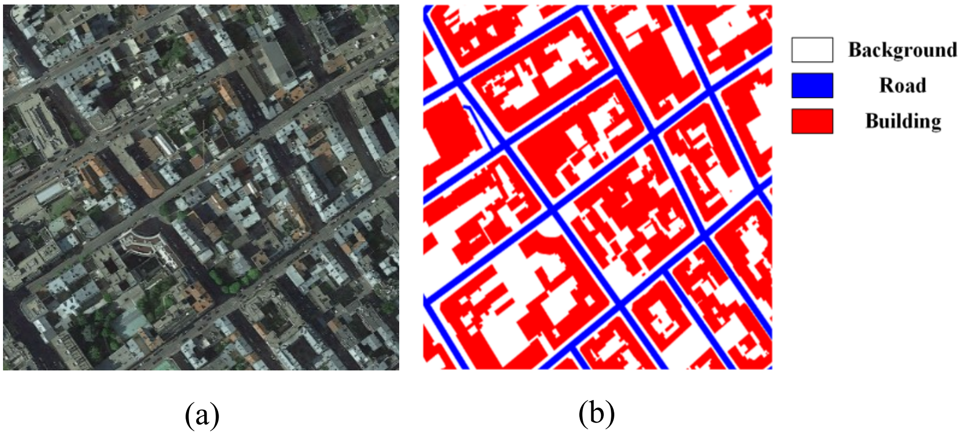


Figure 10. Original image and label example from Potsdam-A. In (**b**), the red is building; the blue is road background; the white is background. (**a**) image; (**b**) label.

Because the original image was too large to be directly input to the model training, we used Python to crop the picture of 3000 × 3000 into the picture of 512 × 512 and obtained a total of 1728 pictures finally. In the case of a small amount of data, the generalization effect was poor and the feature learning ability of the model was weak. Therefore, data enhancement was needed to ensure that the model has reliable learning capability. The original data set was randomly flipped horizontally, vertically and rotated by 90 degrees to expand to 4307 pictures.

3.1.2. ISPRS Dataset

The ISPRS 2D Semantic Labeling Contest dataset is a high-resolution aerial image dataset with complete Semantic Labeling published by the International Society for Photogram-metry and Remote Sensing (ISPRS). Similarly, we selected the Potsdam region in ISPRS to verify the generalization performance of the model, and name the data set Potsdam-B. Potsdam-B was made up of 38 accurately labeled images and five foregrounds: impervious surfaces, buildings, low vegetation, tree and car. The example of the original image and label is shown in Figure 11.

We adopted the same cropping strategy of the Potsdam-A dataset on the Potsdam-B dataset to obtain 5184 pictures of 512 × 512 size.

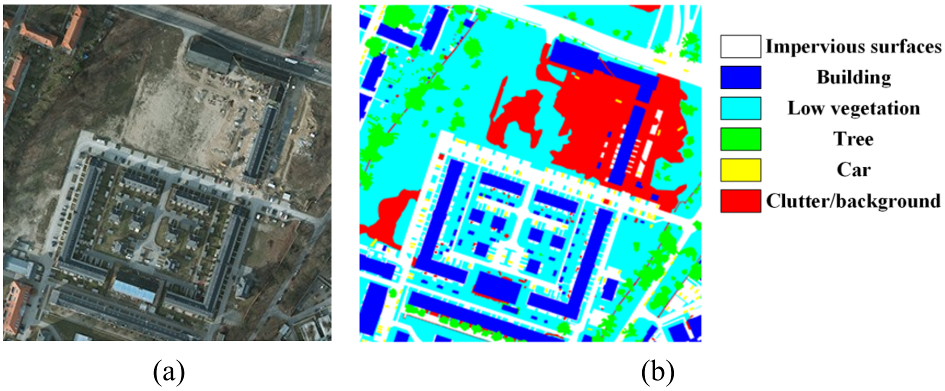


Figure 11. Original image and label example from Potsdam-B; (**a**) image; (**b**) label.

3.2. Implementation Details

In this experiment, we selected five evaluation indicators, including overall accuracy rate (*OA*), recall rate (*Recall*), F_1-Score and intersection over union (*IoU*). They are as follows:

$$OA = \frac{TP + TN}{P + N}, \tag{16}$$

$$Recall = \frac{TP}{TP + FN}, \tag{17}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall'}, \tag{18}$$

$$Precision = \frac{TP}{TP + FP'}, \tag{19}$$

$$IoU = \frac{TP}{TP + FP + FN}. \tag{20}$$

The cross-entropy loss function (CEloss) was applied to calculate the difference value between the true value and the predicted value. The model performed backpropagation and learned the best parameters under the guidance of the difference value. The derivation process of CE_{loss} is shown in Equation (21):

$$CE_{loss}(p, q) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} p(x_{ij}) log(q(x_{ij})), \tag{21}$$

where m is the quantity of samples, n represents the quantity of categories, $p(x_{ij})$ is a variable (if the category j is the same as the sample i, it is 1, otherwise it is 0), $q(x_{ij})$ is the probability sample, i is predicted to be class j.

All experiments were carried out on Ubuntu16.04 LTS with a Intel(R)Core(TM)i7-8750F CPU @2.20 GHz, 16 G of memory (RAM), and a NVIDIA GeForce RTX1060(8 GB). Python 3.8 was used, and the model was built using Pytorch1.0.1. All models were trained for 300 epochs with a batch size of 4, and the initial learning rate was 0.001.

This paper improves the post-processing of model prediction. The model prediction method adds multi-scale and sliding window splicing, which can significantly improve prediction results. The execution method of the multi-scale strategy is to enlarge the picture by 1.0, 1.25, 1.5, 1.75, 2.0 times on the basis of the original predicted picture and then predict.

After the prediction result is obtained, the size of the picture is reduced to the original picture size and added to obtain the final prediction result. The sliding window splicing strategy is to set the stride sliding window to predict the picture according to the rule from left to right and top to bottom in the upper left corner of the predicted picture. The schematic diagram of the sliding window splicing prediction strategy is shown in Figure 12. Figure 12a represents the size of the prediction window, Figure 12b represents the stride for sliding right, Figure 12c represents the stride for sliding down. In the panning process, in order to ensure that the entire picture can be predicted by the model and obtain the output, we set the panning stride to be less than or equal to the prediction window size. If the stride is smaller than the prediction window, repeated prediction parts will appear. Therefore, we use the overall summation method to realize the prediction for the repeated prediction parts.

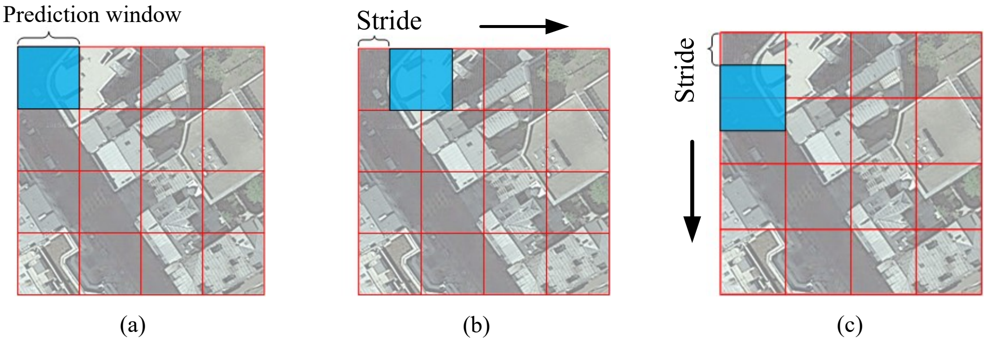


Figure 12. Schematic diagram of the sliding window splicing prediction strategy; (**a**) initial window postion; (**b**) stride for sliding right. (**c**) stride for sliding down.

3.3. Analysis of Results

3.3.1. Evaluation Metrics and Prediction Effect

(1) Main experiment

In order to test the proposed HMRT module, comprehensive experiments were carried out on the Potsdam-A dataset. The evaluation metrics are shown in Table 3, and a comparison of the prediction results is shown in Figure 13. Moreover, ablation experiments were carried out to verify the effectiveness of the multi-resolution semantic extraction branch. The network without the multi-resolution semantic extraction branch module was tested and named HMRT-1.

In Table 3, recall, F1, OA and MIoU of the HMRT obtained 85.32%, 84.88%, 85.99% and 74.19%, respectively. All four indicators were better than the comparison networks [32–35]. Among them, OA reached 85.99%, which was 0.92 higher than DeeplabV3+, and MioU reached 74.19, which was 1.37 higher than DeeplabV3+.

Table 3. The evaluation metrics on Potsdam-A test set.

Methods	Backbone	Recall (%) ↑	F1 (%) ↑	OA (%) ↑	MIoU (%) ↑
FCN-8S	VGG16	79.99	80.93	83.09	68.55
U-Net	-	82.63	82.94	84.49	71.28
PSPNet	ResNet-50	83.02	83.57	84.54	72.09
DeeplabV3+	ResNet-50	83.90	84.05	85.07	72.82
HMRT-1	-	85.17	85.14	85.80	73.75
HMRT	-	85.32	84.88	85.99	74.19

The IOU of each model on the Potsdam-A test set are illustrated in Table 4. The IOU indexes of the HMRT were 65.21%, 73.15% and 84.21%, respectively, exceeding the four comparison networks [32–35]. The IOU results showed that the HMRT has absolute advantages in segmentation accuracy.

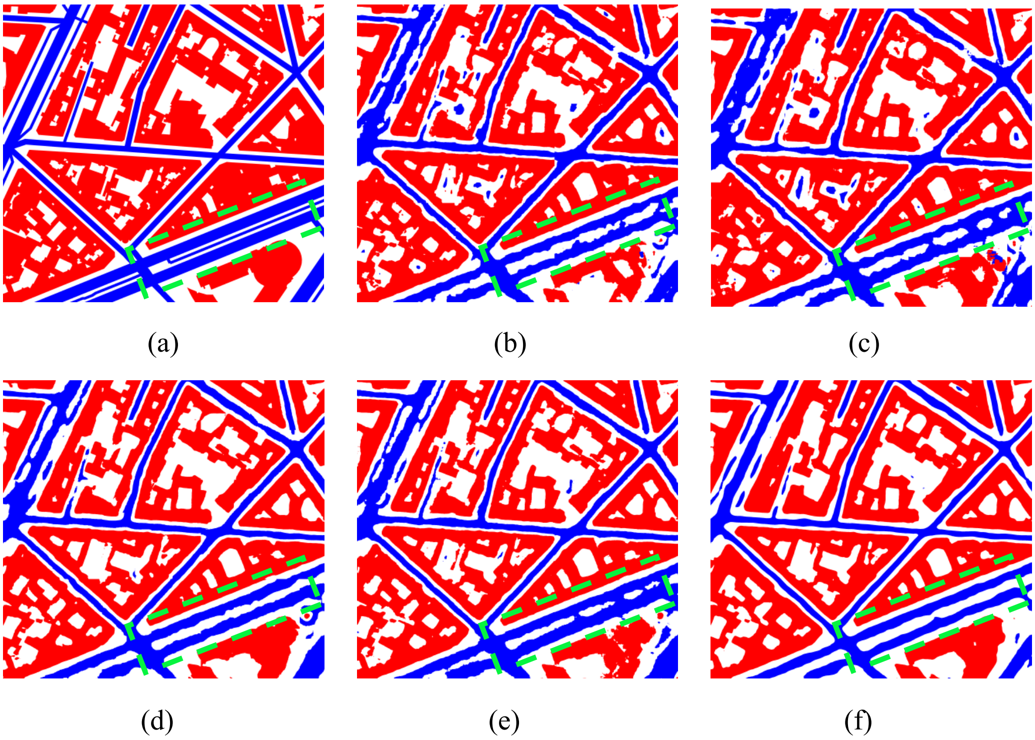


Figure 13. Comparison of the prediction results on Potsdam-A; (**a**) the superposition of the image and the label; (**b**) FCN-8S; (**c**) U-Net; (**d**) PSPNet; (**e**) DeeplabV3+; (**f**) HMRT.

Table 4. The IoU results on Potsdam-A test set.

Methods	Backbone	Background (%) ↑	Road (%) ↑	Building (%) ↑	MIoU (%) ↑
FCN-8S	VGG16	59.18	64.10	82.36	68.55
U-Net	-	62.29	68.64	82.91	71.28
PSPNet	ResNet-50	63.34	71.73	81.21	72.09
DeeplabV3+	ResNet-50	64.15	71.65	82.65	72.82
HMRT-1	-	65.00	72.54	83.71	73.75
HMRT	-	65.21	73.15	84.21	74.19

(2) Generalization experimental

To verify the generalization performance of the models proposed in this paper, the Potsdam-B data set was used for further experiment. The evaluation metrics are shown in Table 5. In Table 5, the recall, F1, OA and MIoU of HMRT reached 91.29%, 90.41%, 91.32% and 84.00%, respectively. All indicators were at their greatest levels, which could prove that the model proposed in this paper is not only effective, but has good generalization performance.

Table 5. The evaluation metrics on Potsdam-B test set.

Methods	Backbone	Recall (%) ↑	F1 (%) ↑	OA (%) ↑	MIoU(%) ↑
FCN-8S	VGG16	86.31	85.48	86.86	78.43
U-Net	-	87.74	87.41	88.51	80.87
PSPNet	ResNet-50	88.74	88.34	88.59	81.02
DeeplabV3+	ResNet-50	88.95	87.73	88.48	81.55
HMRT	-	91.29	90.41	91.32	84.00

Moreover, this paper visualizes the prediction results of each model. The comparison of the prediction results is shown in Figure 14. Figure 14a is the real label, Figure 14b–f correspond to the prediction results of FCN-8S, U-Net, PSPNet, DeeplabV3+ and HMRT, respectively. Through comparison, we can find that the HMRT model proposed in this paper has a global receptive field, and the segmentation accuracy is higher than that of the comparison model. The dashed box in the figure highlights the area where the segmentation effect is obvious.

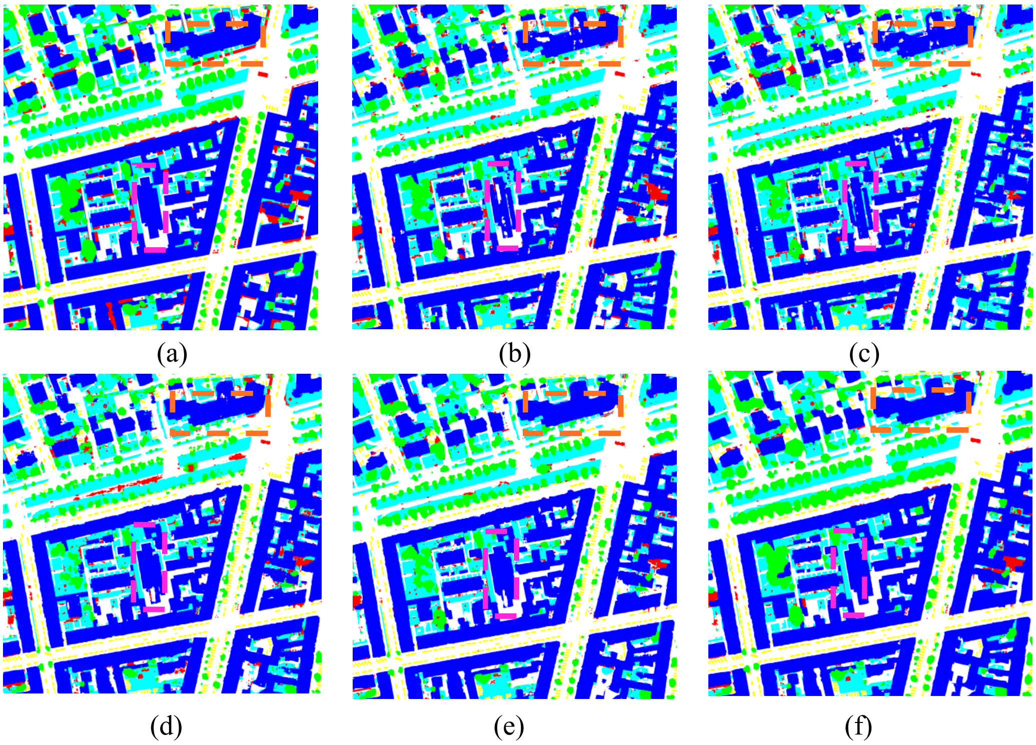


Figure 14. Comparison of the prediction results on Potsdam-B; (**a**) the superposition of the image and the label; (**b**) FCN-8S; (**c**) U-Net; (**d**) PSPNet; (**e**) DeeplabV3+; (**f**) HMRT.

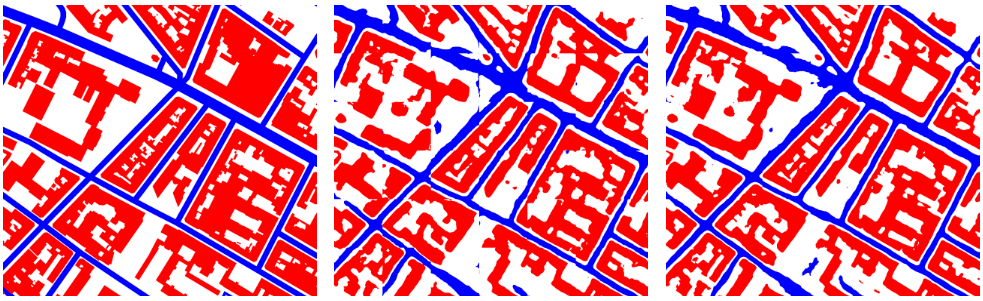
3.3.2. Quantitative Analysis of Model Prediction Results Promotion Strategy

This work adopted two post-processing methods, multi-scale fusion and sliding stitching, to improve the prediction accuracy. The multi-scale fusion strategy is to adapt to different size targets in remote sensing images, and the experimental parameters are that the predicted picture is magnified by 1.0, 1.25, 1.5, 1.75, 2.0 times, respectively. The sliding stitching strategy is to alleviate the problem of jagged edges when the pictures are directly stitched, the experimental parameter is that the step size is half of the sliding window (512 × 512). Finally, we used the controlled variable method to experiment on the two post-processing strategies, and each network obtained 4 sets of experimental results. The results of the quantitative analysis experiment are shown in Table 6.

It is concluded from Table 6 that the two strategies of both multi-scale fusion and sliding splicing can improve the prediction accuracy to a certain extent, and the prediction accuracy reaches the highest when the two strategies of multi-scale fusion and sliding splicing are used at the same time. In order to visually demonstrate the effectiveness of the post-processing strategy, Figure 15 shows the comparison between the predicted result without using post-processing strategies and predicted the result using two post-processing strategies. From the comparison of Figure 15b,c, it can be seen that the post-processing strategies reduce the stitching traces of splicing pictures, and the outlines of the foreground target buildings and roads in the picture are clearer.

Table 6. Comparison of multi-scale fusion and sliding splicing prediction quantitative analysis. √ means this post-processing method is adopted, ↑ means the higher, the better.

Methods	Multi-Scale	Sliding Stitching (%) ↑	F1 (%) ↑	OA (%) ↑	MIoU(%) ↑
FCN-8S	-	-	80.93	83.09	68.55
	√	-	81.05	83.15	68.69
	-	√	81.13	83.18	68.97
	√	√	81.41	83.35	69.22
U-Net	-	-	82.94	84.54	71.28
	√	-	83.06	84.63	71.44
	-	√	83.15	84.71	71.51
	√	√	83.25	84.90	71.73
PSPNet	-	-	83.57	84.49	72.09
	√	-	83.62	84.53	72.18
	-	√	83.71	84.61	72.25
	√	√	84.04	85.04	72.78
DeeplabV3+	-	-	84.05	85.17	72.82
	√	-	84.12	85.21	72.91
	-	√	84.18	85.25	72.97
	√	√	84.30	85.36	73.18
HMRT	-	-	84.88	85.58	74.19
	√	-	85.40	86.55	74.82
	-	√	85.48	86.31	74.90
	√	√	85.79	86.85	75.39



(a) (b) (c)

Figure 15. Comparison of prediction results before and after post-processing; (**a**) label image; (**b**) the predicted result without using post-processing strategies; (**c**) predicted result using two post-processing strategies.

4. Conclusions

This paper proposes the HMRT to extract buildings and roads from high resolution remote sensing images. In comparison with the current networks, HMRT has three advantages: (1) The multi-resolution semantic extraction branch is constructed to use branches with different resolutions for feature fusion, which ensures that high resolution and multi-resolution can always be maintained during the down-sampling process, and feature information is fully retained. It solves the problem that feature map compression leads to loss of details and the convolutional neural network lacks long-distance scene understanding when the current semantic segmentation algorithm uses a convolutional neural network (CNN) to extract image features. (2) The Transformer sequence feature extraction network is introduced through which the global receptive field of the feature map can be obtained, the long-distance dependence of the segmentation target is improved, and the issue of reduced resolution is solved, which is caused by feature map compression during the use of convolutional feature extraction. (3) The model has the following advantages, such as the highest accuracy index, absolute superiority in segmentation accuracy, and sufficient robust performance.

However, there are still some shortcomings in the segmentation of buildings and roads: (1) The use of the Transformer global receptive field to extract features is still in the development stage, so there is development space in the accuracy of the edge segmentation of buildings and roads and the structure of the model. (2) The complexity of the parameters of the Transformer encoder and decoder is high. (3) When the remote sensing image contains a lot of noise, the accuracy of segmentation will decrease. As a result, we will optimize HMRT to improve the segmentation accuracy and overcome the problem of a decrease in segmentation accuracy in case of a lot of noise in remote sensing images.

Author Contributions: Zhongyu Sun: conceptualization, methodology, writing—original draft. Wangping Zhou: conceptualization, supervision, software, writing—review and editing. Min Xia: funding acquisition, writing—review and editing. Chen Ding: formal analysis, validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of PR China (42075130).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and the code of this study are available from the corresponding author upon request.

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

References

1. Pham, H.M.; Yamaguchi, Y.; Bui, T.Q. A case study on the relation between city planning and urban growth using remote sensing and spatial metrics. *Landsc. Urban Plan.* **2011**, *100*, 223–230. [CrossRef]
2. Song, L.; Xia, M.; Jin, J.; Qian, M.; Zhang, Y. SUACDNet: Attentional change detection network based on siamese U-shaped structure. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *105*, 102597. [CrossRef]
3. Xia, M.; Qu, Y.; Lin, H. PADANet: Parallel asymmetric double attention network for clouds and its shadow detection. *J. Appl. Remote Sens.* **2021**, *15*, 046512. [CrossRef]
4. Wen, Q.; Jiang, K.; Wang, W.; Liu, Q.; Guo, Q.; Li, L.; Wang, P. Automatic building extraction from google earth images under complex backgrounds based on deep instance segmentation network. *Sensors* **2019**, *19*, 333. [CrossRef] [PubMed]
5. Behera, M.D.; Gupta, A.K.; Barik, S.K.; Das, P.; Panda, R.M. Use of satellite remote sensing as a monitoring tool for land and water resources development activities in an Indian tropical site. *Environ. Monit. Assess.* **2018**, *190*, 401. [CrossRef]
6. Qu, Y.; Xia,M.; Zhang, Y. Strip pooling channel spatial attention network for the segmentation of cloud and cloud shadow. *Comput. Geosci.* **2021**, *157*, 104940. [CrossRef]
7. Yuan, J.; Wang, D.; Li, R. Remote sensing image segmentation by combining spectral and texture features. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 16–24. [CrossRef]
8. Li, D.; Zhang, G.; Wu, Z.; Yi, L. An edge embedded marker-based watershed algorithm for high spatial resolution remote sensing image segmentation. *IEEE Trans. Image Process.* **2010**, *19*, 2781–2787. [PubMed]
9. Fan, J.; Han, M.; Wang, J. Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation. *Pattern Recognit.* **2009**, *42*, 2527–2540. [CrossRef]
10. 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 and Information Technology 2017, Helsinki, Finland, 21–23 August 2017; pp. 191–201. [CrossRef]
11. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
12. 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]
13. Sun, W.; Wang, R. Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with DSM. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 474–478. [CrossRef]
14. Liu, W.; Zhang, Y.; Fan, H.; Zou, Y.; Cui, Z. A New Multi-Channel Deep Convolutional Neural Network for Semantic Segmentation of Remote Sensing Image. *IEEE Access* **2020**, *8*, 131814–131825. [CrossRef]
15. Qi, X.; Li, K.; Liu, P.; Zhou, X.; Sun, M. Deep Attention and Multi-Scale Networks for Accurate Remote Sensing Image Segmentation. *IEEE Access* **2020**, *8*, 146627–146639. [CrossRef]
16. Li, J.; Xiu, J.; Yang, Z.; Liu, C. Dual Path Attention Net for Remote Sensing Semantic Image Segmentation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 571. [CrossRef]
17. Lan, M.; Zhang, Y.; Zhang, L.; Du, B. Global Context based Automatic Road Segmentation via Dilated Convolutional Neural Network. *Inf. Sci.* **2020**, *535*, 156–171. [CrossRef]

18. He, N.; Fang, L.; Plaza, A. Hybrid first and second order attention Unet for building segmentation in remote sensing images. *Inf. Sci.* **2020**, *63*, 140305. [CrossRef]
19. Xia, M.; Zhang, X.; Liu, W.; Weng, L.; Xu, Y. Multi-stage Feature Constraints Learning for Age Estimation. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2417–2428. [CrossRef]
20. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460. [CrossRef]
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
23. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
24. Xia, M.; Wang, K.; Song, W.; Chen, C.; Li, Y. Non-intrusive load disaggregation based on composite deep long short-term memory network. *Expert Syst. Appl.* **2020**, *160*, 113669. [CrossRef]
25. Xie, E.; Wang, W.; Wang, W.; Sun, P.; Xu, H.; Liang, D.; Luo, P. Segmenting transparent object in the wild with transformer. *arXiv* **2021**, arXiv:2101.08461.
26. 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: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
27. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229. [CrossRef]
28. 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.
29. Vaswani, A.; Shazeer, N.; Parmar, N. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
30. Kaiser, P.; Wegner, J.D.; Lucchi, A.; Jaggi, M.; Hofmann, T.; Schindler, K. Learning aerial image segmentation from online maps. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6054–6068. [CrossRef]
31. Rottensteiner, F.; Sohn, G.; Gerke, M.; Wegner, J.D. ISPRS Semantic Labeling Contest. *ISPRS* **2014**, *1*, 4.
32. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
33. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
34. 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(ECCV), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
35. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.