remote sensing

MDPI

Article

MQANet: Multi-Task Quadruple Attention Network of Multi-Object Semantic Segmentation from Remote Sensing Images

Yuxia Li [1], Yu Si [1], Zhonggui Tong [1], Lei He [2,3,*], Jinglin Zhang [1], Shiyu Luo [1] and Yushu Gong [1]

[1] School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2] School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China
[3] Sichuan Province Engineering Technology Research Center of Support Software of Informatization Application, Chengdu 610225, China
* Correspondence: helei1978@cuit.edu.cn

Abstract: Multi-object semantic segmentation from remote sensing images has gained significant attention in land resource surveying, global change monitoring, and disaster detection. Compared to other application scenarios, the objects in the remote sensing field are larger and have a wider range of distribution. In addition, some similar targets, such as roads and concrete-roofed buildings, are easily misjudged. However, existing convolutional neural networks operate only in the local receptive field, and this limits their capacity to represent the potential association between different objects and surrounding features. This paper develops a Multi-task Quadruple Attention Network (MQANet) to address the above-mentioned issues and increase segmentation accuracy. The MQANet contains four attention modules: position attention module (PAM), channel attention module (CAM), label attention module (LAM), and edge attention module (EAM). The quadruple attention modules obtain global features by expanding the receptive fields of the network and introducing spatial context information in the label. Then, a multi-tasking mechanism which splits a multi-category segmentation task into several binary-classification segmentation tasks is introduced to improve the ability to identify similar objects. The proposed MQANet network was applied to the Potsdam dataset, the Vaihingen dataset and self-annotated images from Chongzhou and Wuzhen (CZ-WZ), representative cities in China. Our MQANet performs better over the baseline net by a large margin of +6.33 OA and +7.05 Mean F1-score on the Vaihingen dataset, +3.57 OA and +2.83 Mean F1-score on the Potsdam dataset, and +3.88 OA and +8.65 Mean F1-score on the self-annotated dataset (CZ-WZ dataset). In addition, each image execution time of the MQANet model is reduced 66.6 ms compared to UNet. Moreover, the effectiveness of MQANet was also proven by comparative experiments with other studies.

Keywords: deep learning; remote sensing; semantic segmentation; multi-task learning; attention mechanism

Citation: Li, Y.; Si, Y.; Tong, Z.; He, L.; Zhang, J.; Luo, S.; Gong, Y. MQANet: Multi-Task Quadruple Attention Network of Multi-Object Semantic Segmentation from Remote Sensing Images. *Remote Sens.* **2022**, *14*, 6256. https://doi.org/10.3390/rs14246256

Academic Editors: Qian Du, Yanni Dong and Xiaochen Yang

Received: 7 November 2022
Accepted: 5 December 2022
Published: 10 December 2022

1. Introduction

Semantic segmentation tasks classify each pixel in an image into several regions with specific semantic categories and often appear in fields such as human–computer interaction, computer photography, image search engines, and augmented reality. In these applications, the extraction targets are usually clear in semantics and have a small coverage area. However, things are different in remote sensing images. The targets in remote sensing images have a wider range of distribution and more complex features. On the one hand, such characteristics provide richer target detail information for feature detection, such as color, contour, and texture. On the other hand, much more complex interference is introduced into segmentation tasks.

The traditional remote sensing image segmentation methods mainly include support vector machine (SVM) methods [1], superpixel-based methods [2], and semisupervised geodesic-based methods [3]. Traditional methods can achieve better results with a small sample size. Still, with the increase in sample data, the accuracy of traditional methods has not significantly improved to meet the application requirements. Usually, traditional methods are only effective for some specific scenarios, with poor universality.

Focused on the above situation, compared with traditional methods in machine learning, Deep Convolutional Neural Networks (DCNNs), such as FCN [4], have shown excellent feature extraction and object representation abilities [5]. Many approaches have been proposed to increase the receptive field of convolutional neural networks. Unet [6] and SegNet [7] propose skip connection, trying to connect the same-sized feature maps in the encoder and decoder layers. The DeepLab series network models are all based on encoder–decoder architecture. Atrous convolution to expand the receptive field is introduced in DeepLab v1 [8] and DeepLab v2 [9]. Furthermore, Atrous Spatial Pyramid Pooling (ASPP) was proposed to expand the perceptual field further and enhance the spatial feature extraction capability using multi-layer cavity convolution. DeepLab v3 [10] introduced image-level multiscale features in ASPP to further improve the feature extraction capability. DeepLab v3+ [11] used a modified Xception [12] encoder and a lightweight decoder to improve the resolution of segmentation results. FarSeg [13] uses two encoder branches to enhance the extraction of foreground and background, respectively. In addition, some studies have been conducted to improve the deep learning methods according to the characteristics of remote sensing images. EFCNet [14] introduces the separable convolutional module (SCM) to alleviate the problem of numerous parameters for the semantic segmentation of high-resolution remote sensing images. DSPCANet [15] introduces the internal residual block (R2_Block) to enhance the receptive field of the network and learn the ground feature representation from additional DSM images. Sharifi et al. [16] proposed the ResUNet-a model with post-processing. The model contains the output of joint connection, which significantly improves the efficiency and robustness of the farmland extraction model. However, stacking and aggregating convolutional layers perform poorly in covering global receptive fields. Additionally, these methods do not effectively extract global contextual information.

A helpful method for obtaining global contextual information is the self-attention mechanism. Nonlocal [17] proposes a generalized, simple, nonlocal operation operator that can be directly embedded into neural networks. DANet [18] introduces the self-attention mechanism to capture feature dependencies in the spatial and channel dimensions. A2-Nets [19], expectation–maximization attention networks [20], and CBAM [21] introduced a self-attention mechanism to merge global features by different descriptors. LANet [22] proposes a patch attention module and an attention embedding module to merge high-level and low-level features in the model. HMANet [23] proposes a class augmented attention module to obtain class-level information. SSFTT [24] utilizes the self-attention mechanism to construct 3D and 2D convolutional layers to jointly extract shallow spectral and spatial features. However, methods that rely on self-attention only cause the network to pay attention to itself and overlook the spatial context relationship hidden in the labels.

To solve the above-mentioned issues, we suggest a brand new structure termed the Label Attention Module (LAM). LAM fully uses the label's spatial context via the attention module. However, the way to generate attention is different than the self-attention module. LAM optimizes the attention probability map by introducing label information.

Furthermore, we proposed a triple-attention network called TANet, which contains LAM and two self-attention modules presented in DANet: PAM and CAM. TANet can help enhance semantic segmentation accuracy due to the triple attention module's ability to strengthen global information extraction.

Adding LAM can significantly increase the segmentation accuracy of a large range of targets. However, misjudgment problems would appear with respect to some targets presenting similar features, such as impervious surfaces and concrete roof buildings. Due

to the competition between different categories, the probability of the misjudgment category affects the real category. In order to reduce the competition between categories, we introduced a multi-task mechanism. The multi-task mechanism converts a multi-category segmentation task into multiple binary-classification segmentation tasks. All categories share an encoder, and each category owns a separate decoder. The multi-task architecture can effectively improve the segmentation accuracy of similar categories. Furthermore, we perform some edge optimization to improve the accuracy of the edge area of different categories. The edge optimization includes two new edge branches and an edge attention module. Combined with the triple attention and multi-task architecture mentioned above, the model contains four attention modules, so we name this network Multi-Task Quadruple Attention Network. We conducted experiments on two public datasets (Potsdam and Vaihingen datasets) and a self-made dataset (CZ-WZ dataset) to demonstrate the effectiveness of the proposed model.

The main contributions of this paper are as follows:

(1) We propose the label attention module (LAM) to learn the spatial contextual information of features from the label instead of information from the network itself.
(2) A Triple Attention Network is designed to obtain global features of large objects. It significantly improves the semantic segmentation accuracy of large objects in remote sensing images.
(3) A Multi-task TANet (MTANet) architecture is proposed to reduce the misjudgment between similar categories.
(4) Based on the MTANet model, A MQANet model is constructed to optimize the edge area of semantic segmentation.

2. Related Search

2.1. Semantic Segmentation

Semantic segmentation divides images into regional blocks with certain semantics and obtains a segmented image with pixel-by-pixel semantic annotation. Traditional segmentation methods focus on designing a feature descriptor for each pixel. However, the specially designed handcrafted feature descriptors are challenging to adapt to other scenarios.

Deep learning methods extract features directly from the data itself. Patch-wise classification achieves semantic segmentation based on the classification results of patches. However, this method is too expensive to compute. FCN [4] proposes a paradigm to obtain semantic segmentation directly from feature upsampling, significantly improving the computational cost. After FCN was proposed, various structures were proposed to improve the segmentation accuracy. Most networks use encoder–decoder architectures, such as Unet [6], SegNet [7], and DeeplabV3+ [11]. Encoders are used for feature extraction, and decoders perform pixel-level classification based on the features obtained by the encoder to obtain semantic segmentation results. Furthermore, Zhou et al. [25] introduced D-LinkNet with multiscale dilation rates to collect contextual information and extract additional global features. GPSNet [26] tries to dynamically aggregate a discriminative semantic context using a comparative feature aggregation module. Furthermore, it can gather free-form semantic context information adaptively. CGNet [27] constructs the joint feature from local features and the surrounding context effectively and efficiently. In addition, CGNet utilizes the global context to improve the joint feature.

2.2. Attention Mechanism

The human visual mechanism inspired the formation of the spatial attention mechanism. When the human eye sees an image, it will automatically give greater attention to critical locations. Therefore, different components of the image feature map should have diverse weights.

Vaswani et al. [28] introduced the self-attention mechanism, which does not use complex models like RNN or CNN, and it only relies on the attention model to parallelize training with global information. Self-attention obtains contextual dependency information

at a long distance by capturing the spatial dependencies between any two locations in the feature map. Specifically, assume the standard form of each element in the sequence is (Q, K, V). Q denotes query, K denotes key, and V denotes value. In DCNNs, we generate Q, K, and V from feature maps extracted by the networks' backbones. The attention mechanism calculates attention weight by the similarity between Q and K. Then, attention output comes from the V value and the attention weight's weighted summation. Self-attention differs from standard attention mechanisms: three matrices—the question, key, and value—are identical in the self-attention method. The attention output result is shown in Equation (1). The output of the attention module integrates features from Q, K, and V. A network might benefit from the attention module by catching more global context over local features and enhancing the pixel-level prediction's feature representations.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where d_k denotes the normalization coefficient, and K^T refers to the transposition of K.

DANet [18] is a classic self-attention network application that introduces a self-attention mechanism to obtain the feature dependence separately in the spatial and channel dimensions. BAM [29] performs better in element-wise summation of spatial and channel attention. PSANet [30] is designed with a bidirectional information dissemination path, where each location aggregates information from all other locations to help predict itself. Unlike previous works, we propose a label attention module to capture global contexts from the label directly, ensuring that the spatial context information can be smoothly transferred to the attention map.

2.3. Multi-Task Learning

Initially introduced by Caruana [31], multi-task learning aims to solve multiple tasks with one architecture. In order to theoretically explain that multi-task learning has better learning effects than single-task learning, relevant papers give theoretical proof in different aspects. For example, the inductive bias learning method proposed by Baxter et al. [32] proved that multi-task learning could achieve better generalization errors than single-task learning in a specific environment. Ben-David [33] proposed a more general concept of connection between tasks, which can be used in many real-life scenarios. This concept proved the strict upper bound of the generalization error of multi-task learning. Maurer [34] applied dictionary learning and sparse coding to multi-task learning and gave the generalization error of multi-task learning by measuring the complexity of the hypothesis. Ando and Zhang et al. [35] assume that all tasks share a standard structure and give reliable estimates of the shared parameters of multiple tasks when there are enough tasks.

Multi-task learning in deep neural networks can be divided into two categories based on whether parameters are shared softly or firmly between tasks [36].

Soft parameter sharing allows for the regularization of different tasks' bottom parameters instead of keeping the bottom parameters the same. For example, Dung et al. [37] apply the L2 norm to restrict the distance between the parameters of different tasks to ensure the relevance between tasks. Yang et al. [38] proposed regularizing bottom parameters with a trace norm. Compared with the complex parameter-constrained multi-task deep learning model, the soft-constrained multi-task learning model has fewer constraints. It can produce better outcomes when the task association is not quite tight. Multi-task deep learning models need to learn a network architecture suitable for multiple tasks simultaneously. Thus, the models are more robust and resistant to overfitting.

The hard sharing method is another multi-task learning method. This method assumes that there is some connection between the different tasks. As in Figure 1, different tasks obtain standard low-level features in the shared bottom layers and confirm the uniqueness of various tasks; each task uses a unique layer to capture high-level features. Our work transforms a multi-classification task into multiple binary classification tasks with high similarity between tasks, so we choose to use hard sharing as an approach.

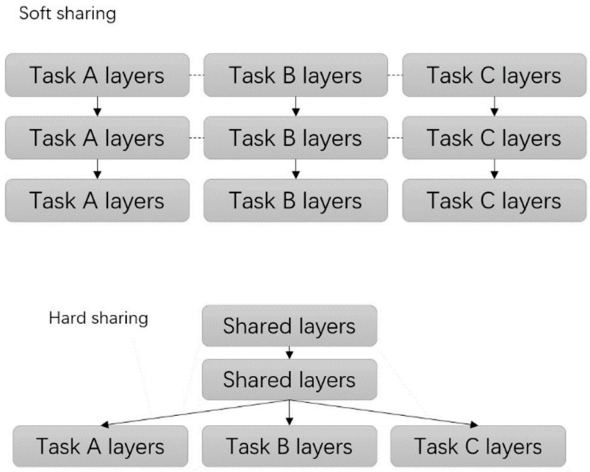

Figure 1. Two different parameter-sharing methods in multi-task learning.

3. Methodology

The proposed network model consists of encoder, decoder, and edge optimization module. The encoder structure is the same as the VGG architecture adopted by UNet. For the decoder, the proposed network converts a multi-category semantic segmentation task into multiple binary-segmentation tasks. The number of decoders is the same as the number of semantic segmentation categories of the ground object. Each decoder contains quadruple attention modules, PAM, CAM, LAM, and EAM. The optimization algorithm for edge extraction consists of three parts, the edge map branch, the distance map branch, and the edge attention module. The architecture of the proposed method is shown in Figure 2.

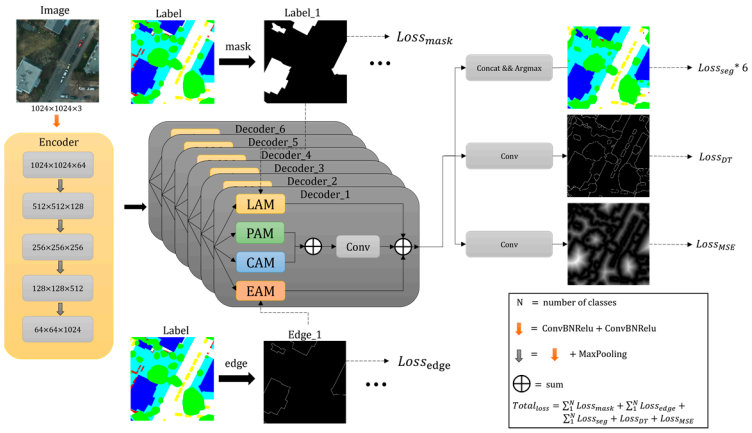

Figure 2. An overview of the Multi-task QANet (MQANet).

Sections 3.1 and 3.2 describe the PAM, CAM, and LAM. In Section 3.3, we introduce the multi-task architecture of this network. In Section 3.4, edge optimization is introduced, including two edge map branches and the EAM.

In Figure 2, all categories share one encoder, and each category owns a separate triple attention decoder. Each decoder contains four attention modules. LAM needs label information, and EAM needs edge labels during training.

3.1. PAM and CAM

Similar to human learning, machine learning is considered attention. The attention mechanism's core goal is to find critical information from various information items for the current task. Position Attention Module (PAM) and Channel Attention Module (CAM) are practical self-attention modules. PAM captures spatial global dependencies, and CAM pays attention to the importance of each channel dimension.

Figure 3 illustrates the structure of the PAM. In the PAM, the input generates two parts of feature maps: one is represented as Q and K to calculate an attention probability map in the shape $(H \times W) \times (H \times W)$, and the other is used as V. Q, K, and V denote query features, key features, and value features. Furthermore, C, H, and W represent the attention probability map's channel, height, and weight. Then, the optimization attention map is reshaped to obtain the final prediction map.

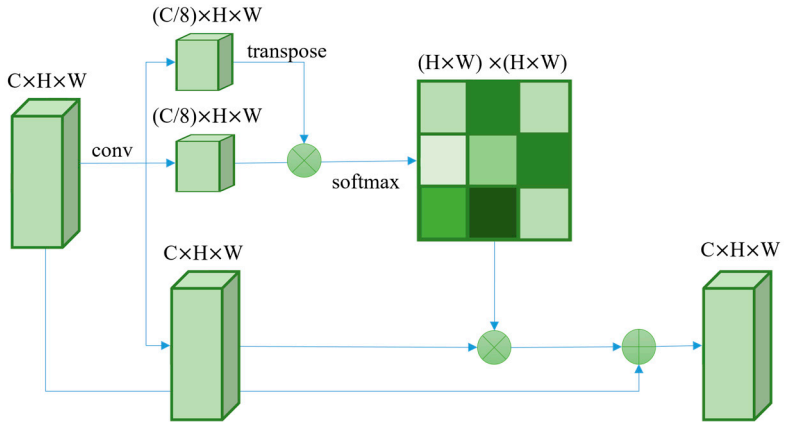

Figure 3. Architecture of Position Attention Module [18].

The overall structure of the PAM is shown in Equation (2).

$$\begin{aligned} Att &= Softmax\left(Q_{(HW\times C)} \cdot K_{(C\times HW)}\right) \\ F_{out} &= \left(V_{(C\times HW)} \cdot Att\right) reshape(C \times H \times W) + Input_{(C\times H\times W)} \end{aligned} \tag{2}$$

where Att denotes the attention probability map calculated, and F_{out} is the final output, obtained by summing the input and the optimization attention map. $reshape(C \times H \times W)$ refers to the size of the feature map reshape to $C \times H \times W$.

The width and height of the feature are multiplied by a large value and reduce the number of channels to one-eighth of the original one by using convolution to facilitate operations. Then the dimension of the number of channels is eliminated by matrix multiplication. This operation does not affect the shape of the feature.

PAM uses a spatial attention map to select aggregating contexts. In addition, PAM has a global contextual view. Similar semantic features enhance intra-class compactness and semantic consistency [18].

The CAM is similar in structure to the PAM, but it has a few differences. The structure of the CAM is given in Figure 4. The first difference is that the number of channels is smaller, so there is no need to change the feature map's shape using convolution to reduce the number of operations. The other point is that the shape of the generated attention weight map is changed, and CAM focuses on the connection between the different channels of the features. In the network structure, CAM swaps the position of the location attention module dot product, and the shape of the generated attention weight map is $(C \times C)$, thus establishing the influence relationship between features in different channels.

The overall structure of CAM is shown in Equation (3).

$$\begin{aligned} Att &= softmax\left(Q_{(C\times HW)} \cdot K_{(HW\times C)}\right) \\ F_{out} &= \left(Att \cdot V_{(C\times HW)}\right) reshape(C \times H \times W) + Input_{(C\times H\times W)} \end{aligned} \tag{3}$$

where Att denotes the attention probability map calculated, and F_{out} is the final output, obtained by summing the input and the optimization attention map. $reshape(C \times H \times W)$ refers to the size of the feature map reshape to $C \times H \times W$.

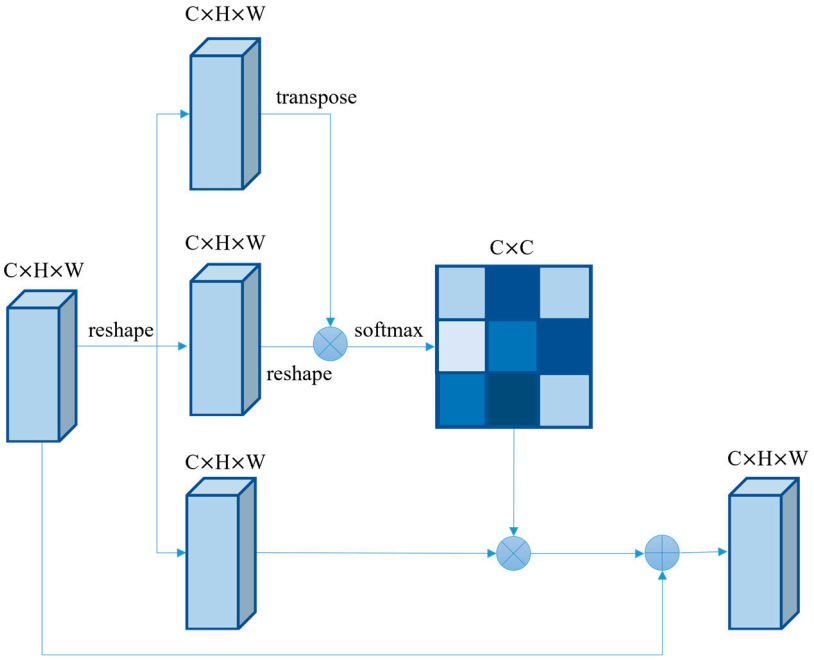

Figure 4. Architecture of Channel Attention Module [18].

3.2. Label Attention Module (LAM)

We built a brand-new attention mechanism at different views, inspired by the effectiveness of attention-based methods. Unlike the self-attention mechanism, we use the label to generate attention probability maps. Thus, LAM can gather more global features. The structure is shown in Figure 5.

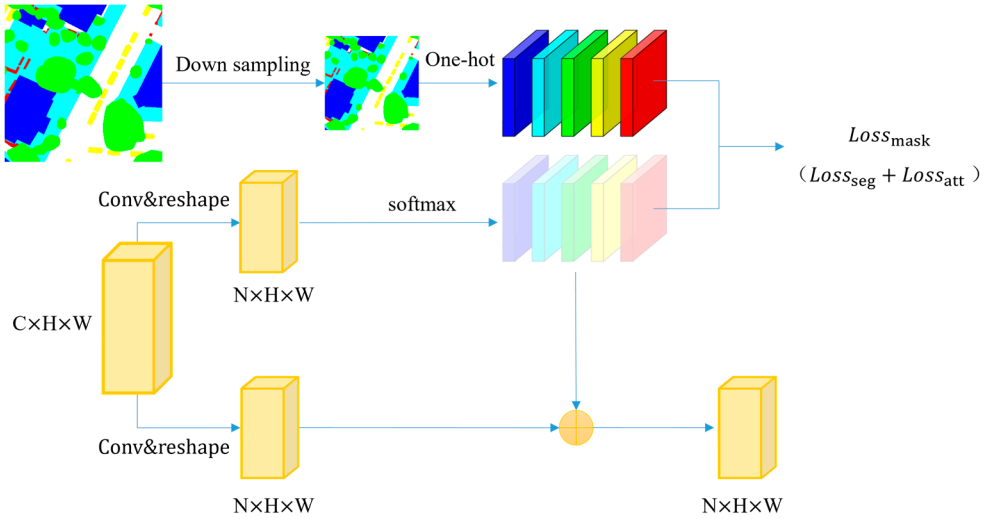

Figure 5. Architecture of Label Attention Module.

The input ($C \times H \times W$) is turned into two parts shaped ($N \times H \times W$) by convolution and reshaping, where N is the number of classifications. LAM's output is the weighted summation of the attention and value parts. In addition, after the SoftMax function, a loss function is computed the attention probability map and a reshaped one-hot label.

$$\begin{aligned} Att &= softmax(Conv\left(Input_{(C\times H\times W)}\right)_{(N\times H\times W)}) \\ F_{out} &= Conv(Conv(Input_{(C\times H\times W)})_{(N\times H\times W)} + Att)_{(N\times H\times W)} \end{aligned} \tag{4}$$

where Att denotes the attention probability map, and F_{out} denotes the output features of the attention module.

The convolution neural network's backpropagation technique parameter optimization is Equation (5). The loss function is shown in Equation (6).

$$W = W - \frac{\eta}{batch\_size} \sum \frac{\partial L_{seg}}{\partial W} \tag{5}$$

$$\begin{aligned} Loss_{seg} &= CE(predict, label) \\ Loss_{att} &= CE\left(predict, label_{down\_sampling}\right) \\ W &= W - \frac{\eta}{batch\_size} \sum \left(\frac{\partial L_{seg}}{\partial W} + \frac{\partial L_{att}}{\partial W}\right) \end{aligned} \tag{6}$$

where η is the learning rate, and $\frac{\partial L_{seg}}{\partial W}$ is the derivative of the loss function with respect to the parameters of the layer. CE refers to the cross-entropy loss function.

The loss function $Loss_{mask}$ consists of two parts: the segmentation part and the label attention part, which are, respectively, defined as $Loss_{seg}$, $Loss_{att}$. The label attention loss helps LAM generate prediction map prototypes that facilitate the transfer of feature information.

3.3. Multi-Task TANet

The attention mechanism helps improve the semantic segmentation accuracy of large objects. However, with the attention model it is difficult to distinguish some similar features. Some trees without leaves are very similar to low vegetation (Figure 6), and even humans will misunderstand them without carefully checking. Thus, the attention module failed to distinguish the tree area.

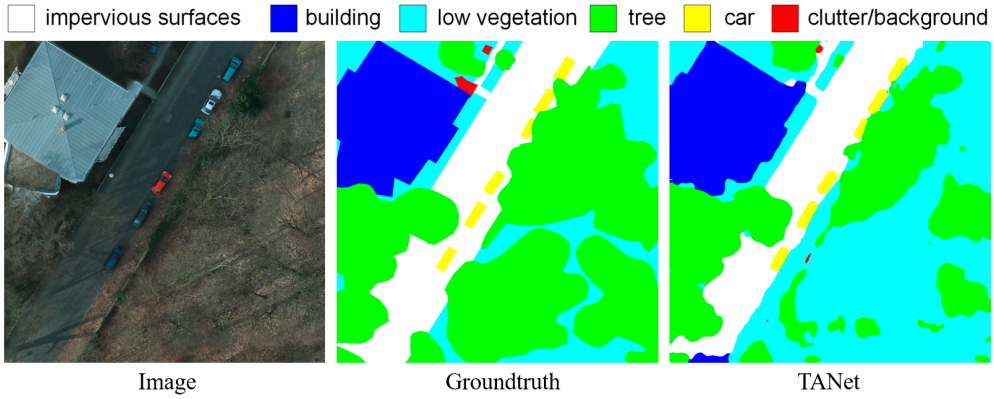

Figure 6. The features of the trees in the red rectangular area are similar to those of low vegetation, and TANet misjudged the area as low vegetation.

In the traditional encoder–decoder structure, one decoder generates probability maps of multiple output results by the Softmax function in Equation (7). For each pixel, the category with the highest probability is determined as the classification result for this point.

$$\begin{aligned} Softmax(Z_i) &= \frac{e^{z_i}}{\sum_{c=1}^{C} e^{z_c}} \\ c_i &= argmax(Softmax(Z_i)) \end{aligned} \tag{7}$$

where Z_i denotes the input vector to the softmax function, and C is the number of categories classified.

However, there is inter-class competition between different categories of each pixel. The sum of the probabilities of all categories is 1, and different categories share a decoder to restrict each other. If the probability of a particular point after decoding by the decoder is relatively uniform, it is easy to cause misjudgments. Three solutions are proposed to solve this problem.

The first and most straightforward idea is to change the loss function, using multiple binary-classification sigmoid cross-entropy loss instead of SoftMax cross-entropy loss. We call this model TANet with multiple losses. The loss function is shown in Equation (8).

$$\begin{aligned} p_i &= Sigmoid(Z_i) \\ Loss_{seg} &= \sum_i^n BCE(p_i, label) \\ Loss_{att} &= \sum_i^n BCE\Big(att, label_{down\_sampling}\Big) \\ Loss &= Loss_{seg} + Loss_{att} \end{aligned} \tag{8}$$

where p_i denotes the output of the inference model, and att denotes the attention map of LAM. BCE denotes the binary cross entropy loss.

The second method is Multi-model TANet. This method is to train a semantic segmentation network for each category and combine all the binary segmentation results. It is not easy to merge the results of multiple models. Here, we use the most intuitive method to combine the predicted probability maps of multiple binary segmentation models, and each pixel takes the category with the highest probability.

Another considerable solution is introducing multi-task learning, which converts a multi-category segmentation task into multiple binary-classification segmentation tasks. We call this method Multi-task TANet. The loss function is the same as TANet with multiple losses. All categories share an encoder, and each category owns a separate decoder. As shown in Figure 2, we use the TANet decoder mentioned above. For each pixel, we use the probability of output for each category as the confidence level and select the category with the highest confidence level as the classification result of the pixel.

For the above three methods, we have conducted experiments to find the best model.

3.4. Edge Optimization

In order to further improve the accuracy of the extracted edges, we made some targeted improvements to the model. Firstly, inspired by [39], we add a new branch of edge extraction. The purpose of this branch is to obtain edge maps of different categories. By computing loss between the edge branch extraction edge result and the edge truth map, the edge bifurcation of the extraction results can be improved.

However, the standard semantic segmentation loss function, like cross-entropy, is unsuitable for computing edge map loss. The extraction result of the edge map may have a slight misalignment with the actual value map, and the standard loss function will produce a significant deviation. Hence, a suitable loss function for the edge map must be selected. We changed the loss function from cross entropy to DT Loss [40].

DT Loss uses distance transform (DT), which transforms an edge map into a distance map. In our task, the DT Loss can be represented in Equation (9).

$$L_{DT} = \sum\nolimits_{(p_{jk} != l_{jk})} I_{DT}(j,k) \tag{9}$$

where I_{DT} denotes distance map, and j, k denotes the pixel coordinates. p_{jk} and l_{jk} represent the predicted edge map and the actual edge map.

Inspired by the distance transform method, we can transform the discrete edge map into a continuous distance map. Since the distance map already contains edge information, we can directly add a new branch to predict the distance map, and the loss function can be the common MSE.

Furthermore, an edge attention module (EAM) is added to each decoder. As shown in Figure 7, the overall architecture of EAM is consistent with of LAM, the difference is that the attention probability map used in EAM is the edge map of labels.

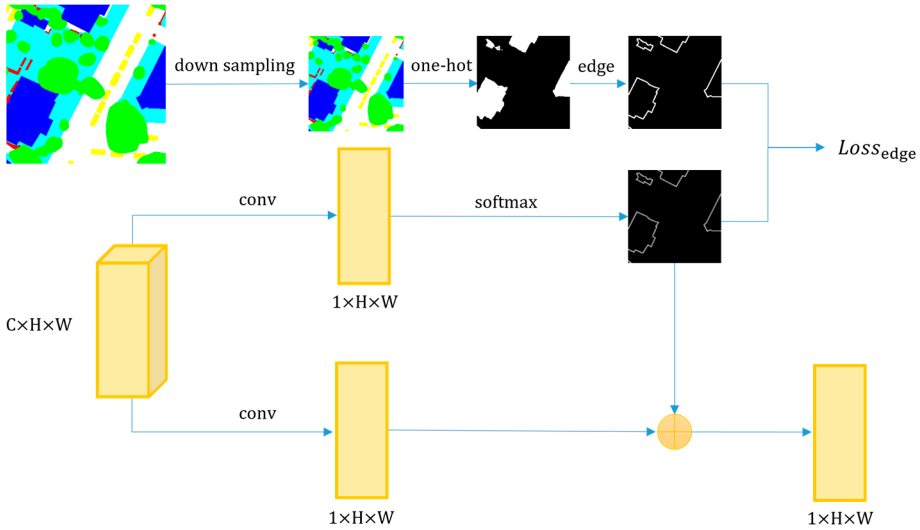

Figure 7. Architecture of Edge Attention Module.

The loss function $Loss_{edge}$ is just the edge map attention part. Structure and loss of EAM is shown as follows.

$$att_{edge} = Softmax\left(Conv(Input_{C\times H\times W})_{1\times H\times W}\right) \tag{10}$$

$$F_{out} = Conv(Conv\Big(Input_{(C\times H\times W)}\Big)_{(1\times H\times W)} + Att\Big)_{(1\times H\times W)} \tag{11}$$

$$Loss_{edge} = CE\Big(att_{edge}, edgemap_{down\_sampling}\Big) \tag{12}$$

where Att denotes the attention probability map, and F_{out} denotes the output features of the attention module. EAM's output is the weighted summation of the attention and value parts.

Using the boundaries of different feature categories in the labels, an edge map is drawn as the guiding information of the edge attention probability map, and the rest of the model is consistent with all the label attention modules. Since the model contains multiple decoders, each corresponding to a feature class, the edge attention module in each decoder has the same basic structure.

By combining these three improvements, the edges of the extraction results can be further improved.

3.5. Descriptions of Datasets

In order to demonstrate the effectiveness of the model, several remote sensing image semantic segmentation datasets are used in this paper. On the one hand, we use two publicly available ISPRS remote sensing image semantic segmentation datasets. On the other hand, a self-made CZ-WZ multi-category semantic segmentation dataset is used. The differences in resolution and image sensors between the two datasets are significant, so experiments on the three datasets separately can verify the model structure's effectiveness in different data styles.

Firstly, the Potsdam 2D semantic labeling dataset [41] contains 38 patches, each consisting of a true orthophoto (TOP) extracted from a larger TOP mosaic. The label contains six categories: impervious surfaces, buildings, low vegetation, trees, cars, and clutter/background. Four spectral bands exist in each TOP image (red, green, blue, and near-infrared), and we only use the RGB channels in this work.

The second dataset is the Vaihingen dataset [42], which contains 33 TOP patches of different sizes. The ground sampling distance of the TOP is 9 cm. The reference data are divided into the same six categories as the Potsdam dataset. Each TOP image has three spectral bands (red, green, and near-infrared).

The third dataset is the CZ-WZ dataset. The original images come from the Changzhou area in Sichuan Province and the Wuzhen area in Zhejiang Province, China. The spatial resolution of the experimental remote sensing images is 0.51 m. Each TOP image has three spectral bands (red, green, and blue). The label is self-made and classifies the remote sensing image features into five categories: buildings, roads, vegetation, water bodies, and backgrounds.

Each training image is cut into 1024 × 1024 patches. After cutting, the Potsdam dataset contains 1176 training samples and 504 test samples; the Vaihingen dataset contains 221 training samples and 63 test samples; the CZ-WZ dataset contains 1080 training samples and 260 test samples. The training of convolution network models usually requires a large number of samples. Based on the existing dataset, this work increases the network training samples through data enhancement. The original and label images are flipped horizontally or vertically, cut at random positions, and randomly transformed HSV. Data enhancement changes the number of training samples to five times the original.

3.6. Evaluation Metrics

Following the evaluation method used in the literature [22], we evaluate the performance of methods by three metrics: overall accuracy (*OA*), per-class *F*1 score, and average *F*1 score. *OA* is the ratio of the number of correct pixels to the total number of pixels. *F*1 score for classification is calculated as the harmonic mean of precision and recall [22].

We calculate the *F*1 score for each foreground category to assess the proposed network's performance. We also calculate the *OA* for the whole dataset. The calculation formula is as follows.

$$recall = \frac{TP}{TP + FN}, precision = \frac{TP}{TP + FP} \tag{13}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}, mean\_F1 = \sum_{i}^{N-1} F1_i / N \tag{14}$$

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

with the following terms: True Positive example (*TP*), False Positive Example (*FP*), True Negative example (*TN*), False Negative example (*FN*).

4. Results and Discussion

In this section, we validate the effectiveness of the proposed attention modules and the multi-task framework. Firstly, we use UNet as our baseline and then utilize an ablation study to show the tests of the proposed triple attention modules. Then, we experimented with the three methods above to find the best model.

4.1. Ablation Study of Triple Attention Modules

The proposed TANet contains PAM, CAM, and LAM, three attention modules. PAM and CAM are self-attention modules proposed by DANet, and LAM is proposed by our paper which is a label attention module. In order to verify the effectiveness of the three attention modules, we replaced UNet's decoder with PAM + CAM, LAM, and TANet (PAM + CAM + LAM).

4.1.1. Experiments Results on Potsdam Datasets

The attention modules focus on targets with a large area and a wide distribution range, such as buildings, low vegetation, and impervious surfaces. As shown in Table 1, after replacing the decoder with PAM and CAM, the F1 scores of impervious surfaces increases by 0.85%, and building F1 score increases by 2.13%. As for LAM, the F1 score for building increases by 1.24% and that for tree by 0.51%. This is because TANet combines the advantages of the previous two models and thus achieves the best classification results. TANet increases F1 of impervious surfaces, buildings, low vegetation, and trees compared

with UNet by 1.80%, 2.48%, 0.85%, and 0.72%, respectively. An interesting result is that UNet obtains the highest score for the segmentation result of the car. Since the attention module in this article pays more attention to large-scale and complex targets, there is no noticeable improvement for small targets such as cars.

Table 1. F1 score, Mean F1, and OA index results of multi-object on Potsdam dataset.

Method	Per-Class F1 Score (%)					Mean F1 (%)	OA (%)
	Imp. Surf.	Building	Low Veg.	Tree	Car		
UNet	87.91	91.31	81.76	82.72	**88.91**	86.52	85.48
PAM + CAM	88.76	93.44	82.09	82.21	88.21	86.94	86.41
LAM	88.34	92.45	81.74	83.23	88.01	86.75	86.06
TANet	**89.71**	**93.79**	**82.61**	**83.44**	88.78	**87.67**	**87.23**

4.1.2. Experiment Results on Vaihingen Datasets

We perform experiments on the ISPRS Vaihingen benchmark to assess TANet's performance further. We used the same training and testing setup in the experiments on the Vaihingen dataset. As shown in Table 2, TANet increases the F1 score of impervious surfaces, buildings, low vegetation, trees, and cars compared with UNet by 3.43%, 3.77%, 8.63%, 1.45%, and 9.15%. In this experiment, the performance of using the LAM decoder alone is closer to TANet, which means LAM played a more critical role in this model.

Table 2. F1 score, Mean F1, and OA index results of multi-object on Vaihingen dataset.

Method	Per-Class F1 Score (%)					Mean F1 (%)	OA (%)
	Imp. Surf.	Building	Low Veg.	Tree	Car		
UNet	84.45	87.32	69.77	83.16	63.12	77.56	81.27
PAM + CAM	86.13	88.88	74.43	82.54	71.40	80.68	82.83
LAM	87.85	91.01	78.25	**84.73**	70.85	82.51	85.06
TANet	**87.88**	**91.09**	**78.40**	84.61	**72.27**	**82.85**	**85.26**

4.1.3. Experiments Results on CZ-WZ Datasets

We conducted experiments on CZ-WZ datasets to further evaluate the effectiveness of TANet. As shown in Table 3, in terms of individual category accuracy, it is evident that the improvement is excellent for roads and water bodies. Using UNet as a benchmark, the F1 scores for roads are improved by 9.16%, 12.16%, and 13.41% for PAM + CAM, LAM, and TANet, respectively. This is because UNet cannot effectively learn the global features for this type of distribution with a large range and infrequent sample occurrence, which results in lower accuracy. Moreover, after the attention mechanism is introduced, the extraction accuracy of roads is significantly improved, which shows that the global features are very effective for road extraction. Like roads, water bodies also belong to the category with a larger distribution range and lower frequency of occurrence. They are improved by 6.20%, 6.26%, and 6.59% relative to UNet, PAM + CAM, LAM, and TANet, respectively. Vegetation was also entered as a feature type with a large distribution range. However, vegetation occurred in the sample at a high frequency, so the extraction accuracy of each scheme exceeded 90%. For the vegetation category, PAM + CAM has more improvement than LAM, while LAM has more improvement on roads and water bodies. The two schemes have some overlapping parts for the overall accuracy improvement. However, there are still differences in different categories, combining the advantages of TANet and fusing the two to obtain optimal accuracy.

Table 3. F1 score, Mean F1, and OA index results of multi-object on CZ-WZ dataset.

Method	Per-Class F1 Score (%)				Mean F1 (%)	OA (%)
	Road	Building	Veg.	Water		
UNet	60.75	74.13	90.26	77.32	75.62	81.41
PAM + CAM	69.91	77.22	91.05	83.52	80.43	83.46
LAM	72.91	**77.82**	90.87	83.58	81.30	83.35
TANet	**74.16**	77.31	**91.20**	**83.91**	**81.65**	**83.96**

We compare the results before using the proposed module and after in Figures 8 and 9. It can be clearly observed that the ability to recognize a wide range of object types is enhanced after combining the attention module. For example, in the second row in Figure 8, part of the impervious surface area is covered by shadows. TANet recognizes the impervious surfaces correctly after the global features introduced by the attention module's introduction.

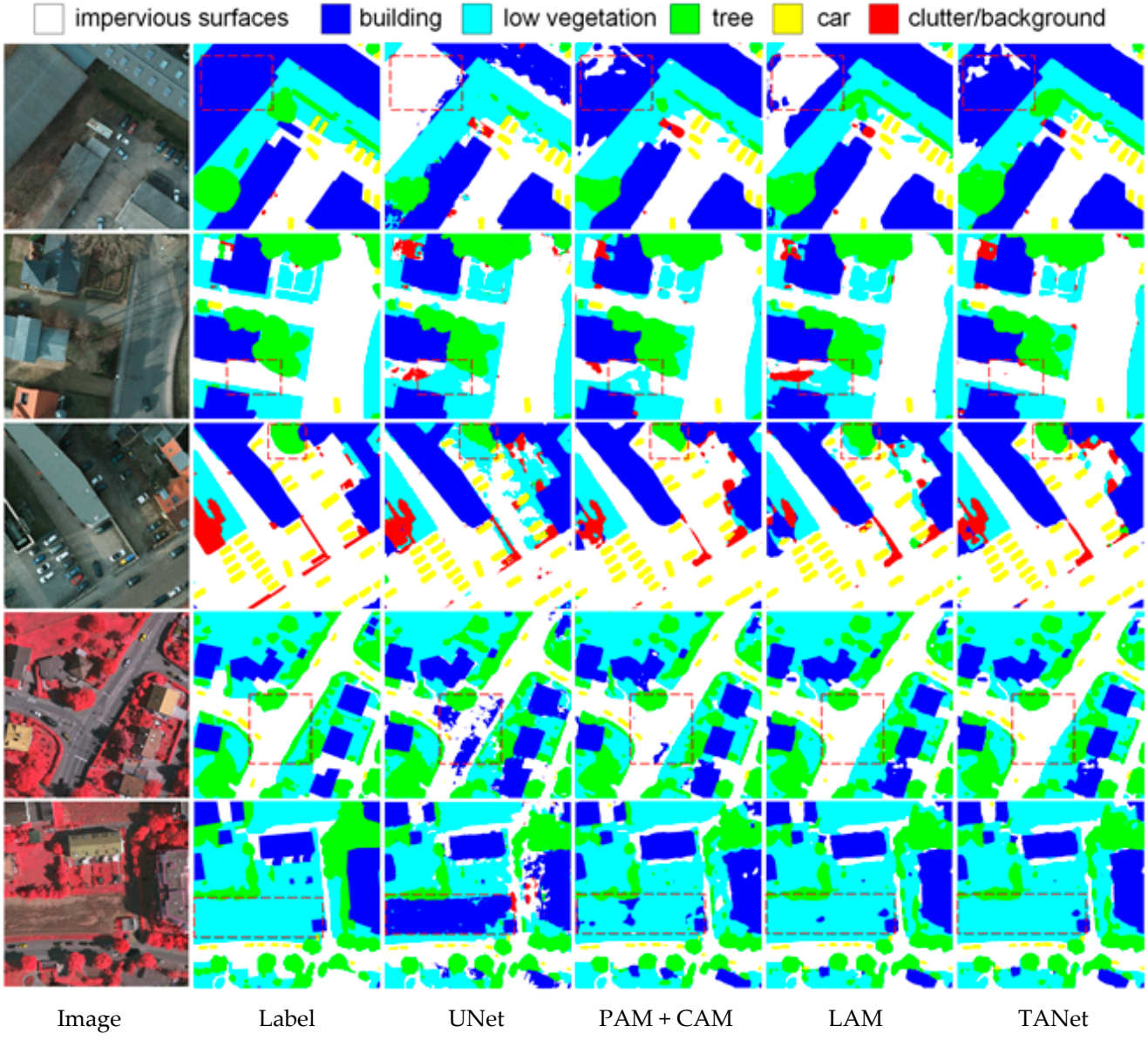

Figure 8. Results on the ablation study of triple attention modules. The first three rows are from the Potsdam dataset and the other is from the Vaihingen dataset.

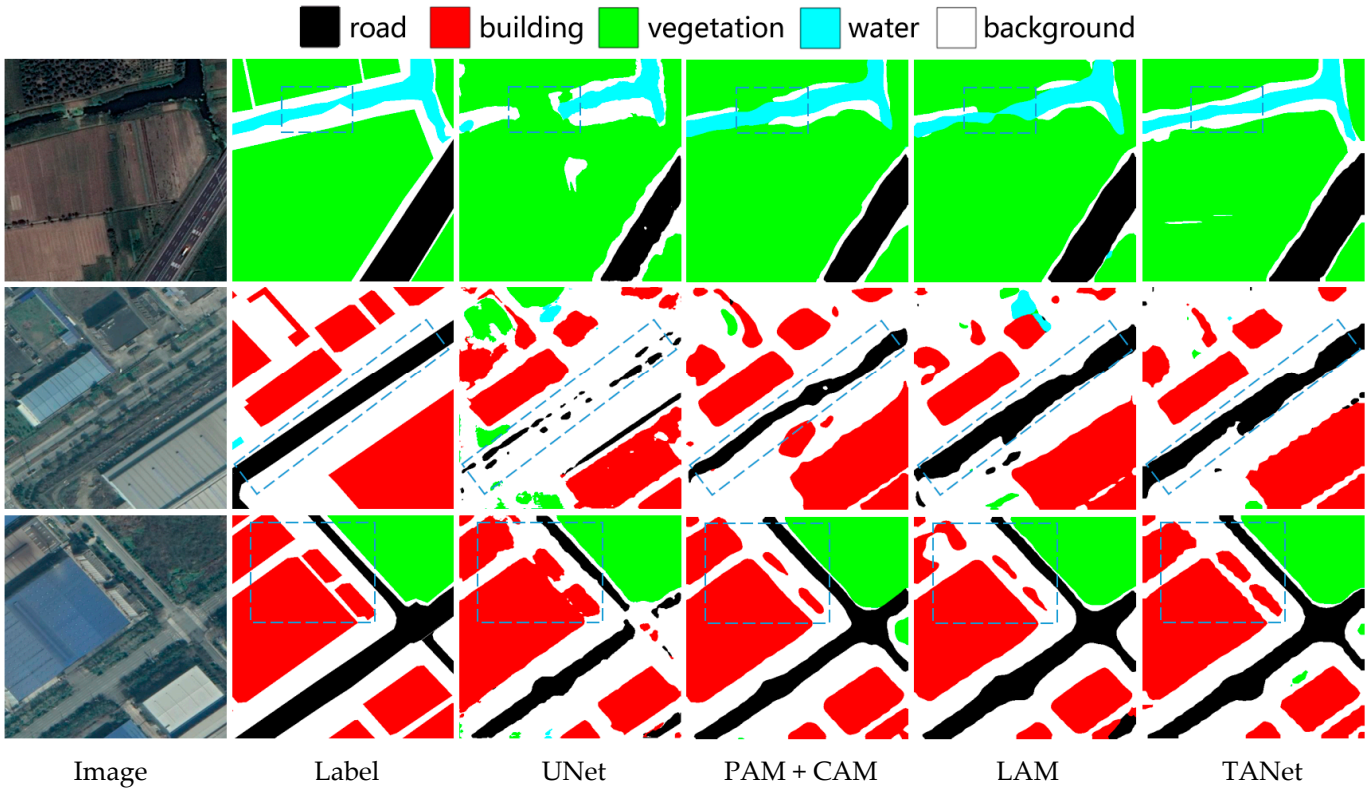

Figure 9. Examples of semantic segmentation results on ablation study of triple attention modules (CZ-WZ dataset).

4.2. Visualization of LAM

We visualize attention maps in LAM to better understand our attention modules. The number of channels of attention maps generated by LAM is the same as the number of segmentation categories, and we overlay the results of each category of the attention map on the original image. Figure 10 shows the attention map results of impervious surfaces, buildings, and cars. The feature response extracted by the attention modules is similar to the segmentation result. Moreover, attention maps of LAM have made some corrections to PAM and CAM extraction results. In the red rectangle, the segmentation results of PAM and CAM are wrong, and with the help of the LAM's attention map, TANet obtains the correct recognition result.

4.3. Ablation Study of Multi-Task Learning

After the introduction of the attention module, the segmentation accuracy of a wide range of targets was improved. Observing the extraction results of the TANet model, it is found that there are still some misjudgments of similar features. We conducted several experiments to solve this problem according to the ideas proposed in Section 3.3.

(1) TANet with multiple losses

We use binary cross-entropy loss for each category instead of softmax cross-entropy loss.

(2) Multi-model TANet

For each category, we trained an individual network. For each input image, we fused the output probability maps of each category to obtain the final output result.

(3) Multi-task TANet

We convert a multi-category segmentation task into multiple binary-classification segmentation tasks. All tasks share one encoder, and each task owns an individual decoder.

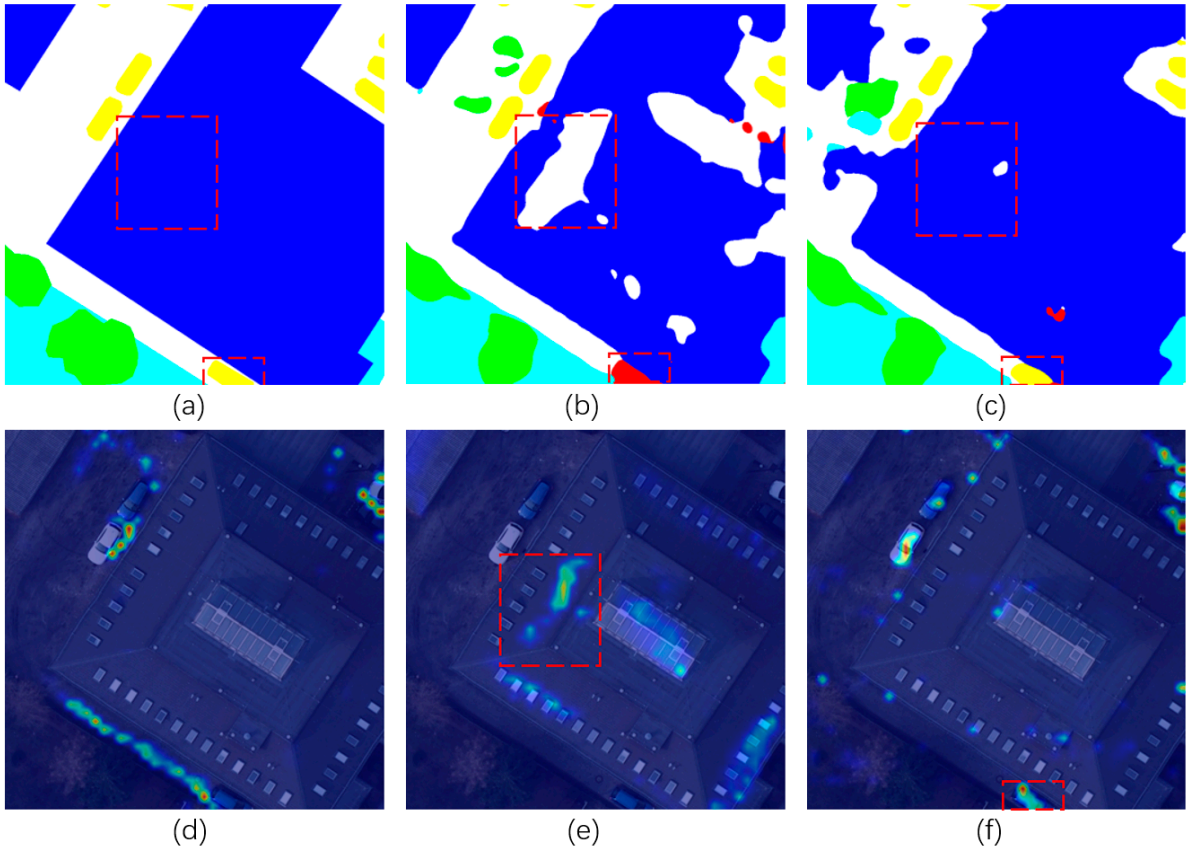

Figure 10. Visualization of LAM attention map. (**a**) label, (**b**) PAM + LAM result, (**c**) TANet result, (**d**) imperious surface attention map, (**e**) building attention map, (**f**) car attention map.

4.3.1. Experiment Results on Potsdam Datasets

As shown in Table 4, Multi-task TANet improves the performance remarkably. Compared with the baseline UNet, they employ a multi-task decoder yielding 88.35% in OA, which brings a 2.87% improvement. The results show that the accuracy can only be improved slightly if the loss function is replaced without increasing the number of decoders. Since all categories share one decoder, and the characteristics of different categories are quite different, it is difficult for one decoder to summarize the global characteristics of all categories. Another finding is that the fusion results of multiple binary segmentation models are poor. Since the models are trained separately, merging the results of multiple models is not an easy task.

Table 4. F1 score, Mean F1, and OA index results of different models on Potsdam dataset.

Method	Per-Class F1 Score (%)					Mean F1 (%)	OA (%)
	Imp. Surf.	Building	Low Veg.	Tree	Car		
UNet	87.91	91.31	81.76	82.72	88.91	86.52	85.48
TANet	89.71	93.79	82.61	83.44	88.78	87.67	87.23
TANet with multiple loss	89.52	94.00	82.91	83.82	88.49	87.72	87.35
Multi-model TANet	86.09	89.16	81.33	81.59	86.81	82.33	83.75
Multi-task TANet	**90.54**	**94.60**	**83.90**	**84.87**	**89.98**	**88.54**	**88.35**

4.3.2. Experiment Results on Vaihingen Dataset

Table 5 reports the quantitative results of the Vaihingen datasets. Compared with the baseline UNet, the methods that combine multi-task ideas achieved higher accuracy. Similar to the performance of the Potsdam dataset, changing the loss function does not

significantly improve the model. However, the multi-model fusion performed better on the Vaihingen dataset, especially the tree and car categories, which achieved the highest F1 scores. The performance of multi-model TANet is not stable in different datasets. In terms of large-area features, Multi-task TANet achieved better results. Similar categories are easier to distinguish under the action of multiple decoders.

Table 5. F1 score, Mean F1, and OA index results of different models (Vaihingen dataset).

Method	Per-Class F1 Score (%)					Mean F1 (%)	OA (%)
	Imp. Surf.	Building	Low Veg.	Tree	Car		
UNet	84.45	87.32	69.77	83.16	63.12	77.56	81.27
TANet	87.88	91.09	78.40	84.61	72.27	82.85	85.26
TANet with multiple loss	88.95	92.11	79.49	84.89	71.02	83.29	85.35
Multi-model TANet	88.37	90.77	79.43	**85.65**	**74.83**	83.81	85.67
Multi-task TANet	**89.55**	**92.41**	**80.44**	85.60	74.00	**84.40**	**86.89**

4.3.3. Experiment Results on CZ-WZ Dataset

As shown in Table 6, multi-task TANet significantly improves the performance of image semantic segmentation. Compared to UNet, TANet with multiple losses, Multi-model TANet, and Multi-task TANet improved 5.27%, 5.00%, and 8.04% in F1 score, and 2.14%, 1.88%, and 3.39% in OA score, respectively. Compared to TANet, the extraction accuracy of Multi-loss TANet and Multi-model TANet decreased, and Multi-task TANet achieved the highest accuracy. Different multitasking strategies have different effects on the model. TANet with multiple loss only uses the multitasking loss function without changing the model structure. The model still has parameter competition, and it is still difficult to avoid the problem of misclassification of similar categories. The multi-model strategy used by Multi-model TANet has higher extraction accuracy in a single model. However, it is more difficult to fuse the multi-category results. The method of directly taking the most significant term of probability value does not completely merge the multi-category results, which is caused by the differences in the predicted probability values of different models, so the multitask approach of the multi-model strategy has some instability.

Table 6. F1 score, Mean F1, and OA index results of different models (CZ-WZ dataset).

Method	Per-Class F1 Score (%)				Mean F1 (%)	OA (%)
	Road	Building	Veg.	Water		
UNet	60.75	74.13	90.26	77.32	75.62	81.41
TANet	74.16	77.31	91.20	83.91	81.65	83.96
TANet with multiple loss	73.06	76.64	91.06	82.78	80.89	83.55
Multi-model TANet	72.86	76.44	90.90	82.28	80.62	83.29
Multi-task TANet	**77.02**	**79.44**	**91.54**	**86.67**	**83.66**	**84.80**

Figures 11 and 12 compare the segmented results on different approaches to multitasking ideas. The multi-task mechanism focuses on solving the problem of misidentification of similar features. Multi-task TANet with multiple decoders achieved the best results. Since each category has a dedicated decoder, each decoder is more focused, thereby improving the ability of category discrimination.

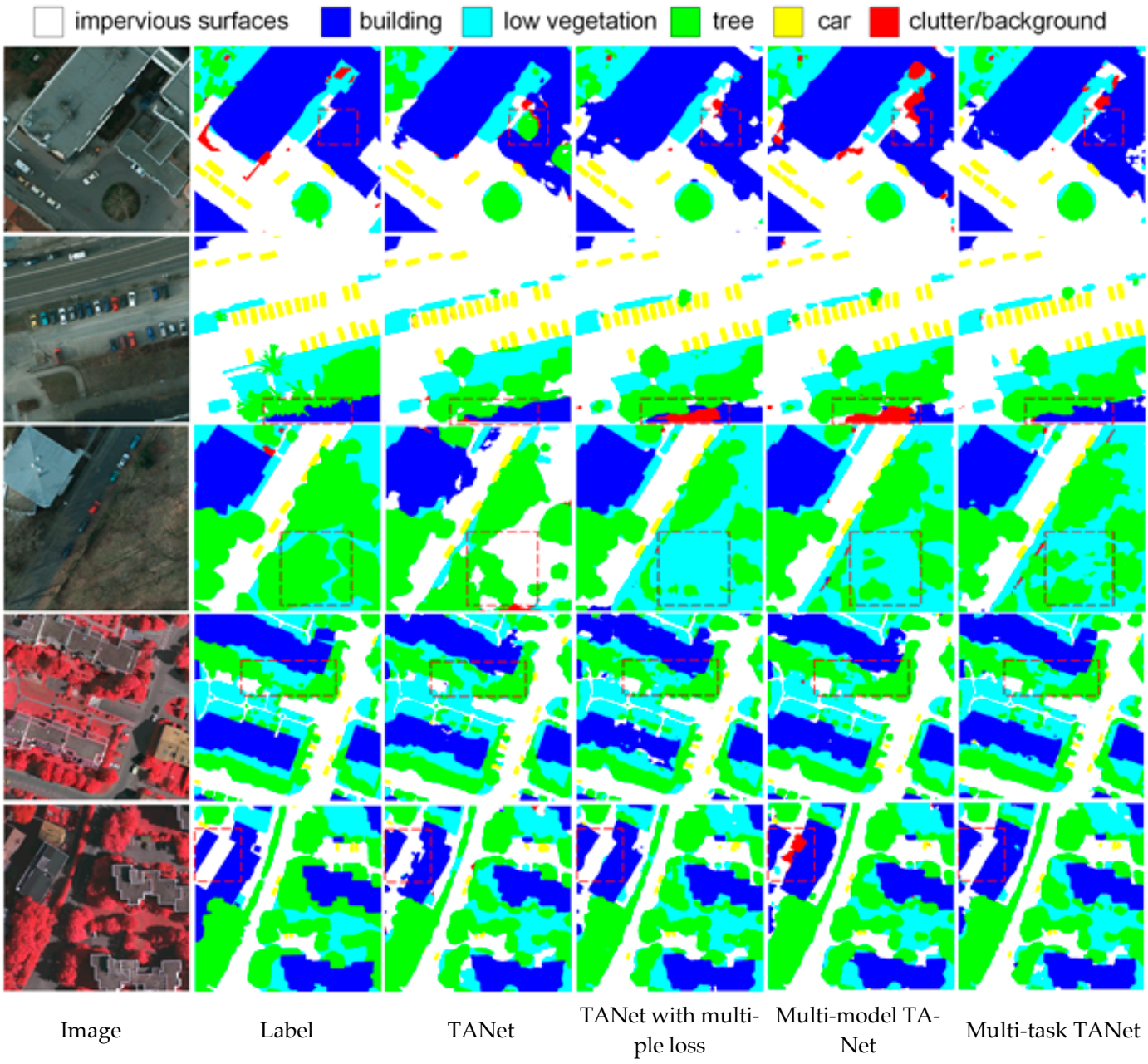

Figure 11. Examples of semantic segmentation results on ablation study of multi-task learning. The first three rows are from the Potsdam dataset and the others are from the Vaihingen dataset.

4.4. Edge Optimization Results and Discussion

The optimization includes the edge map branch (EB), the distance map branch (DB), and EAM. Firstly, to prove the validity of the DT Loss, we introduce the edge map branch and compare the results using DT Loss and cross-entropy (CE). In Table 7, EB_DT denotes the edge map branch with DT Loss, and EB_CE denotes the edge map branch with CE Loss. According to Table 7, the results of different modules of edge optimization on three datasets can be obtained, and the edge optimization of EB_DT and DB is the best result in MQANet.

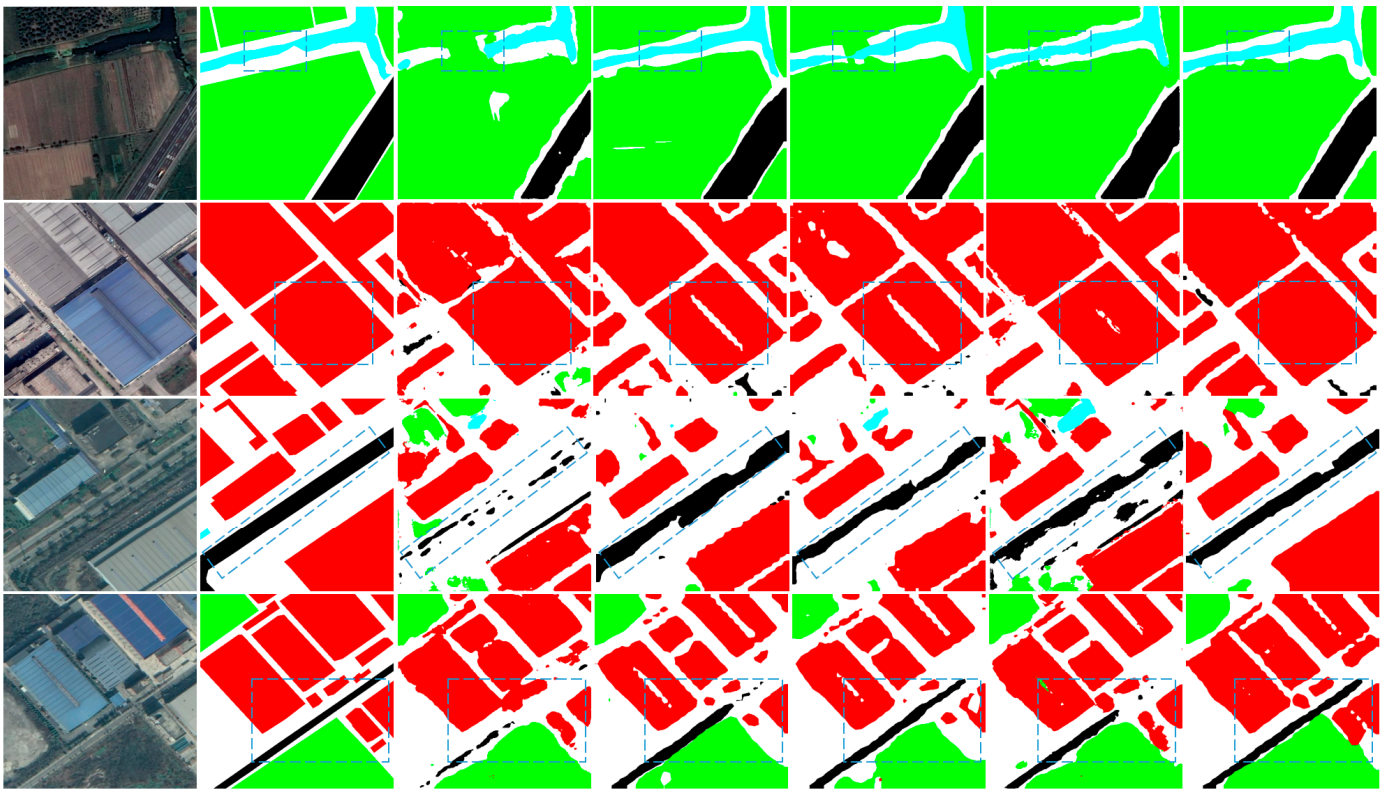

Figure 12. Examples of semantic segmentation results on ablation study of multi-task learning (CZ-WZ dataset).

Table 7. Mean F1 and OA index results on edge optimization.

Method	EB_DT	EB_CE	DB	Potsdam Dataset		Vaihingen Dataset		CZ-WZ Dataset	
				Mean F1 (%)	OA (%)	Mean F1 (%)	OA (%)	Mean F1 (%)	OA (%)
MTANet	×	×	×	88.54	88.35	84.40	86.89	83.66	84.80
MTANet	✓	×	×	89.16	88.77	84.36	86.93	83.94	85.10
MTANet	×	✓	×	89.02	88.50	84.14	86.32	82.48	84.28
MQANet	✓	×	×	89.29	88.96	84.48	87.32	84.10	85.15
MQANet	✓	×	✓	**89.35**	**89.05**	**84.61**	**87.60**	**84.27**	**85.29**

Then we evaluate the effects of the proposed distance branch and EAM. In Table 7, DB denotes the distance branch. MQANet means the previous MTANet combined with edge attention to form a Multi-task Quadruple Attention Network. Table 7 shows that all three modules have improved the segmentation results. In Table 7, MTANet is Multi-mask TANet, and MQANet is Multi-mask QANet.

The two datasets from Potsdam and Vaihingen were applied to tests for Unet, TANet, and Multi-task QANet. The results are shown in Figures 13 and 14. Compared with the three test datasets, the segmentation results of Multi-task QANet present a better effect than Multi-task TANet. The modules, the first line of Figure 13, can clearly show the performance, especially for the edge contour of the target.

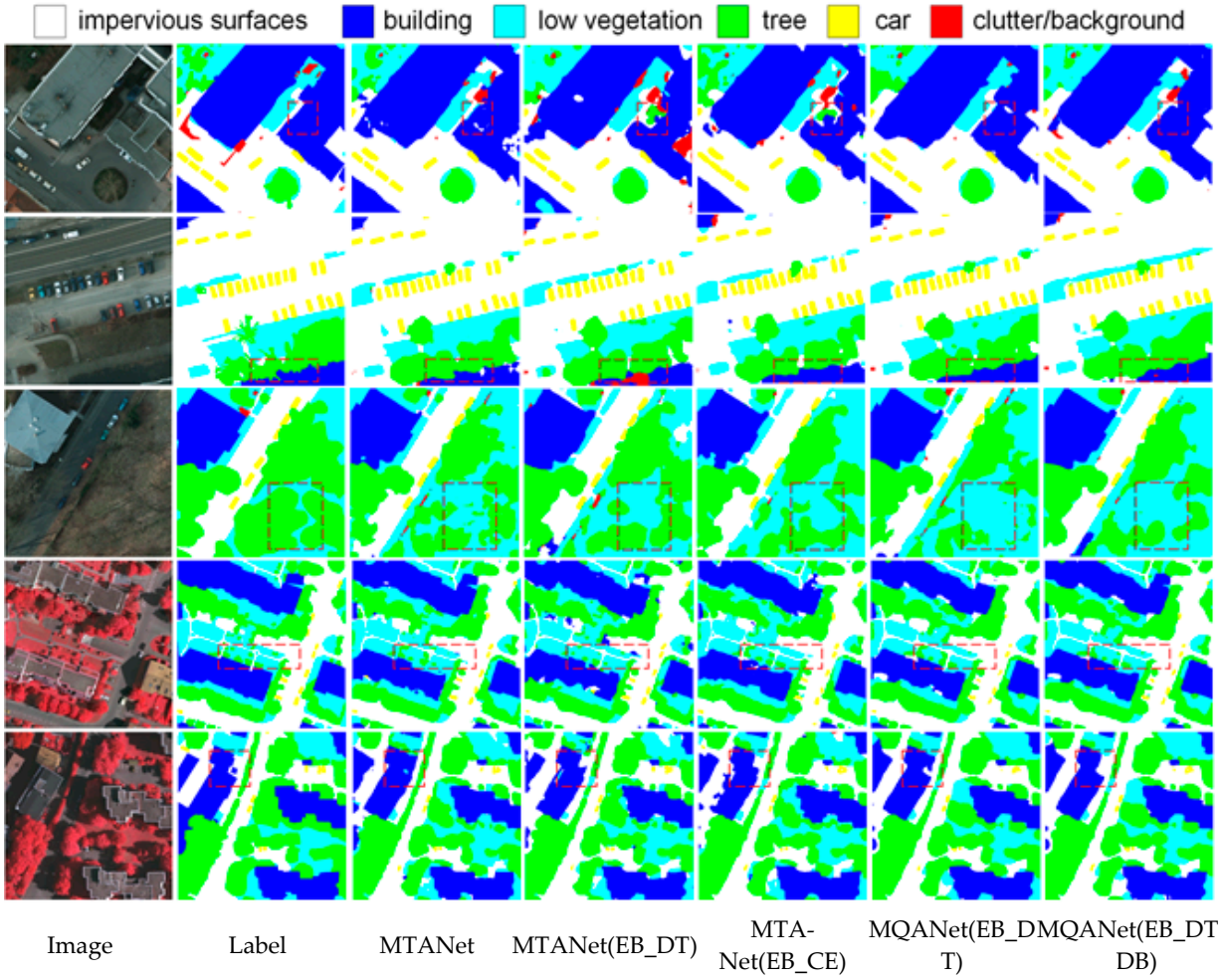

Figure 13. Results on ablation study of edge optimization. The first three rows are from the Potsdam dataset and the others are from the Vaihingen dataset.

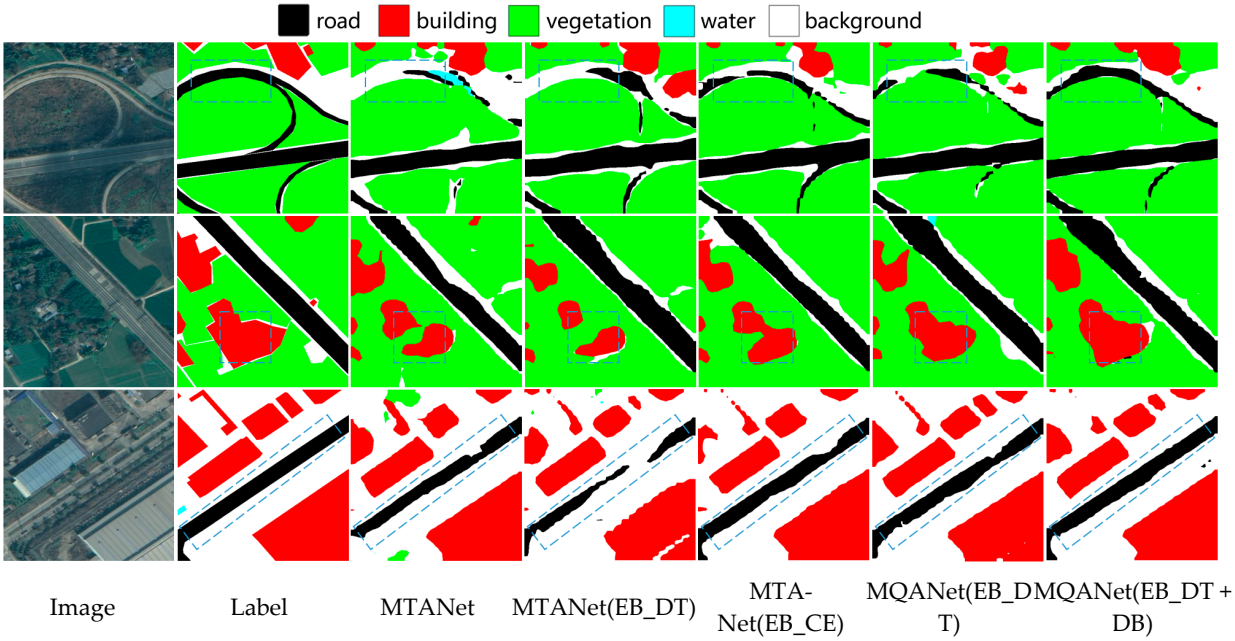

Figure 14. Examples of semantic segmentation results on ablation study of edge optimization (CZ-WZ dataset).

Table 7 shows that after optimization of each module of edge extraction, the overall accuracy of the model is improved to a certain extent, and compared with MTANet (EB_CE), MQANet (EB_DT + DB) shows an improvement of 0.61% and 0.49% in F1 score and OA, respectively. The slight improvement since the edge part accounts for a smaller proportion of the total image area. The optimized edge extraction accuracy contributes less to the overall accuracy improvement. From the results of MTANet (EB_DT) and MQANet (EB_DT), it can be seen that different edge extraction loss functions significantly impact the overall accuracy, and using the cross-entropy loss function to extract edges even reduces the original model accuracy. Moreover, the distance transform loss function can be better applied to the edge extraction task. The results of MQANet (EB_DT + DB) show that the newly introduced distance transform branch and edge attention module can both help to improve the accuracy.

4.5. Discussion of Overall Experimental Results

Table 8 is presented to analyze the results of our methods based on the two public datasets. The Mean F1 and OA of the surface objects have been promoted for the two datasets. For the Potsdam dataset, the 2.83% Mean F1 and 3.57% OA of MQANet are higher than those of Unet. For the Vaihingen dataset, the 7.05% Mean F1 and 6.33% OA of MQANet are higher than Unet.

Table 8. Quantitative results of our methods.

Method	Potsdam Dataset		Vaihingen Dataset		Execution Time/Per Image (ms)
	Mean F1 (%)	OA (%)	Mean F1 (%)	OA (%)	
UNet (baseline)	86.52	85.48	77.56	81.27	125.4
TANet (ours)	87.67	87.23	82.85	85.26	49.8
MTANet (ours)	88.54	88.35	84.40	86.89	55.0
MQANet (ours)	89.35	89.05	84.61	87.60	58.8

In addition, we tested the execution time of our proposed networks and Unet based on the method provided by Dong et al. [43], and the test results is the right column in Table 8. The test method sets the batch size to 1 and lets the network predict 200 images, the final execution time is the average of the total running time. That is the execution time of the network for a single image. From Table 8, it can be seen that the execution time of each image is reduced 66.6 ms compared to UNet, and the execution time of UNet is more than doubled compared to MQANet. Because our proposed multi-tasking mechanism splits a multi-category segmentation task into several binary-classification segmentation tasks, it can effectively reduce execution time and achieve better performance than the UNet baseline.

The experiment's results demonstrate that MQANet obtains optimal accuracy on two public datasets of ISPRS. Our module introduces two attention mechanisms: self-attention and label-attention. Thus, we achieve an enhanced overall accuracy compared to the standard self-attention model. In addition, this paper uses a multi-decoder model to reduce the parameter competition among different classes and performs additional optimization for the edge regions, thus achieving optimal accuracy.

To further describe the model accuracy distribution of different datasets, we count the sample number of each model in different accuracy intervals. We use overall accuracy (OA) as the accuracy evaluation index to draw histograms, as shown in Figure 15.

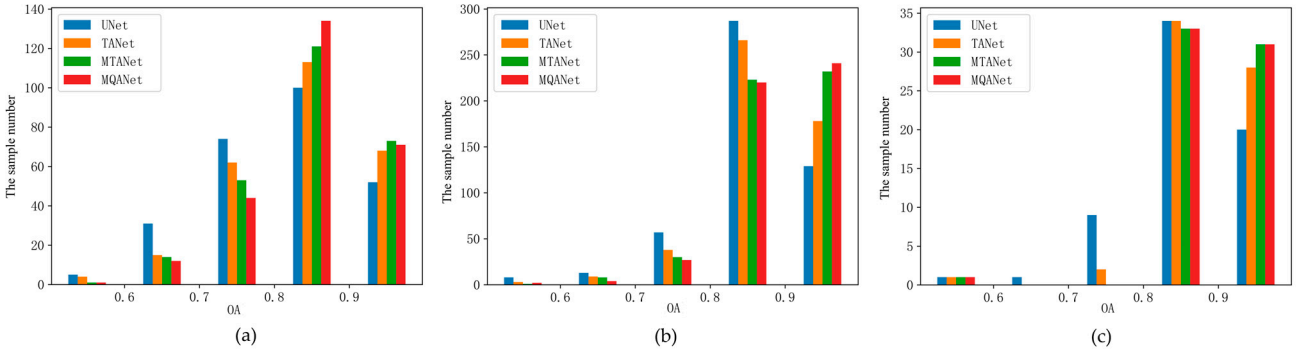

Figure 15. OA distribution of different models results in three datasets. (**a**) OA distribution map on CZ-WZ data set, (**b**) OA distribution map on Potsdam data set, (**c**) OA distribution map on Vaihingen data set.

As shown in Figure 15, in the Chongzhou–Wuzhen dataset (CZ-WZ dataset), when OA is below 0.8, the number of samples in UNet is significantly higher than that of other models. TANet, MTANet, and MQANet show a decreasing trend. In contrast, when the accuracy is above 0.8, MQANet has the largest number of samples. The above results show that MQANet achieved optimal accuracy in most samples of the CZ-WZ dataset. In the Potsdam and Vaihingen datasets, the regularities are similar to those in the CZ-WZ dataset. When the accuracy is above 0.9, the number of samples of UNet is significantly lower than that of other models, and MQANet has the largest number of samples. This indicates that MQANet also achieves the optimal accuracy in most samples of Potsdam and Vaihingen datasets.

To obtain a more detailed sample distribution, we plotted the extraction accuracy of each sample as a broken line diagram in the ISPRS-Vaihingen test dataset. As shown in Figure 16, the extraction results of MQANet are the highest in four models. TANet, MTANet, and MQANet show an upward trend in the Vaihingen test sample, which further indicates that the proposed model has a high OA index in the Vaihingen test dataset.

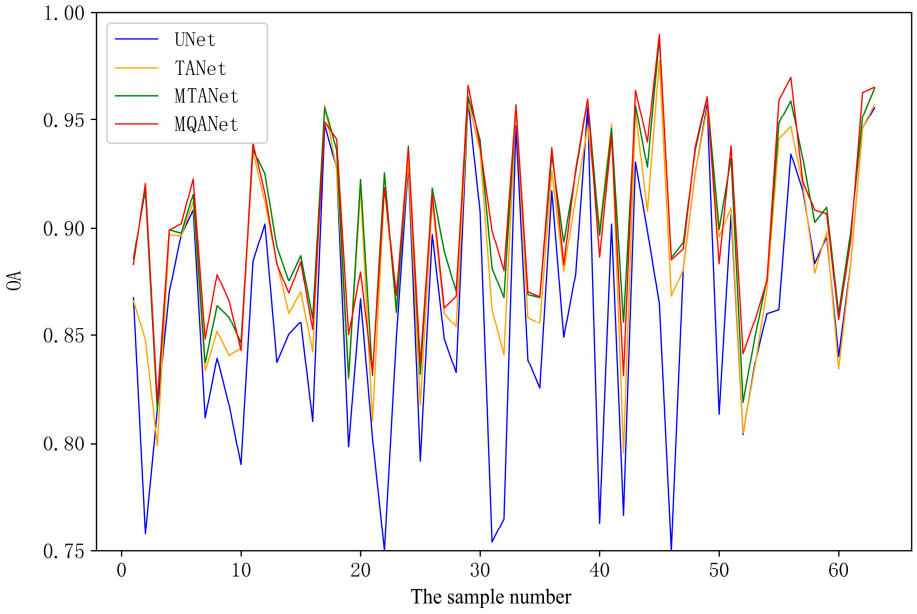

Figure 16. OA Statistical distribution map of model results in Vaihingen dataset.

In addition, we compared the current work on two publicly available datasets to verify the effectiveness of our optimal model, MQANet, still using two evaluation metrics, the average F1 and OA. Among the existing works compared with this paper, the work CBAMNet [21] containing the self-attentive mechanism, SENet [44], and the latest network

Deeplabv3+ [11] with the spatial feature pyramid structure are included, in addition to the latest CRMS [45] network with optimal feature extraction using the multiscale residual module. The experimental results are shown in Table 9. As can be seen from Table 9, the results of our proposed model MQANet show better results on both datasets. On the Vaihingen dataset, our MQANet network has higher Mean F1 and OA than other networks, and on the Potsdam dataset, our MQANet network has higher Mean F1 than other networks, and only OA is 0.08% lower than DSPCANet [15]. Mean F1 score for classification is calculated as the harmonic mean of precision and recall [22], and OA is the ratio of the number of correct pixels to the total number of pixels. This shows that our MQANet has certain advantages in the equalization of various objects identification.

Table 9. Quantitative results of the current work.

Method	Potsdam Dataset		Vaihingen Dataset	
	Mean F1 (%)	**OA (%)**	**Mean F1 (%)**	**OA (%)**
CBAMNet [21]	86.04	85.14	83.77	86.47
Deeplabv3+ [11]	88.01	87.06	83.77	85.71
SENet [44]	87.97	87.63	82.85	85.26
CRMS [45]	89.02	88.92	83.25	86.40
EFCNet [14]	80.17	81.77	81.87	85.46
DSPCANet [15]	87.19	**90.13**	84.46	87.32
MQANet (ours)	**89.35**	89.05	**84.61**	**87.60**

5. Conclusions

A Multi-task Quadruple Attention Network (MQANet) is proposed to improve the accuracy of multi-object semantic segmentation of remote sensing images. We introduce the attention mechanism to obtain more global features and improve the accuracy of the large object area. Furthermore, two self-attention modules are introduced, which are named PAM + CAM, and the OA and Mean F1 are improved. Then, we build a label attention module (LAM) and combine all three attention modules into a triple attention network (TANet). Meanwhile, we proposed three alternative methods to improve the ability to identify similar objects: Multi-task TANet (MTANet). Experiment results show that the multi-task learning method obtains the highest accuracy. Finally, some edge optimizations are made to improve the accuracy of the edge area further, and we combine Multi-task TANet and edge optimizations as the Multi-task QANet (MQANet).

Three datasets were used to verify the accuracy of the proposed model. Compared with the baseline UNet in the Vaihingen dataset, MQANet improved the OA and Mean F1 by 6.33% and 7.05%, respectively. MTANet improved the OA and Mean F1 by 5.48% and 6.84%, respectively. Compared with the baseline UNet in the Potsdam dataset, MQANet improved the OA and Mean F1 by 3.57% and 2.83%, respectively. MTANet improved the OA and Mean F1 by 2.87% and 2.02%, respectively. Compared with the baseline UNet in the CZ-WZ dataset, MQANet improved the OA and Mean F1 by 3.88% and 8.65%, respectively. MTANet improved the OA and Mean F1 by 3.39% and 8.04%, respectively. Through extensive experiments, the proposed MQANet outperforms other methods by a large margin on Vaihingen, Potsdam and self-annotated datasets (CZ-WZ dataset). The results demonstrate that the proposed model (MQANet) has a large accuracy improvement in both F1 and OA indices, and the quadruple attention modules are helpful for large object semantic segmentation of RS images.

The proposed multi-tasking mechanism splits a multi-category segmentation task into several binary-classification segmentation tasks, each of which requires a separate decoder. The types of multi-object semantic segmentation involve 5 or 6 categories in our study, which can achieve better results. However, if the objects are subdivided into dozens or even more categories, the model needs to construct a decoder for each category. A large number of decoders may cause the size expansion of the model, and the applicability of

the model may be decreased. In the future, more types of multi-object will be tested to optimize a more robust multi-task semantic segmentation.

Author Contributions: Conceptualization, Y.L. and Y.S.; methodology, Y.L. and L.H.; software, Y.L.; validation, Y.S., Z.T. and J.Z.; formal analysis, Y.S.; investigation, Z.T.; resources, L.H.; data curation, Y.G.; writing—original draft preparation, Y.L.; writing—review and editing, L.H. and S.L.; visualization, Y.S.; supervision, Y.L. and Z.T.; project administration, L.H.; funding acquisition, L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Projects of Global Change and Response of Ministry of Science and Technology of China under Grant 2020YFA0608203, and in part by the fundamental Research Funds for the Central Universities, UESTC (ZYGX2019J064), in part by Major Science and Technology Projects of Sichuan Province under Grant No.2022ZDZX0001, in part by the Science and Technology Support Project of Sichuan Province under Grant 2021YFS0335, in part by China Meteorological Administration Project under Grant FY-APP-2021.0304 and CXFZ2022J031.

Data Availability Statement: The authors would like to thank the team of Potsdam 2-D semantic labeling data and the Vaihingen dataset for the data and experiments.

Acknowledgments: The authors appreciate the reviewers for their constructive comments and kind help to improve the manuscript.

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

References

1. Ye, Q.; Zhao, H.; Li, Z.; Yang, X.; Gao, S.; Yin, T.; Ye, N. L1-Norm distance minimization-based fast robust twin support vector $ k $-plane clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 4494–4503. [CrossRef]
2. Sun, L.; Ma, C.; Chen, Y.; Shim, H.J.; Wu, Z.; Jeon, B. Adjacent superpixel-based multiscale spatial-spectral kernel for hyperspectral classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1905–1919. [CrossRef]
3. Duan, Y.; Huang, H.; Wang, T. Semisupervised feature extraction of hyperspectral image using nonlinear geodesic sparse hypergraphs. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [CrossRef]
4. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
5. Gualtieri, J.A.; Cromp, R.F. Support vector machines for hyperspectral remote sensing classification. In *27th AIPR Workshop: Advances in Computer-Assisted Recognition*; SPIE: Bellingham, WA, USA, 1999; pp. 221–232.
6. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
7. 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]
8. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
9. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef]
10. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Rethinking atrous convolution for semantic image segmentation liang-chieh. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *5*.
11. 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.
12. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
13. Zheng, Z.; Zhong, Y.; Wang, J.; Ma, A. Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 13–19 June 2020; pp. 4096–4105.
14. Chen, L.; Dou, X.; Peng, J.; Li, W.; Sun, B.; Li, H. EFCNet: Ensemble Full Convolutional Network for Semantic Segmentation of High-Resolution Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
15. Li, Y.C.; Li, H.C.; Hu, W.S.; Yu, H.L. DSPCANet: Dual-Channel Scale-Aware Segmentation Network With Position and Channel Attentions for High-Resolution Aerial Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 8552–8565. [CrossRef]
16. Sharifi, A.; Mahdipour, H.; Moradi, E.; Tariq, A. Agricultural field extraction with deep learning algorithm and satellite imagery. *J. Indian Soc. Remote Sens.* **2022**, *50*, 417–423. [CrossRef]

17. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
18. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3146–3154.
19. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. Aˆ 2-nets: Double attention networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
20. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; Liu, H. Expectation-maximization attention networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9167–9176.
21. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
22. Ding, L.; Tang, H.; Bruzzone, L. LANet: Local attention embedding to improve the semantic segmentation of remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 426–435. [CrossRef]
23. Niu, R.; Sun, X.; Tian, Y.; Diao, W.; Chen, K.; Fu, K. Hybrid multiple attention network for semantic segmentation in aerial images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–18. [CrossRef]
24. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–Spatial Feature Tokenization Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14.
25. 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, 18–22 June 2018; pp. 182–186.
26. Geng, Q.; Zhang, H.; Qi, X.; Huang, G.; Yang, R.; Zhou, Z. Gated path selection network for semantic segmentation. *IEEE Trans. Image Process.* **2021**, *30*, 2436–2449. [CrossRef]
27. Wu, T.; Tang, S.; Zhang, R.; Cao, J.; Zhang, Y. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Trans. Image Process.* **2020**, *30*, 1169–1179. [CrossRef]
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
29. Park, J.; Woo, S.; Lee, J.; Kweon, I. BAM: Bottleneck Attention Module. In Proceedings of the British Machine Vision Conference, Newcastle, UK, 3–6 September 2018.
30. Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C.C.; Lin, D.; Jia, J. Psanet: Point-wise spatial attention network for scene parsing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 267–283.
31. Caruana, R. Multitask learning. *Mach. Learn.* **1997**, *28*, 41–75. [CrossRef]
32. Baxter, J. A model of inductive bias learning. *J. Artif. Intell. Res.* **2000**, *12*, 149–198. [CrossRef]
33. Ben-David, S.; Schuller, R. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 567–580.
34. Maurer, A.; Pontil, M.; Romera-Paredes, B. Sparse coding for multitask and transfer learning. In Proceedings of the International Conference on Machine Learning PMLR, Atlanta, GA, USA, 16–21 June 2013; pp. 343–351.
35. Ando, R.K.; Zhang, T.; Bartlett, P. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.* **2005**, 6.
36. Nakamura, A.T.M.; Grassi, V., Jr.; Wolf, D.F. An effective combination of loss gradients for multi-task learning applied on instance segmentation and depth estimation. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104205. [CrossRef]
37. Duong, L.; Cohn, T.; Bird, S.; Cook, P. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, 26–31 July 2015; pp. 845–850.
38. Yang, Y.; Hospedales, T. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
39. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7482–7491.
40. Borse, S.; Wang, Y.; Zhang, Y.; Porikli, F. Inverseform: A loss function for structured boundary-aware segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 5901–5911.
41. ISPRS. 2D Semantic Labeling Contest—Potsdam. Available online: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx (accessed on 4 September 2018).
42. ISPRS. 2D Semantic Labeling Contest—Vaihingen. Available online: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx (accessed on 4 September 2018).
43. Chu, X.; Chen, L.; Yu, W. NAFSSR: Stereo Image Super-Resolution Using NAFNet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, New Orleand, LA, USA, 19–24 June 2022; pp. 1239–1248.

44. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
45. Liu, Z. Semantic Segmentation of Remote sensing images via combining residuals and multi-scale modules. In Proceedings of the ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application, Shenyang, China, 17–19 December 2021; pp. 1–4.