remote sensing

MDPI

Article

MKANet: An Efficient Network with Sobel Boundary Loss for Land-Cover Classification of Satellite Remote Sensing Imagery

Zhiqi Zhang [1,2,†], Wen Lu [1,†], Jinshan Cao [1,*] and Guangqi Xie [1]

[1] School of Computer Science, Hubei University of Technology, Wuhan 430068, China
[2] State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China
* Correspondence: caojs@hbut.edu.cn
† These authors contributed equally to this work.

Abstract: Land cover classification is a multiclass segmentation task to classify each pixel into a certain natural or human-made category of the earth's surface, such as water, soil, natural vegetation, crops, and human infrastructure. Limited by hardware computational resources and memory capacity, most existing studies preprocessed original remote sensing images by downsampling or cropping them into small patches less than 512 × 512 pixels before sending them to a deep neural network. However, downsampling incurs a spatial detail loss, renders small segments hard to discriminate, and reverses the spatial resolution progress obtained by decades of efforts. Cropping images into small patches causes a loss of long-range context information, and restoring the predicted results to their original size brings extra latency. In response to the above weaknesses, we present an efficient lightweight semantic segmentation network termed MKANet. Aimed at the characteristics of top view high-resolution remote sensing imagery, MKANet utilizes sharing kernels to simultaneously and equally handle ground segments of inconsistent scales, and also employs a parallel and shallow architecture to boost inference speed and friendly support image patches more than 10× larger. To enhance boundary and small segment discrimination, we also propose a method that captures category impurity areas, exploits boundary information, and exerts an extra penalty on boundaries and small segment misjudgments. Both visual interpretations and quantitative metrics of extensive experiments demonstrate that MKANet obtains a state-of-the-art accuracy on two land-cover classification datasets and infers 2× faster than other competitive lightweight networks. All these merits highlight the potential of MKANet in practical applications.

Keywords: semantic segmentation; convolutional neural network; land-cover classification

Citation: Zhang, Z.; Lu, W.; Cao, J.; Xie, G. MKANet: An Efficient Network with Sobel Boundary Loss for Land-Cover Classification of Satellite Remote Sensing Imagery. *Remote Sens.* **2022**, *14*, 4514. https://doi.org/10.3390/rs14184514

Academic Editors: Sidike Paheding and Ashraf Saleem

Received: 6 August 2022
Accepted: 7 September 2022
Published: 9 September 2022

1. Introduction

Nowadays, various satellite constellations with shorter revisit periods and wider observation coverage have formed the global earth observation system which can quickly obtain huge amounts of high-spatial-resolution, high-temporal-resolution, and high-spectral-resolution remote sensing imagery [1]. For example, China has 30 to 50 high-resolution remote sensing satellites in orbit, and by a conservative estimation, several hundred TB of data are acquired every day [2]. However, with regard to the acquisition speed, the rapid intelligent processing of remote sensing data still lags [3,4]. In the new era of artificial intelligence, how to realize instant perception and cognition of remote sensing imagery has become an urgent problem to be solved.

Land cover classification is a multiclass segmentation task to classify each pixel into a certain natural or human-made category of the earth's surface, such as water, soil, natural vegetation, crops, and human infrastructure. The land cover and its change influence the ecosystem, human health, social development, and economic growth. The last several decades of years have witnessed the improvement of the spatial resolution of remote

sensing imagery from 30 m to submeter. With richer details and structural information of objects emerging in remote sensing imagery, land cover classification methods have shifted from discriminating the spectral or spectral–spatial information of local pixels to extracting contextual information and spatial relationship of ground objects [5]. Among them, deep neural networks (DNN) have been widely used for their strong feature extraction and high-level semantic modeling ability. However, a large computational resource consumption brings slow inference speeds and restricts the practical application of DNN in remote sensing imagery. Meanwhile, the incapability of processing large-size image patches causes the cropping size to be too small, and the resulting loss of long-range context information is detrimental to prediction accuracy.

To obtain a high accuracy, conventional semantic segmentation networks, such as UNet [6], FC-DenseNet [7], and DeepLabv3+ [8], usually adopt a wide and deep backbone as an encoder at the cost of a large computational complexity and memory occupation. In the task of land-cover classification, limited by GPU memory capacity, most existing studies preprocess the original remote sensing images by downsampling or cropping them into small patches less than 512 × 512 pixels before sending them to a deep neural network. For example, CFAMNet [9] proposed a class feature attention mechanism fused with an improved Deeplabv3+ network. To avoid memory overflow, 150 remote sensing images of 7200 × 6800 pixels were cropped into 20,776 images of 128 × 128 pixels. DEANet [10] used a dual-branch encoder structure that depended on VGGNet [11] or ResNet [12]; in the experiments, each image with a resolution of 2448 × 2448 pixels was compressed to half the size and then divided into subimages with a resolution of 512 × 512 pixels. DISNet [13] integrated the dual attention mechanism module, including the spatial attention mechanism and channel attention mechanism, into the Deeplabv3+ network. In the experiments, the original images were also cropped into small patches of 512 × 512 pixels before being sent into the network.

However, it takes decades of efforts to improve the spatial resolution of remote sensing imagery, and downsampling reverses this progress and incurs a spatial detail loss. The rich details of objects, such as the geometrical shape and structural content of objects, are blurred by downsampling. It renders small segments hard to discriminate, thus offsetting the gain enabled by the large backbone.

On the other hand, cropping the original images into small patches less than or equal to 512 × 512 pixels causes a loss of long-range context information and leads to misjudgments. As shown in Figure 1, a remote sensing image with a resolution of 2048 × 1536 pixels is cropped into 12 small patches with a resolution of 512 × 512 pixels. If one views the whole original image, it is clear that the water surface is a lake; however, if one views the individual small patches, the water surface may be misjudged as a river. Therefore, compared with images in other domains, such as street view images in the autonomous driving field, the support of large-size image input is more important for the correct semantic segmentation of remote sensing images (in Section 4.4.1, we investigate the influence of the input image size on prediction accuracy, and demonstrate that the loss of long-range context information caused by a small cropping size would create a misjudgment and yield a lower accuracy). Another drawback of cropping is that cropping the images and restoring the predicted results to their original size incur extra latency. Hence, aimed at the characteristics of top view high-resolution remote sensing imagery, it is necessary to redesign the architecture of semantic segmentation networks to support large-size image patches.

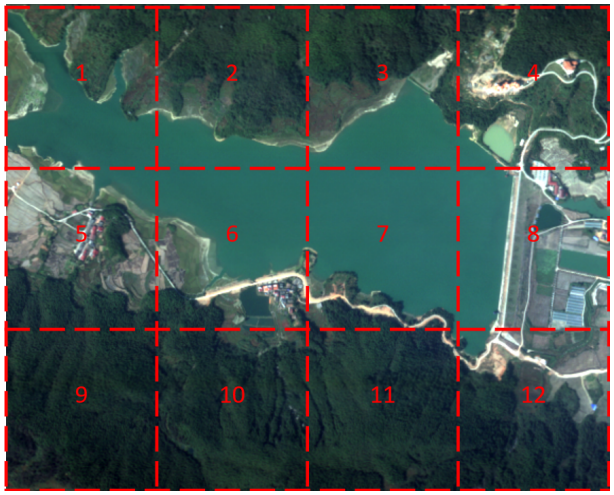

Figure 1. A remote sensing image with a resolution of 2048 × 1536 pixels is cropped into 12 small patches with a resolution of 512 × 512 pixels.

As presented in Figure 1, the abundant small segments, rich boundaries, and small interclass variance in remote sensing images are all likely to cause semantic ambiguity near the boundaries and small segments. Meanwhile, the areas where multiple land-cover categories exist contain richer information and are more prone to be misjudged. In the other aspect, the number of interior pixels grows quadratically with segment size and can far exceed the number of boundary pixels, which only grows linearly. However, the ground truth masks and conventional loss functions value all pixels equally and are less sensitive to boundary quality. Hence, it is necessary to capture category impurity areas and implement an effective measure to reinject boundary information into the semantic segmentation network.

In summary, the slow inference speed, incapability of processing large-size image patches, and easy misjudgment of boundaries and small segments are three factors that restrict the practical applications of semantic segmentation networks. To alleviate these three problems, we present an efficient lightweight semantic segmentation network termed Multibranch Kernel-sharing Atrous convolution network (MKANet) and propose the Sobel Boundary Loss for efficient and accurate land-cover classification of remote sensing imagery. MKANet acquires state-of-the-art accuracy on two land-cover classification datasets and infers 2× faster than other competitive lightweight networks (Figure 2). The contributions of this paper can be summarized in three aspects:

1. Aimed at the characteristics of top view remote sensing imagery, we handcraft the Multibranch Kernel-sharing Atrous (MKA) convolution module for multiscale feature extraction;
2. For large input image size support and a fast inference speed, we design a shallow semantic segmentation network (MKANet) based on MKA modules;
3. For an accurate prediction of boundaries and small segments, we propose a novel boundary loss named Sobel Boundary Loss.

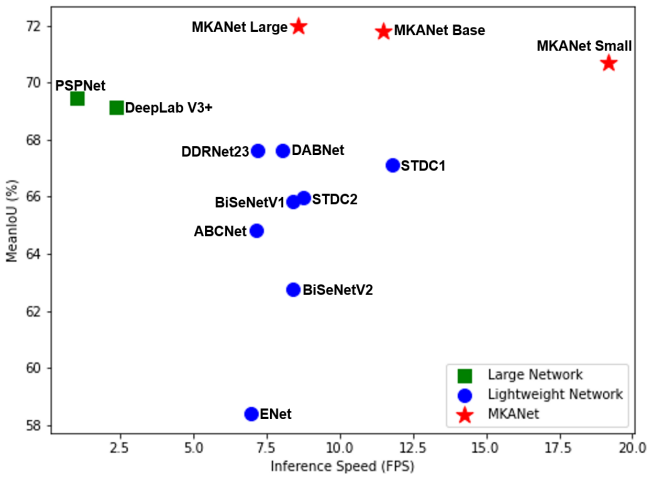

Figure 2. Speed–Accuracy performance comparison on the DeepGlobe Land Cover dataset of image size 2448 × 2448 pixels, where the proposed MKANets achieve state-of-the-art speed–accuracy trade-off.

2. Related Work

In this section, we first present some representative lightweight semantic segmentation networks and our improvement direction; then, we introduce the original kernel-sharing mechanism as well as our reasons to adopt it and its limitation.

2.1. Lightweight Semantic Segmentation Networks

Certain semantic segmentation networks employ lightweight backbones, so compared with large networks, they consume fewer hardware resources. For example, ENet [14] employs an early downsampling strategy and asymmetric architecture that consists of a large encoder and a small decoder in pursuit of real-time processing. BiSeNetV1 [15] and BiSeNetV2 [16] use a two-pathway architecture: the first pathway captures the spatial details with wide channels and shallow layers, and the second pathway extracts the categorical semantics with narrow channels and deep layers and then fuses the output features of these two paths to make the final prediction. Hong et al. proposed a deep dual-resolution network (DDRNet) [17] that consisted of two deep branches between which multiple bilateral fusions were performed. To guarantee accuracy without drastically increasing computational consumption, ABCNet [18] replaced the dot-product attention mechanism of quadratic complexity with a linear attention mechanism for global contextual information extraction. DABNet [19] adopted a depthwise asymmetric convolution and dilated convolution to build a bottleneck structure for parameter reduction. DFNet [20] utilized the partial order pruning algorithm to obtain a lightweight backbone and efficient decoder.

Although these lightweight networks are computationally inexpensive, there is still some gap in accuracy between them and large networks. An empirical observation shows that prediction errors are more likely to occur on boundaries and small segments [21]; this observation prompted us to propose a novel boundary loss as detailed in Section 3.4 to bridge the accuracy gap between lightweight networks and large networks.

2.2. Kernel-Sharing Mechanism

The authors of KSAC [22] argued the weakness of the original ASPP [23] structure was that the kernels in the branch with small atrous rates only learned details and handled small objects well, while the kernels in the branch with large atrous rates were only able to extract features with large receptive fields. The lack of communication among branches compromised the generalizability of individual kernels. To tackle this problem, they proposed that multiple branches with different atrous rates share a single kernel, so the shared kernel is able to scan the input feature maps more than once with both small and large receptive fields. Another benefit is that the objects of various sizes can all contribute

to the training of the shared kernel, so the number of effective training samples increases, and the representation ability of the shared kernel is thus improved. KSAC adopted the architecture of DeepLabV3+, and its modified ASPP structure consisted of a 1×1 convolutional branch, a global average pooling branch followed by a 1×1 convolution, and three kernel-sharing atrous convolutional branches with rates (6, 12, 18).

The spatial resolutions of different satellites vary, which makes land objects belonging to the same category have different scales. On the other hand, lands belonging to the same category have different areas. The features at multiple scales in remote sensing images match with the advantages of the kernel-sharing mechanism, so we decided to adopt this mechanism in the basic module design.

However, the direct introduction of KSAC to the basic module is impossible, because KSAC can only be applied once as the last stage of the encoder. Firstly, its global average pooling branch is supposed to obtain the image-level features; secondly, its large atrous rates (6, 12, 18) are not suitable for extracting low-level features, especially for remote sensing images in which objects have smaller spatial scales than those in general images. Therefore, constructing a backbone by purely stacking the original KSAC structure is not feasible. Therefore, we handcrafted a novel multibranch module as detailed in Section 3.1; the newly proposed module can be stacked in multiple stages as the backbone for semantic segmentation.

3. Proposed Method

In this section, we first introduce the MKA module which constitutes the backbone of the network; then, we show the network architecture that infers $2\times$ faster than other competitive lightweight networks; at last, we present the Sobel Boundary Loss that helps boundary recovery and improves small segment discrimination.

3.1. Multibranch Kernel-Sharing Atrous Convolution Module

Conventional networks usually accumulate contextual information over large receptive fields by stacking a series of convolutional layers, so they have deep network architectures that consist of dozens of layers. Some networks even have more than one hundred layers, for example, FC-DenseNet [7]. However, one of the costs of building a deep architecture is slow inference speed. For a high efficiency and fast inference speed, we designed the Multibranch Kernel-sharing Atrous (MKA) convolution module, as illustrated in Figure 3. Its parallel structure and kernel sharing mechanism can simultaneously capture a wider range of contexts for large segments and local detailed information for small segments and boundaries. Specifically, the receptive field of a typical three-branch MKA module equals that of five 3×3 convolutional layers connected in series. Different from ASPP or KSAC which can only be applied once as the last part of a backbone, MKA modules can be stacked in series as the backbone for semantic segmentation networks. Hence, with MKA modules, it is no longer necessary to build a deep network architecture. Meanwhile, the computation cost of an MKA module is inexpensive, and along with its parallel structure, the MKA module can greatly boost inference speed.

The MKA module consists of three parts:

- Multibranch kernel-sharing depthwise atrous convolutions;
- Multibranch depthwise convolutions;
- Concatenation and pointwise convolution.

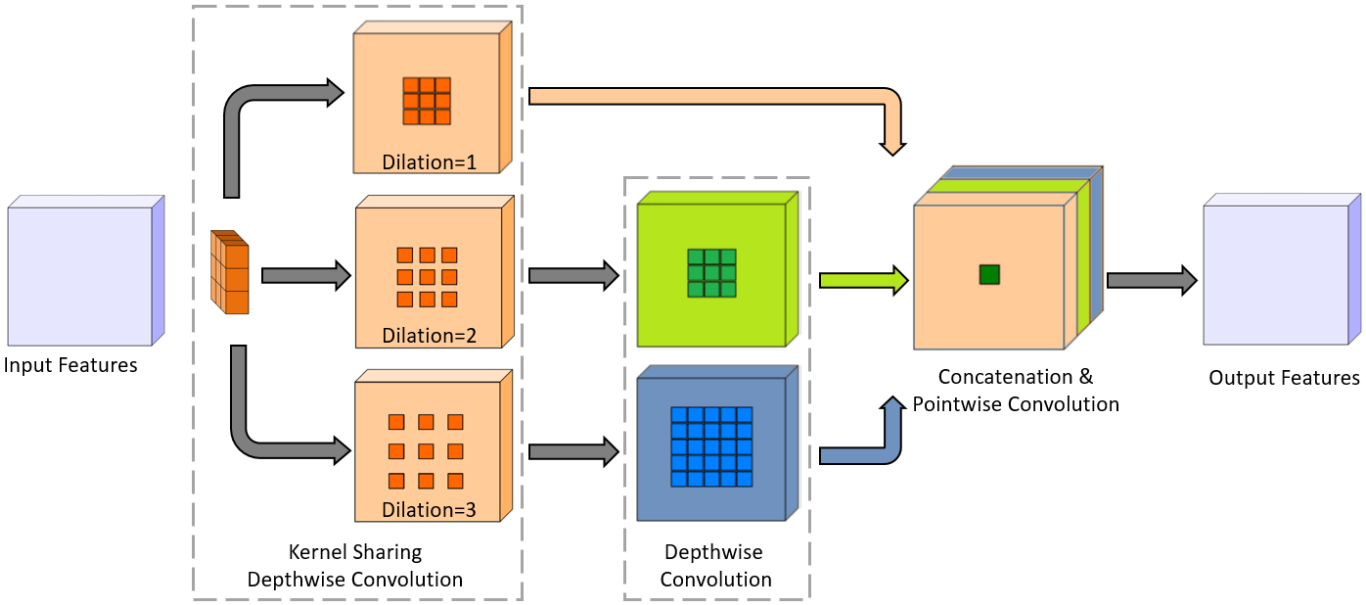

Figure 3. Structure of the MKA module. The orange square represents the kernel shared by three depthwise atrous convolutional branches with dilation rates (1, 2, 3).

3.1.1. Part 1: Multibranch Kernel-Sharing Depthwise Atrous Convolutions

Assume that the number of channels of the input and output features is N and that the number of branches is M. One 3×3 kernel is shared by the M depthwise atrous convolutions with dilation rates of 1, 2, ..., and M. Next, a batch normalization is applied in each branch.

Compared with KSAC [22], the MKA module abandons the 1×1 convolutional branch and the global average pooling branch. The dilation rates also decrease from (6, 12, 18) to (1, 2, 3). To further reduce the computation complexity and memory occupation, regular atrous convolutions are replaced by depthwise atrous convolutions, decreasing the kernel parameters, computation cost, and memory footprint to $1/N$.

This design inherits the merits of the kernel-sharing mechanism. The generalization ability of the shared kernels is enhanced by learning both the local detailed features of small segments and the global semantic features of large segments. The kernel-sharing mechanism can also be considered a feature augmentation performed inside the network, which is complementary to the data augmentation performed in the preprocessing stage, to enhance the representation ability of kernels.

3.1.2. Part 2: Multibranch Depthwise Convolutions

Since atrous convolution introduces zeros in the convolutional kernel, within a kernel of size $k_d \times k_d$, the actual pixels that participate in the computation are just $k \times k$, with a gap of $r - 1$ between them. Hence, a kernel only views the feature map in a checkerboard fashion and loses a large portion of information. Furthermore, the adjacent points of its output feature map do not have any common pixels participating in the computation, thereby causing the output feature map to be unsmooth. This gridding artifact issue is exacerbated when atrous convolutions are stacked layer by layer. To alleviate this detrimental effect, for the ith branch ($i > 1$), a depthwise regular convolution with kernel size $(2 \times i - 1)$ is added, followed by a batch normalization. Note that an atrous convolution with a dilation rate of 1 is just a regular convolution; thus, for the first branch, nothing is added. Again, depthwise convolutions are applied here to reduce computation and memory costs. With the exception of smoothing the output feature maps of the preceding part, these depthwise convolutions can further extract useful information.

3.1.3. Part 3: Concatenation and Pointwise Convolution

After the second part, the output features of each branch are concatenated, and then a 1×1 convolution is applied to the fused features. This part has two functions: generating new features through linear combinations and compressing the number of channels of the fused features from $M \times N$ to N to reduce the computational complexity of the next module.

3.1.4. Complexity Analysis

The number of parameters in the first part is $3 \times 3 \times N = 9N$; in the second part, it is $((4M^3 - M)/3 - 1) \times N$; and in the third part, it is $M \times N \times N = MN^2$. The number of branches M is suggested to be 3, which is substantially less than the number of channels N. Hence, the total parameters of the MKA module are approximately MN^2, which is even less than the parameters of one regular 3×3 convolution.

3.2. Network Architecture

For a faster inference speed and small memory occupation, based on MKA modules, we designed a lightweight semantic segmentation network named MKANet, as illustrated in Figure 4. Attributed to the large receptive field of MKA modules, the network architecture of MKANet is very shallow. It consists of two initial convolutional layers and three MKA modules as the encoder and two coordinate attention modules (CAMs) [24] as the decoder to fuse multiscale feature maps from different stages. By horizontal and vertical extent pooling kernels, CAMs can capture long-range dependencies along one spatial direction and preserve precise positional information along the other spatial direction, thus more accurately augmenting the representations of the objects of interest in the fused feature maps.

As presented in Section 4.2, this shallow but effective architecture makes MKANet capable of supporting an input image size more than 10 times larger than that supported by conventional networks. Furthermore, compared with other competitive lightweight networks, the inference speed of MKANet is twice as fast.

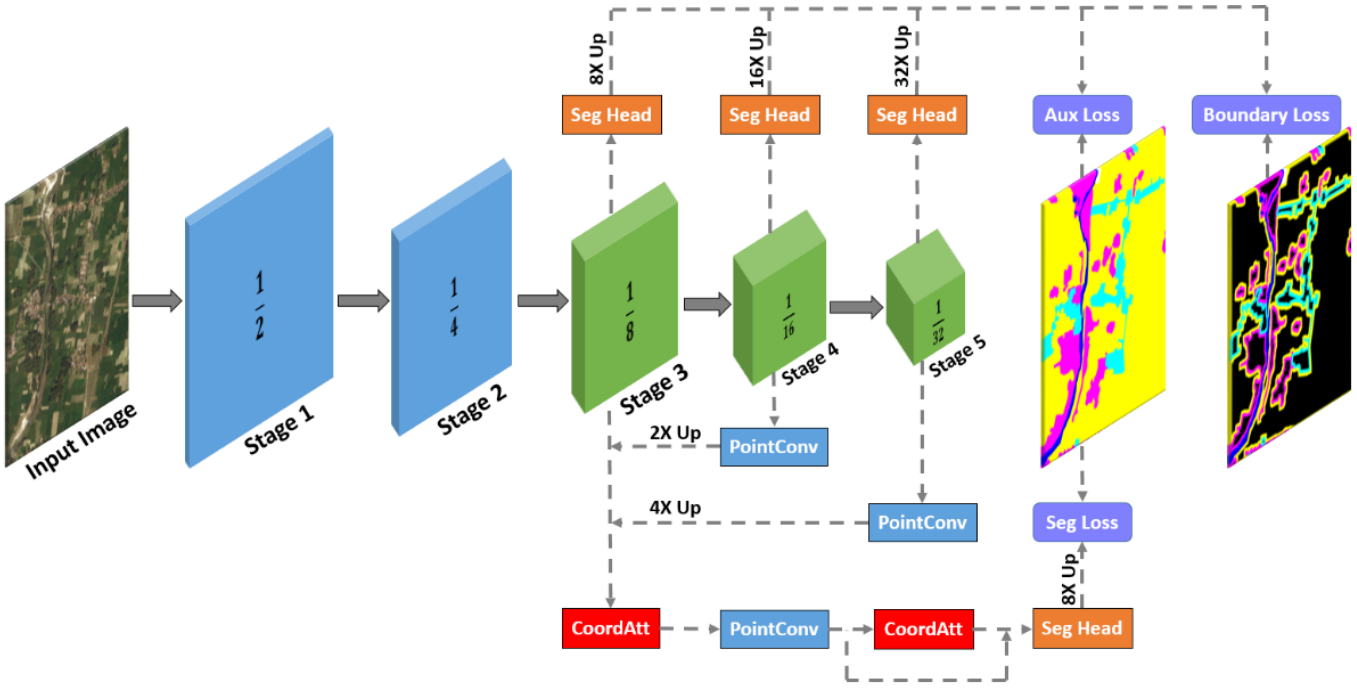

Figure 4. Architecture of MKANet. The blue cuboid represents the 3×3 convolution with stride 2, and the green cuboid represents 3×3 convolution with stride 2 plus one or more MKA modules.

3.2.1. Encoder

The encoder of MKANet has five stages, with each stage downsizing the feature maps by 2×. Its structure is detailed in Table 1.

Table 1. Encoder design.

Stage	Output Size	Operation	Output Channels
Input Image	2400 × 2400		3
Stage 1	1200 × 1200	ConvS2	$c/2$
Stage 2	600 × 600	ConvS2	c
Stage 3	300 × 300	ConvS2	$c \times 2$
	300 × 300	MKA × r	$c \times 2$
Stage 4	150 × 150	ConvS2	$c \times 4$
	150 × 150	MKA × r	$c \times 4$
Stage 5	75 × 75	ConvS2	$c \times 8$
	75 × 75	MKA × r	$c \times 8$

ConvS2: 3 × 3 convolution with stride 2, batch normalization, ReLU activation. c: the parameter controlling the width of the backbone. r: repeating time of MKA module.

Each stage begins with a 3 × 3 convolution of stride 2, followed by a batch normalization and ReLU activation. MKA modules are then repeated r times in each stage until stage 3. r controls the depth, while c controls the width of the backbone. MKANet has 3 typical sizes: Small ($c = 64, r = 1$), Base ($c = 96, r = 1$) and Large ($c = 128, r = 1$).

3.2.2. Decoder

The purpose of the first two stages is to extract simple, low-level features and to quickly downsize the resolution to reduce computations. Hence, the decoder only collects the deep context feature representations extracted by the MKA modules in stages 3 to 5.

To fuse multiscale feature maps, certain efficient networks, such as BiSeNet V1 [15] and STDC [25], use squeeze-and-excitation (SE) attention [26] to transform a feature tensor to a single feature vector via 2D global pooling and rescale the feature maps to selectively strengthen the important feature maps and to weaken the useless feature maps. Although SE attention can raise the representation power of a network at a low computational cost, it only encodes interchannel information without embedding position-sensitive information, which may help locate the objects of interest. To embed positional information into channel attention, Hou et al. proposed the coordinate attention module (CAM), which utilizes two 1D global pooling operations to aggregate features along the horizontal and vertical directions so that the two generated, direction-aware feature maps can capture long-range dependency along one spatial direction and preserve precise positional information along the other spatial direction. The detailed structure of the CAM is shown in Figure 5.

To tell the network "what" and "where" to attend, two CAMs are employed to fuse the multiscale feature maps output by stages 3 to 5. Specifically, the feature maps output by stage 4 and stage 5 are compressed from $4c$ and $8c$, respectively, to $2c$ in the channel by pointwise convolution, upsampled by 2× and 4×, respectively, and concatenated with the feature maps output by stage 3. After the feature maps are put through the first CAM to derive a combination of features with enhanced representation, a pointwise convolution is employed to promote the communication of information among the channels and to further compress the number of channels from $6c$ to $2c$. The compressed feature maps pass the second CAM with the residual connection.

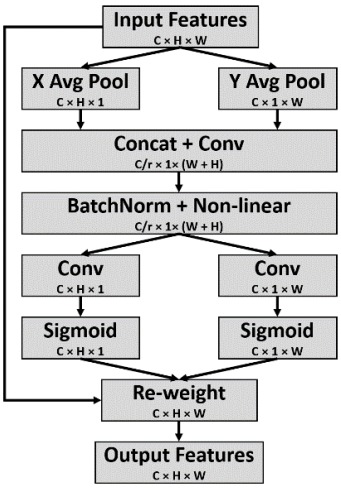

Figure 5. Structure of coordinate attention module (CAM). "X Avg Pool" and "Y Avg Pool" refer to 1D horizontal global pooling and 1D vertical global pooling, respectively.

3.3. Semantic Segmentation Losses

The semantic segmentation head, as illustrated in Figure 6, converts the output feature maps of the decoder into class logits, which are then upsampled by 8× to restore them to the same resolution as the input image. The upsampled class logits are compared with the ground truth by the main semantic segmentation loss function.

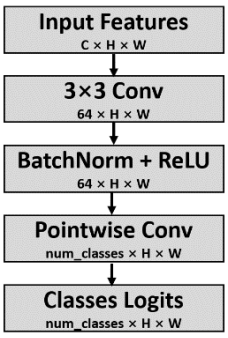

Figure 6. Structure of the semantic segmentation head.

To enhance the feature extraction ability of the MKA modules, three auxiliary semantic segmentation heads are added on top of the output features of stage 3 to stage 5 in the training phase. In the inference phase, the three auxiliary heads are discarded, without additional computational cost. The output class logits of the three auxiliary semantic segmentation heads are upsampled 8×, 16×, and 32× before being sent to three auxiliary semantic segmentation loss functions and three boundary loss functions.

3.4. Sobel Boundary Loss

The Sobel operator is a discrete differentiation gradient-based operator that computes the gradient approximation of the image intensity for edge detection. It employs two 3 × 3 kernels to convolve with the input image to calculate the vertical and horizontal derivative approximations, respectively.

To strengthen spatial detail learning and boundary recovery, Sobel operator convolution and dilation operation are performed on the ground truth mask to generate mask pixels that are within distance d from the contours and to use them as the target of the auxiliary boundary loss. The procedure is illustrated in Figure 7 and detailed in Algorithm 1. For a segment with a length or width less than $2d$, its whole ground truth is displayed in the boundary target mask, such as the small rivers, narrow strips of rangeland, and individual urban buildings shown in Figure 8. Therefore, all the small objects and segments, which

are more likely misjudged, are selected and penalized again by the Sobel Boundary Loss. For any object or segment whose width exceeds $2d$, only its contour of width d is displayed in the boundary target mask, such as the vast agricultural land, large rangeland, and urban residential community shown in Figure 8. By capturing category impurity areas, compared with the ground truth mask, in the boundary target mask, the pixel number ratio of the large segment to the small segments drops from quadratic with the segment size ratio to linear with the segment size ratio. In this way, the Sobel Boundary Loss guides the network to learn the features of spatial details.

Algorithm 1 Sobel boundary target generation

Input: Ground truth $\mathbf{Y}$, Sobel operator $\mathbf{S_x}$, $\mathbf{S_y}$, dilation rate d.
Output: Sobel boundary target $\hat{\mathbf{Y}}$.

$\mathbf{X_b} \leftarrow (|Conv(\mathbf{Y}, \mathbf{S_x})| + |Conv(\mathbf{Y}, \mathbf{S_y})|) > 0$
$\mathbf{X_d} \leftarrow Dilate(\mathbf{X_b})$
$\hat{\mathbf{Y}} \leftarrow \mathbf{Y} \otimes \mathbf{X_d}$
return $\hat{\mathbf{Y}}$

$\otimes$ means elementwise multiplication.

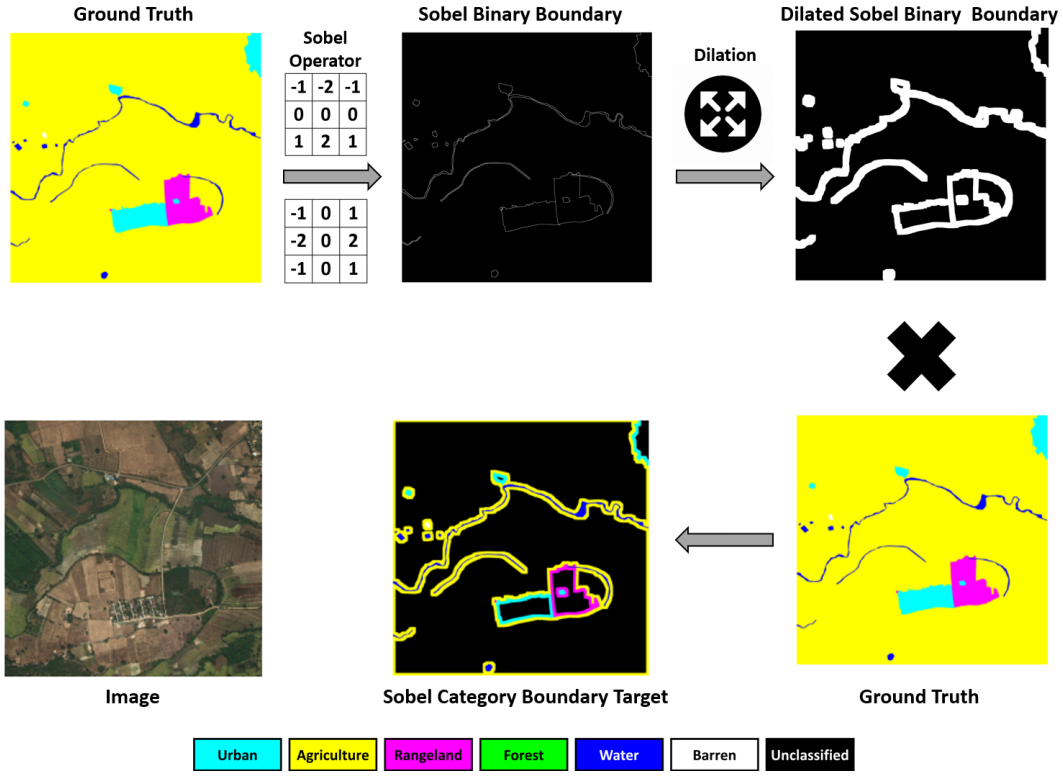

Figure 7. The procedure of generating Sobel category boundary target; dilation rate: $d = 50$ pixels.

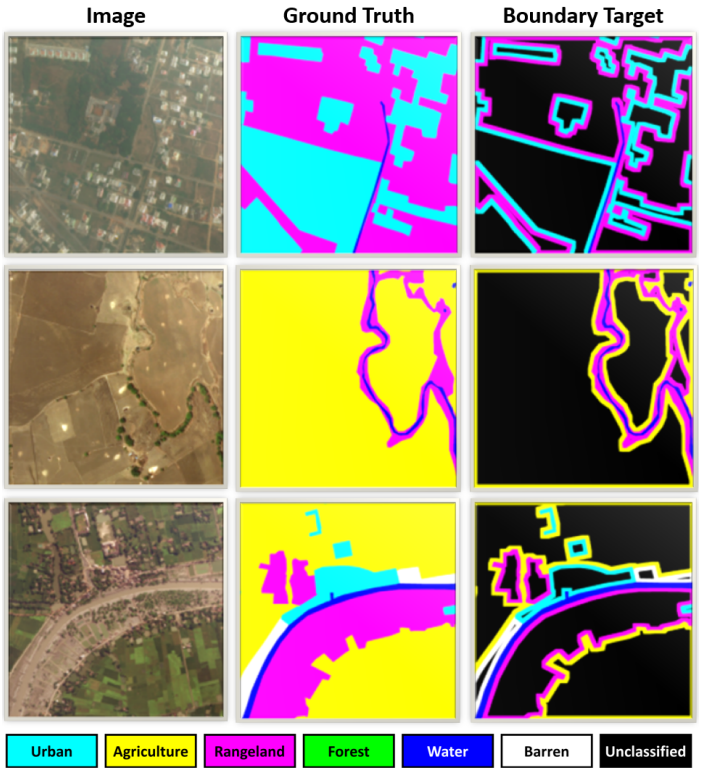

Figure 8. Images, ground truths, and boundary targets generated by the Sobel operator and dilation operation with $d = 50$ pixels.

d is a hyperparameter that controls the extent of contour pixels participating in the Sobel Boundary Loss calculation. It is not advisable to set d too small for four reasons. Firstly, different from general images in which objects have clear contours, the land boundaries in land-cover satellite images are comparatively vague. Secondly, setting buffer zones of width d benefits the network by learning how to discriminate different categories. Thirdly, if d is set too small, the samples participating in the boundary loss calculation are too scarce. Last, the margin of human labeling error should be considered. It is suggested to set d as the value equal to half of the smaller dimension of most small segments, so the whole bodies of most small segments would remain on the boundary target mask. Without loss of generality, d was set to 50 pixels in the illustration figures and experiments.

Any conventional loss function, for example the cross-entropy loss function or the Dice loss function, can be used for the Sobel Boundary Loss. As illustrated in Figure 7, in the dilated Sobel binary boundary mask, the pixels with value zero are relabeled as the category Unclassified and ignored in the Sobel Boundary Loss calculation.

3.5. Total Loss

The total loss L_t is the weighted sum of the main semantic segmentation loss L_m, auxiliary semantic segmentation losses L_a, and boundary losses L_b:

$$L_t = w_1 \times L_m + w_2 \times L_a + w_3 \times L_b \tag{1}$$

The values of the weights are adjusted according to the values of the loss functions and practical results. If the interiors of large segments are predicted fairly well but the boundaries or small segments are not predicted well, it is advisable to increase w_3. To evaluate whether the existence of the auxiliary losses would boost accuracy, without loss of generality, all the weights were set to 1 in the following experiments to avoid any hyperparameter-tuning trick.

4. Experimental Results

Since the MKA module specializes in multiscale feature extraction, to evaluate its effect, we built an image classification network based on the encoder of MKANet and conducted experiments on a scene classification dataset of multiscale remote sensing images. Then, we measured the inference speeds of MKANet at various image sizes to validate whether its architecture design could boost inference speed and support large image sizes. At last, we conducted experiments on two land-cover classification datasets to examine the accuracy of MKANet and the effectiveness of the Sobel Boundary Loss.

4.1. Image Classification

We added an image classification head as illustrated in Figure 9 on top of the MKANet encoder for the task of image classification, and named it MKANet-Class. We compared MKANet-Class with other state-of-the-art lightweight classification networks on the RSSCN7 [27] scene classification dataset of remote sensing images (Figure 10). RSSCN7 contains seven typical scene categories of image size 400 × 400 pixels. For each category, 400 images were sampled on four different scales with 100 images per scale. A total of 2800 images were resized to 384 × 384 pixels and split into a training set, validation set, and test set at a ratio of 2:1:1.

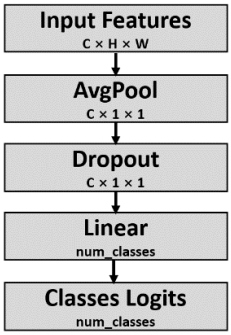

Figure 9. Structure of image classification head.

Figure 10. Images and categories of the RSSCN7 scene classification dataset.

Training details: Cross-entropy was selected as the loss function, and AdamW [28] was selected as the optimizer with a batch size of 32. The base learning rate was 0.001 with cosine decay. The number of epochs was 150 with a warmup strategy in the first 10 epochs.

For a fair comparison, all the networks were trained from scratch without pretraining on other datasets.

Data augmentation: random flipping, random rotation, and color jittering operations were employed on the input images in the training process.

As shown in Table 2, MKANet-Class Small outperformed other state-of-the-art lightweight classification networks with better accuracy and a significantly faster inference speed, which verified the effectiveness of the MKA module and justified the efficiency of the parallel branch design.

Table 2. Comparison of MKANet with other state-of-the-art, lightweight classification networks on the RSSCN7 dataset.

Method	Accuracy	FPS
MobileNetV2 [29]	90.43%	130.2
MobileNetV3 [30]	90.43%	107.9
EfficientNet-B0 [31]	91.28%	80.1
ShuffleNetV2 x1.0 [32]	91.85%	123.5
ResNet18 [12]	91.71%	178.8
ResNet34 [12]	92.14%	97.7
STDC1 [25]	92.43%	143.8
MKANet-Class Small	**92.85%**	**314.3**

Inference speed frames per second (FPS) were measured on a computer with an AMD 4800HS CPU, an NVIDIA RTX 2060 Max-Q 6G GPU, and a Pytorch environment.

4.2. Semantic Segmentation Inference Speed

For the semantic segmentation task, the inference speeds (FPS) of various networks were measured at four image sizes. As shown in Table 3, all the lightweight networks had an obvious advantage over large networks in large image size support and inference speed. On a computer with an NVIDIA RTX 3060 12G GPU, none of the large networks could process images with a resolution of 4096 × 4096 pixels, but all the lightweight networks could. MKANet Small was even capable of processing images up to 7200 × 7200 pixels and was approximately 2× faster than other lightweight networks and more than 13× faster than the large networks. The large size and fast acquisition speed of satellite images highlight the value of MKANet in accelerating the cognition speed of remote sensing images.

Table 3. Inference speed at 4 image sizes; the number of classes was 10.

Method	Inference Speed (FPS)			
	1024 × 1024	2048 × 2048	4096 × 4096	7200 × 7200
Large Networks:				
UNet (VGG16) [6]	4.27 (0.15×)	*	*	*
PSPNet (ResNet50) [33]	5.28 (0.19×)	1.51 (0.16×)	*	*
DeepLabV3+ (Xception) [8]	7.41 (0.26×)	2.03 (0.22×)	*	*
Lightweight Networks:				
ENet [14]	28.36 (1.0×)	9.31 (1.0×)	2.38 (1.0×)	0.71 (1.0×)
ABCNet [18]	35.24 (1.2×)	10.05 (1.1×)	2.70 (1.1×)	*
BiSeNetV1 (Resnet18) [15]	42.94 (1.5×)	11.85 (1.3×)	3.23 (1.4×)	*
BiSeNetV2 [16]	43.22 (1.5×)	11.74 (1.3×)	2.99 (1.3×)	*
STDC1 [25]	52.05 (1.8×)	15.13 (1.6×)	4.18 (1.8×)	*
STDC2 [25]	38.89 (1.4×)	11.59 (1.2×)	3.11 (1.3×)	*
DDRNet23 [17]	36.82 (1.3×)	10.04 (1.1×)	2.77 (1.2×)	0.97 (1.4×)
DABNet [19]	43.89 (1.6×)	11.54 (1.2×)	3.05 (1.3×)	*
MKANet Small	**98.24 (3.5×)**	**26.98 (2.9×)**	**7.47 (3.1×)**	**2.41 (3.4×)**
MKANet Base	59.38 (2.1×)	16.51 (1.8×)	4.61 (1.9×)	*
MKANet Large	43.77 (1.5×)	12.31 (1.3×)	3.44 (1.5×)	*

* means not executable due to GPU memory overflow. Inference speed frames per second (FPS) were measured on a computer with an INTEL i5-3470 CPU, an NVIDIA RTX 3060 12G GPU, and a Pytorch environment.

4.3. Land-Cover Classification

To assess the semantic segmentation performance of MKANet, experiments were conducted on two land-cover classification datasets of satellite images: DeepGlobe Land Cover [34] and GID Fine Land Cover Classification [5].

The DeepGlobe Land Cover dataset consists of RGB satellite images of size 2448 × 2448 pixels, with a pixel resolution of 50 cm. The total area size of the dataset is 1716.9 km^2. There are six rural land-cover categories. Only the labels of the original training set of the competition have been released; thus, the original training set, which contains 803 images, was split into a training set, validation set, and test set at a ratio of 2:1:1, as described in a previous study [10].

The GID Fine Land Cover Classification dataset consists of 10 submeter RGB satellite tiles of size 6800 × 7200 pixels. There are 15 land-cover categories. Due to the limitation of the GPU memory capacity, the 10 tiles were cropped into 90 subimages of size 2400 × 2400 pixels, and then the 90 subimages were split into a training set, validation set, and test set at a ratio of 3:1:1, similar to a previous study [35].

Training details: For all the lightweight networks, cross-entropy was selected as the loss function, and AdamW [28] was selected as the optimizer with a batch size of six, and the base learning rate was 0.001 with cosine decay. The networks were trained for 300 epochs with the DeepGlobe Land Cover dataset and for 500 epochs with the GID dataset, using a warmup strategy in the first 10 epochs. For a fair comparison, all the networks were trained from scratch without pretraining on other datasets.

Data augmentation: Random flipping, random rotation, random scaling of rates (0.7, 0.8, 0.9, 1.0, 1.25, 1.5, 1.75), random cropping into size 1600 × 1600 pixels, and color jittering operations were employed on the input images during the training process. In the test process, no data augmentation operations were implemented.

Evaluation metrics: The performance of the networks was evaluated by the mean intersection over union (MIoU) and the mean F1 score which are defined as:

$$\mathrm{MIoU} = \frac{1}{N}\sum_{c=1}^{N}\frac{TP_c}{TP_c + FP_c + FN_c}. \tag{2}$$

$$\mathrm{MF1} = \frac{1}{N}\sum_{c=1}^{N}\frac{2 \times \frac{TP_c}{TP_c+FP_c} \times \frac{TP_c}{TP_c+FN_c}}{\frac{TP_c}{TP_c+FP_c} + \frac{TP_c}{TP_c+FN_c}}. \tag{3}$$

where N represents the number of categories, and TP_c, FP_c, and FN_c denote the number of true positive pixels, false positive pixels, and false negative pixels, respectively, in category c.

4.3.1. DeepGlobe Land Cover Dataset Experimental Results

The DeepGlobe Land Cover Classification dataset provides high-resolution submeter satellite imagery focusing on rural areas. Due to the variety of land cover types, it is more challenging than the ISPRS Vaihingen and Potsdam datasets [36] and the Zeebruges dataset [37]. The image size of this dataset is 2448 × 2448 pixels, and only lightweight networks can support such large-size images in training and predicting. We selected this dataset to evaluate the performance of MKANet on high-spatial-resolution remote sensing imagery with image size beyond 2K pixels.

As presented in Tables 4 and 5, MKANets led other competitive lightweight networks by at least 3% and even surpassed the large networks with pretrained backbones. In a previous study [10], to fit the large networks into GPU memory, the authors compressed the images to half size and then divided them into subimages with a resolution of 512 × 512 pixels. For the large networks, spatial detail loss due to compression offsets their stronger feature extraction ability enabled by a larger backbone, while long-range context information loss due to subdivision weakens their better modeling ability from a more complex

structure. Hence, for large-sized remote sensing patches, lightweight networks have their advantage and can have comparable and even better accuracy than large networks.

Table 4. Comparison of MKANets with other state-of-the-art lightweight networks on the DeepGlobe Land Cover dataset.

Method	Category IoU (%)						MIoU (%)	Speed (FPS)
	Urban	Agricult.	Range	Forest	Water	Barren		
Proportion (%)	9.35	56.76	10.21	13.75	3.74	6.14		
ENet [14]	71.14	80.86	0.42	75.44	73.10	49.41	58.39	6.96
ABCNet [18]	73.52	82.03	29.03	75.95	72.50	55.94	64.83	7.15
BiSeNetV1 (Resnet18) [15]	72.68	83.91	28.09	78.97	74.71	56.71	65.85	8.41
BiSeNetV2 [16]	72.46	82.23	23.52	77.86	69.37	51.03	62.75	8.42
STDC1 [25]	74.68	85.05	31.75	76.82	75.34	59.11	67.12	11.81
STDC2 [25]	73.48	83.79	30.57	76.27	73.48	58.11	65.95	8.76
DDRNet23 [17]	75.01	84.53	32.62	77.81	77.29	58.50	67.63	7.21
DABNet [19]	74.44	84.69	33.51	79.03	75.57	58.63	67.64	8.04
MKANet Small	76.14	86.04	39.07	80.62	79.28	62.96	70.68	**19.18**
MKANet Base	76.35	**87.30**	39.63	81.44	**81.16**	64.77	71.78	11.51
MKANet Large	**76.53**	**87.30**	**40.36**	**81.66**	81.04	**65.08**	**72.00**	8.59

Inference speed frames per second (FPS) were measured on a computer with an INTEL i5-3470 CPU, an NVIDIA RTX 3060 12G GPU, and a Pytorch environment.

Table 5. Comparison of MKANets with large networks on the DeepGlobe Land Cover dataset.

Method	MIoU (%)	MF1 (%)
UNet (Res2Net50) [6]	67.57	79.50
PSPNet (Res2Net50) [33]	69.45	81.07
DeepLabV3 (ResNet50) [23]	68.39	80.27
DeepLabV3 (ResNet101) [23]	68.94	80.55
DeepLabV3+ (Res2Net50) [8]	69.12	80.85
DeepLabV3+ (Res2Net101) [8]	69.39	81.06
EncNet (Res2Net50) [38]	68.53	80.42
EncNet (Res2Net101) [38]	68.60	80.40
PSANet (ResNet50) [39]	68.27	80.13
GCNet (ResNet50) [40]	69.09	80.47
DEANet [10]	71.80	82.60
MKANet Small	70.68	81.69
MKANet Base	71.78	82.42
MKANet Large	**72.00**	**82.62**

All the large networks employed the backbones pretrained on ImageNet [41]. For all the large networks, the experimental results reported by Wei et al. [10] were quoted.

As illustrated in Figure 11, the predicted masks demonstrate that compared with other lightweight networks, MKANets can better identify land cover of small dimensions, for example, the river. This superiority is attributed to two factors, one is the MKA module, which has multiscale receptive fields without losing spatial resolution, so spatial details can be preserved. As shown in the lower right subfigure, even without any auxiliary loss, MKANet-NA Large still predicted the river better than other networks. The other factor is the Sobel Boundary Loss, which benefits the network in small segments recognition and boundary recovery. As shown in the bottom of Figure 11, MKANet Base and Large predicted the river more precisely than MKANet-NA Large.

The above observation agrees with the per-category IoUs in Table 4, where MKANets lead other networks by a large margin in the categories of small segments, such as water and range.

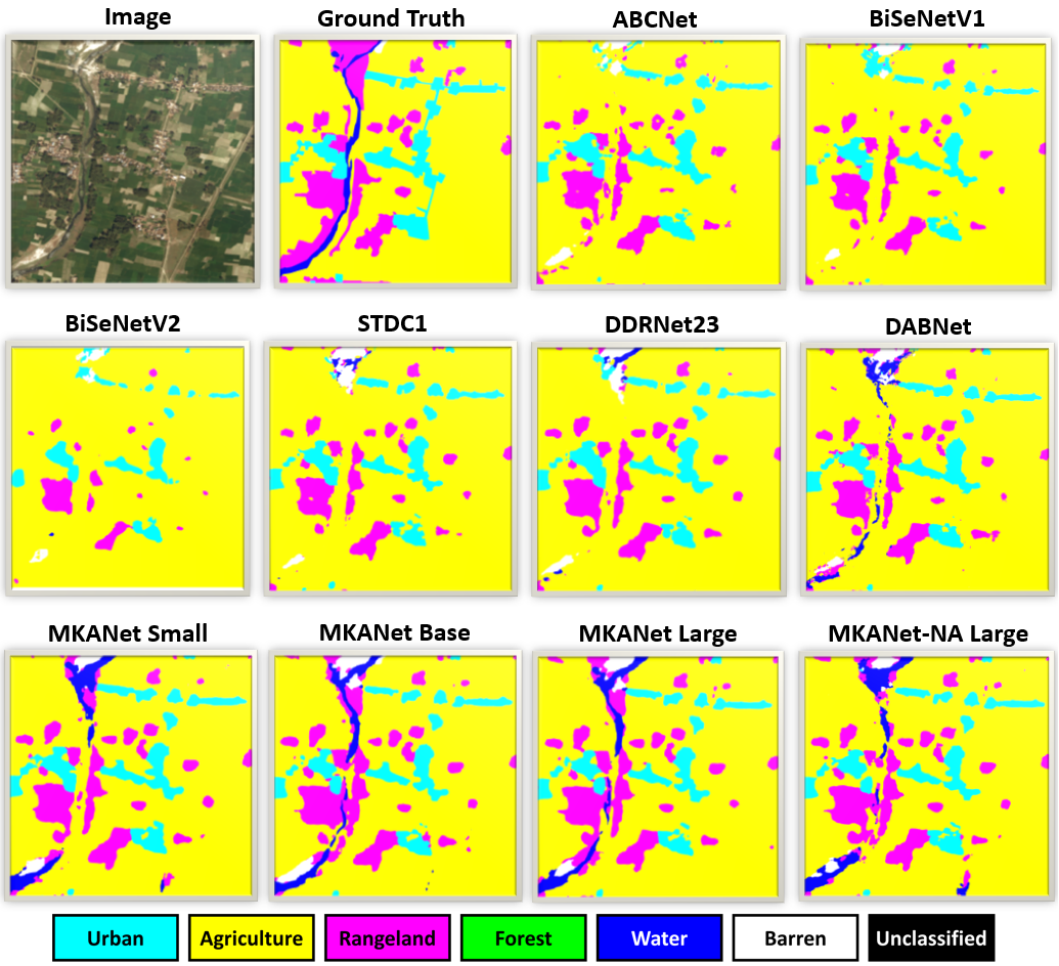

Figure 11. Comparison of the masks predicted by different methods on the DeepGlobe Land Cover dataset.

4.3.2. GID Fine Land Cover Classification Dataset Experimental Results

The GID Fine Land Cover Classification dataset is very challenging due to its small sample size (only 10 tiles of size 6800 × 7200 pixels) and highly skewed category distribution, within which the proportions of the three categories are scarce, at less than 1%. We selected this rich category dataset to assess the discrimination ability of MKANet on fine and similar land-cover categories, and also evaluate its robustness with regard to highly unbalanced remote sensing imagery.

As shown in Tables 6 and 7, MKANets outperformed all other lightweight networks and the large networks by a large margin. The per-category IoUs in Table 6 indicated that the superiority of MKANets was mainly manifested in minor categories, small dimensional categories, and hard-discriminating categories. In these categories, MKANet Small surpassed the average IoU of other lightweight networks by more than 18%. For example, the large category of farmland (consisting of paddy land, irrigated land, and dry cropland) was highly skewed in distribution, the samples of irrigated land were approximately 10 times greater than those of paddy land and dry cropland. MKANet Small exceeded the average IoU of other lightweight networks by 37.7% on paddy land and 25.1% on dry cropland.

Table 6. Comparison of MKANets with other state-of-the-art lightweight networks on the GID dataset.

Method	Category IoU (%)															MIoU (%)	Speed (FPS)
	Indust.	Urban	Rural	Traff.	Paddy	Irrig.	Crop	Gard.	Arbor	Shrub	Natur.	Artif.	River	Lake	Pond		
Prop.(%)	7.26	13.96	5.59	4.74	4.38	36.39	2.86	0.82	9.04	0.3	1.68	0.83	5.65	3.16	3.34		
ENet	49.98	56.85	45.31	1.08	0.00	67.74	29.29	0.00	86.79	0.00	62.28	0.00	38.78	47.04	3.67	32.59	6.73
ABCNet	47.09	58.88	41.21	34.01	17.47	70.32	7.29	12.05	88.96	16.48	83.43	22.50	39.74	43.10	57.85	42.69	7.20
BiSeNetV1	61.09	63.63	41.75	42.56	16.70	72.72	3.73	11.27	87.82	17.44	84.42	45.18	64.35	50.94	57.10	48.05	8.74
BiSeNetV2	59.26	65.17	49.34	43.99	56.00	79.00	3.13	9.38	93.69	9.32	86.40	55.42	39.37	44.29	50.37	49.61	8.76
STDC1	62.77	64.07	46.37	52.91	13.22	73.72	4.97	9.35	69.36	3.73	86.27	56.94	86.15	75.78	59.21	50.99	12.09
STDC2	57.14	60.13	44.04	45.45	29.12	72.98	13.89	11.69	87.15	6.93	87.53	44.16	62.94	54.08	54.78	48.80	9.04
DDRNet23	63.57	65.11	50.43	59.66	53.26	80.12	11.60	9.87	84.83	0.00	87.22	63.23	86.82	75.92	53.11	56.32	7.46
DABNet	67.52	71.08	59.64	71.10	73.86	83.79	24.44	11.72	95.48	4.15	**93.27**	56.99	80.38	63.41	66.73	61.57	8.37
MKANet S	67.42	69.68	55.82	70.91	70.12	82.67	37.41	16.44	93.82	10.98	92.30	61.39	86.34	**78.90**	69.70	64.26	**19.77**
MKANet B	**68.82**	**71.39**	56.83	71.92	**76.18**	83.43	35.57	13.40	96.00	12.49	92.22	65.08	86.70	74.46	71.93	65.09	12.01
MKANet L	67.90	70.35	**60.00**	**72.88**	73.31	**84.50**	**41.20**	**20.76**	**96.75**	**19.66**	93.22	**68.77**	**89.30**	77.37	**73.49**	**67.30**	8.86

Inference speed frames per second (FPS) were measured on a computer with an INTEL i5-3470 CPU, an NVIDIA RTX 3060 12G GPU, and a Pytorch environment.

Table 7. Comparison of MKANets with large networks on the GID dataset.

Method	Category F1 Score (%)															MF1 (%)	MIoU (%)
	Indust.	Urban	Rural	Traff.	Paddy	Irrig.	Crop	Gard.	Arbor	Shrub	Natur.	Artif.	River	Lake	Pond		
Prop. (%)	7.26	13.96	5.59	4.74	4.38	36.39	2.86	0.82	9.04	0.3	1.68	0.83	5.65	3.16	3.34		
FCN [42]	68.92	73.99	64.51	68.73	74.08	84.20	68.42	24.34	87.79	4.07	53.04	25.86	83.30	66.66	77.47	62.89	49.52
PSPNet [33]	69.18	74.41	64.87	68.09	74.53	84.69	68.23	25.26	87.84	10.36	51.87	29.07	83.15	66.71	77.40	63.57	49.98
DeepLabV3+	69.11	75.02	64.96	67.33	75.26	85.68	69.54	18.45	88.25	5.57	49.88	33.01	88.36	79.00	80.21	64.45	51.59
DANet [43]	69.77	74.81	65.62	69.19	75.58	84.99	66.72	20.71	88.33	13.53	59.18	29.45	83.46	67.93	78.10	64.29	50.78
SCAttNet [44]	68.64	73.97	64.63	64.42	71.47	85.25	70.33	22.85	87.57	3.28	56.59	24.30	86.83	73.39	77.23	63.24	50.12
MSCA [45]	69.75	76.58	66.63	68.78	71.22	85.91	66.74	8.41	87.59	8.46	58.55	23.26	89.17	76.68	80.02	63.72	51.15
LANet [46]	69.03	75.62	65.29	68.03	72.21	85.57	67.39	7.83	88.10	10.24	54.51	30.60	87.28	74.29	78.80	63.51	50.60
WiCoNet [35]	69.61	75.32	65.50	67.23	73.92	86.37	**72.47**	31.80	88.53	13.85	47.71	42.60	87.88	76.55	81.65	65.55	52.48
MKANet S	80.54	82.13	71.65	82.98	82.43	90.51	54.46	28.23	96.81	19.79	96.00	76.08	92.67	**88.20**	82.14	74.97	64.26
MKANet B	**81.53**	**83.30**	72.48	83.67	**86.48**	90.96	52.47	23.63	97.96	22.21	95.95	78.84	92.88	85.36	83.68	75.43	65.09
MKANet L	80.88	82.60	**75.00**	**84.31**	84.60	**91.60**	58.36	**34.38**	**98.35**	**32.86**	**96.49**	**81.49**	**94.35**	87.24	**84.72**	**77.82**	**67.30**

For all the large networks, the experimental results reported by Ding et al. [35] were quoted.

As illustrated in Figure 12, small-dimensional land covers, such as rivers, streets, and rows of rural houses, were better classified by MKANets. MKANet Small exceeded the average IoU of other efficient networks by 27.1% on traffic land, 18.3% on artificial grass, 24% on rivers, 22.1% on lakes, and 19.3% on ponds. As shown in the bottom right subfigure, even without any auxiliary loss, MKANet-NA Large still outperformed other lightweight networks in small segment discrimination and spatial detail reconstruction.

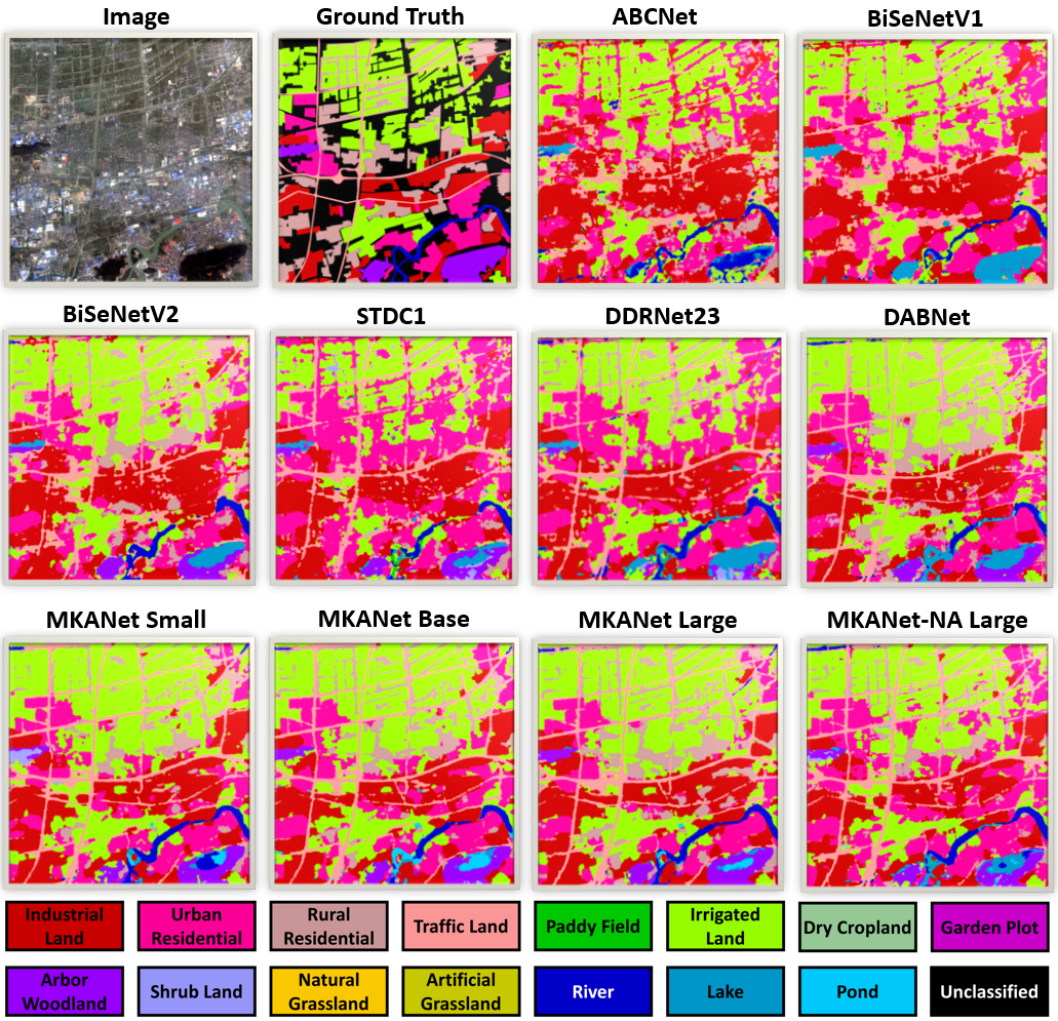

Figure 12. Comparison of the masks predicted by different methods on the GID Fine Land Cover Classification dataset.

4.4. Ablation Analysis

To validate the effectiveness of each component, ablation analysis experiments were conducted based on the DeepGlobe Land Cover dataset.

4.4.1. The Influence of Input Image Size on Prediction Accuracy

To investigate the influence of input image size on prediction accuracy, the original DeepGlobe dataset images with a resolution of 2448 × 2448 pixels were cropped into patches with resolutions of 512 × 512 pixels, 1024 × 1024 pixels, and 1600 × 1600 pixels. The prediction accuracies of these three patch sizes were compared with that of the original image size. As shown in Table 8, the smaller the patch size is, the lower the prediction accuracy is, which demonstrates that the loss of long-range context information caused by cropping images into small patches would cause misjudgments.

Table 8. Comparison of different variants.

Method	Category IoU (%)						MIoU (%)
	Urban	Agricult.	Range	Forest	Water	Barren	
MKANet Large, 2448 × 2448 pixels	76.53	87.30	40.36	81.66	81.04	65.08	**72.00**
MKANet Large, 1600 × 1600 pixels	76.78	87.38	39.84	81.36	80.92	64.50	71.80
MKANet Large, 1024 × 1024 pixels	76.27	86.78	38.99	80.40	81.15	64.46	71.34
MKANet Large, 512 × 512 pixels	76.04	85.81	36.35	79.24	79.54	59.06	69.34
MANet Large	76.87	86.88	37.63	81.09	83.42	63.68	71.59
MKANet-Concat Large	76.51	86.62	37.53	80.01	81.49	61.33	70.58
MKANet-NB Large	75.71	87.25	35.51	81.83	80.51	62.19	70.50
MKANet-NA Large	75.97	86.48	34.22	80.14	79.50	60.99	69.55

4.4.2. Effectiveness of Kernel-Sharing Atrous Convolution

To evaluate the effect of kernel-sharing atrous convolutions in the MKA module, a variant network with kernel-sharing atrous convolutions replaced by regular atrous convolutions was built and denoted as MANet. As shown in Table 8, compared with regular atrous convolutions, kernel-sharing atrous convolutions had stronger feature extraction ability.

4.4.3. Effectiveness of Coordinate Attention Module

To assess the effect of the two coordinate attention modules (CAMs) in the decoder, a variant network with CAMs replaced by simple concatenation operations was built and denoted as MKANet-Concat. As shown in Table 8, compared with simple concatenation operations, the two CAMs better fused the multiscale features from various stages and augmented the representations of the objects of interest.

4.4.4. Effectiveness of Auxiliary Losses

To estimate the contribution of auxiliary semantic segmentation loss and auxiliary boundary loss, a variant network without any auxiliary loss was built and denoted as MKANet-NA, and another variant network without auxiliary boundary loss was built and denoted as MKANet-NB. As shown in Table 8, on the DeepGlobe Land Cover dataset, auxiliary semantic segmentation loss improved the MIoU by nearly 1%, and auxiliary boundary loss further boosted the MIoU by approximately 1.5%. On the GID Fine Land Cover Classification dataset, both improvements were broadened to 1.9% (Table 9). The IoUs of most categories were boosted by the two kinds of auxiliary losses, indicating that these auxiliary heads could promote spatial detail learning at lower levels and semantic context learning at higher levels.

Table 9. Comparison of MKANets with and without auxiliary losses on the GID Fine Land Cover Classification dataset.

Method	Category IoU (%)														MIoU (%)
	Indust.	Urban	Rural	Traff.	Paddy	Irrig.	Crop	Gard.	Arbor	Shrub	Natur.	Artif.	River	Lake	Pond	
MKANet Large	67.90	70.35	**60.00**	**72.88**	73.31	**84.50**	**41.20**	**20.76**	**96.75**	**19.66**	**93.22**	68.77	**89.30**	77.37	**73.49**	**67.30**
MKANet-NB Large	**69.78**	**71.11**	57.96	70.83	**73.90**	82.78	39.63	13.49	96.70	15.60	91.38	**69.30**	84.27	72.38	71.63	65.38
MKANet-NA Large	69.20	69.31	53.34	69.70	70.35	82.32	39.75	10.99	92.09	2.98	90.48	64.59	88.96	**82.63**	65.88	63.51

To visualize the effect of the auxiliary losses, the predicted labels of the above three networks are displayed in Figure 13. The results showed that the auxiliary losses, especially the boundary loss, can guide the networks to better recognize small segments and restore boundaries, which is in accordance with our design.

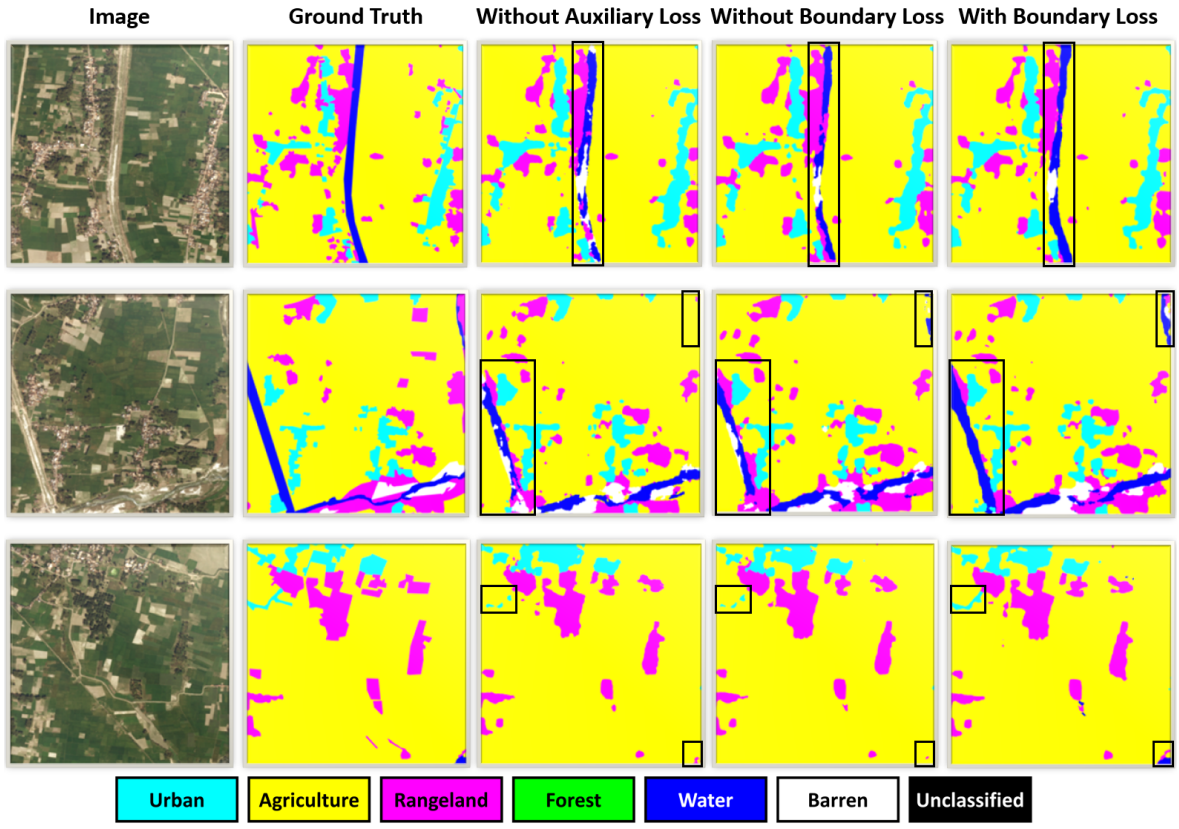

Figure 13. Comparison of the masks predicted by MKANets without any auxiliary loss, without auxiliary boundary loss, and with both types of auxiliary losses.

4.4.5. Stacking More MKA Modules per Stage

MKANet can be expanded not only in width by increasing the number of base channels c but also in depth by increasing the repeating times r of MKA modules in each stage. As shown in Table 10, the performance of MKANet increased by stacking additional MKA modules in each stage.

Table 10. Comparison of MKANets with different widths and depths.

Method	MIoU (%)	Speed FPS
MKANet $c = 64, r = 1$	70.68	**19.18**
MKANet $c = 64, r = 2$	71.70	14.22
MKANet $c = 96, r = 1$	71.78	11.51
MKANet $c = 96, r = 2$	71.84	8.47
MKANet $c = 128, r = 1$	**72.00**	8.59

Inference speed frames per second (FPS) were measured on a computer with an INTEL i5-3470 CPU, an NVIDIA RTX 3060 12G GPU, and a Pytorch environment.

4.4.6. The Optimal Value for the Number of Branches

In the MKA module, the number of parallel branches b is a hyperparameter, which is proportional to the receptive field and computational cost. To determine the optimal value for b, different values were tested. As shown in Table 11, with one more branch of a larger dilation rate atrous convolution to sense larger-scale features, MKANet $b = 3$ performed better than MKANet $b = 2$ by 0.54% on the MIoU metric. However, as more branches were added, the MIoU dropped. This negative effect was mainly attributed to Part 3 of the MKA module, where the output features of each branch were concatenated and a pointwise convolution was then applied to them to compress the channels to $1/b$. The larger b is, the more information losses there are in the channel compression process.

$b = 3$ strikes a good balance among the multiscale receptive field, information loss, and computation efficiency; thus, b defaults to the optimal value of 3 in the MKA module.

Table 11. Comparison of MKANets with a variable number of parallel branches on the DeepGlobe Land Cover dataset.

Method	Category IoU (%)						MIoU (%)
	Urban	Agricult.	Range	Forest	Water	Barren	
MKANet $b = 2, c = 96, r = 1$	76.65	87.22	36.79	80.86	**81.95**	63.99	71.24
MKANet $b = 3, c = 96, r = 1$	76.35	**87.30**	**39.63**	**81.44**	81.16	64.77	**71.78**
MKANet $b = 4, c = 96, r = 1$	**76.71**	87.20	38.51	81.31	81.77	63.92	71.57
MKANet $b = 5, c = 96, r = 1$	76.39	87.17	37.10	81.01	81.92	**65.15**	71.46

5. Discussion

Aimed at the characteristics of multiscale and large image size of top view remote sensing imagery, we merged the specialized multibranch module and the shallow architecture design into MKANet. Through extensive experiments and an ablation analysis, MKANet reached our initial expectation and alleviated the three problems (slow inference speed, incapability of processing large size image patches, and easy misjudgment on boundaries and small segments) that restrict the practical applications of semantic segmentation networks in remote sensing imagery.

In the land-cover classification experiments, the original images of 2448 × 2448 pixels and large patches of 2400 × 2400 pixels were employed as the input of MKANet. Compared with a cropping size of 512 × 512 pixels or even smaller in most existing studies, the number of subimages was reduced to 1/25. In the inference speed experiments, MKANet Small could support an even larger image size of 7200 × 7200 pixels on an NVIDIA RTX 3060 12G GPU and image size up to 10k × 10k pixels on an NVIDIA RTX3090 24G GPU. Its friendly support of large subimages greatly alleviated spatial detail loss due to downsampling or long-range context information loss due to cropping. Meanwhile, MKANet Small was approximately 2× faster than other lightweight networks and 13× faster than large networks. Both merits highlight the value of MKANet for accelerating the perception and cognition speed of remote sensing imagery.

In response to the problem that prediction errors are more likely to occur on boundaries and small segments, the Sobel operator's convolution and dilation operation were innovatively utilized to capture category impurity areas, exploit boundary information, and exert an extra penalty on boundaries and small segments misjudgment. Both quantitative metrics and visual interpretations verified that the Sobel Boundary Loss could promote spatial detail learning and boundary reconstruction.

For the task of land-cover classification, MKANet successfully raised the benchmark on accuracy and demonstrated that if lightweight efficient networks were properly designed, they could have comparable accuracy with that of large networks. In addition, due to the merits of a fast inference speed and a low requirement on hardware, lightweight networks have immense potential in practical applications and are equally important. Notably, MKANet outperformed other state-of-the-art lightweight networks with a significantly better accuracy.

6. Conclusions

Conventional semantic segmentation networks are not able to process large-size satellite images under mainstream hardware resources. To avoid a loss of spatial resolution due to downsampling and a loss of long-range context information due to cropping, we proposed an efficient lightweight network for land-cover classification of satellite remote sensing imagery. Extensive experimental results demonstrated that the MKANet achieved state-of-the-art speed–accuracy trade-off; it ran 2× faster than other lightweight networks and could support large-size images. In addition, the proposed Sobel Boundary Loss could enhance boundary and small segment discrimination.

With the increasing demands of onboard autonomous applications, the next generation of satellites are required to possess the ability to process collected images and execute intelligent tasks on orbit. Although MKANet consumes much less hardware resources than other networks, it still cannot satisfy the tight constraints imposed on the onboard embedded systems. In future research, we will focus on image onboard processing and explore effective methods, such as network pruning, parallel optimization, and hardware acceleration, for embedded system adaptation and deployment.

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

Funding: This work was supported by the National Natural Science Foundation of China (no. 61901307), Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University (no. 20E01), Scientific Research Foundation for Doctoral Program of Hubei University of Technology (no. BSQD2020054, no. BSQD2020055).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Zhang, Z.; Qu, Z.; Liu, S.; Li, D.; Cao, J.; Xie, G. Expandable On-Board Real-Time Edge Computing Architecture for Luojia3 Intelligent Remote Sensing Satellite. *Remote Sens.* **2022**, *14*, 3596. [CrossRef]
2. Li, D.; Wang, M.; Dong, Z.; Shen, X.; Shi, L. Earth observation brain (EOB): An intelligent earth observation system. *Geo-Spat. Inf. Sci.* **2017**, *20*, 134–140. [CrossRef]
3. Wang, M.; Zhang, Z.; Zhu, Y.; Dong, Z.; Li, Y. Embedded GPU implementation of sensor correction for on-board real-time stream computing of high-resolution optical satellite imagery. *J.-Real-Time Image Process.* **2018**, *15*, 565–581. [CrossRef]
4. Mi, W.; Zhiqi, Z.; Zhipeng, D.; Shuying, J.; SU, H. Stream-computing based high accuracy on-board real-time cloud detection for high resolution optical satellite imagery. *Acta Geod. Cartogr. Sin.* **2018**, *47*, 760.
5. Tong, X.Y.; Xia, G.S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sens. Environ.* **2020**, *237*, 111322. [CrossRef]
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. Jégou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 11–19.
8. 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.
9. Wang, Z.; Wang, J.; Yang, K.; Wang, L.; Su, F.; Chen, X. Semantic segmentation of high-resolution remote sensing images based on a class feature attention mechanism fused with Deeplabv3+. *Comput. Geosci.* **2022**, *158*, 104969. [CrossRef]
10. Wei, H.; Xu, X.; Ou, N.; Zhang, X.; Dai, Y. DEANet: Dual Encoder with Attention Network for Semantic Segmentation of Remote Sensing Imagery. *Remote Sens.* **2021**, *13*, 3900. [CrossRef]
11. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
12. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
13. Zhang, X.; Wang, Z.; Cao, L.; Wang, M. A Remote Sensing Land Cover Classification Algorithm Based on Attention Mechanism. *Can. J. Remote Sens.* **2021**, *47*, 835–845. [CrossRef]
14. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.
15. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 325–341.
16. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 3051–3068. [CrossRef]
17. Hong, Y.; Pan, H.; Sun, W.; Jia, Y. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv* **2021**, arXiv:2101.06085.

18. Li, R.; Zheng, S.; Zhang, C.; Duan, C.; Wang, L.; Atkinson, P.M. ABCNet: Attentive bilateral contextual network for efficient semantic segmentation of Fine-Resolution remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *181*, 84–98. [CrossRef]
19. Li, G.; Yun, I.; Kim, J.; Kim, J. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. *arXiv* **2019**, arXiv:1907.11357.
20. Li, X.; Zhou, Y.; Pan, Z.; Feng, J. Partial order pruning: For best speed/accuracy trade-off in neural architecture search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9145–9153.
21. Yuan, Y.; Xie, J.; Chen, X.; Wang, J. Segfix: Model-agnostic boundary refinement for segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 489–506.
22. Huang, Y.; Wang, Q.; Jia, W.; Lu, Y.; Li, Y.; He, X. See more than once: Kernel-sharing atrous convolution for semantic segmentation. *Neurocomputing* **2021**, *443*, 26–34. [CrossRef]
23. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
24. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
25. Fan, M.; Lai, S.; Huang, J.; Wei, X.; Chai, Z.; Luo, J.; Wei, X. Rethinking bisenet for real-time semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9716–9725.
26. 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, 18–23 June 2018; pp. 7132–7141.
27. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
28. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
30. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
31. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*; PMLR: New York, NY, USA, 2019; pp. 6105–6114.
32. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
33. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
34. Demir, I.; Koperski, K.; Lindenbaum, D.; Pang, G.; Huang, J.; Basu, S.; Hughes, F.; Tuia, D.; Raskar, R. Deepglobe 2018: A challenge to parse the earth through satellite images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 172–181.
35. Ding, L.; Lin, D.; Lin, S.; Zhang, J.; Cui, X.; Wang, Y.; Tang, H.; Bruzzone, L. Looking outside the window: Wide-context transformer for the semantic segmentation of high-resolution remote sensing images. *arXiv* **2021**, arXiv:2106.15754.
36. The International Society for Photogrammetry and Remote Sensing. 2D Semantic Labeling Contest. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 5 January 2022).
37. Campos-Taberner, M.; Romero-Soriano, A.; Gatta, C.; Camps-Valls, G.; Lagrange, A.; Le Saux, B.; Beaupere, A.; Boulch, A.; Chan-Hon-Tong, A.; Herbin, S.; et al. Processing of extremely high-resolution Lidar and RGB data: Outcome of the 2015 IEEE GRSS data fusion contest–part a: 2-D contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 5547–5559. [CrossRef]
38. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7151–7160.
39. 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.
40. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
41. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
42. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 640–651. [CrossRef] [PubMed]
43. Nam, H.; Ha, J.W.; Kim, J. Dual attention networks for multimodal reasoning and matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 299–307.

44. Li, H.; Qiu, K.; Chen, L.; Mei, X.; Hong, L.; Tao, C. SCAttNet: Semantic segmentation network with spatial and channel attention mechanism for high-resolution remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 905–909. [CrossRef]
45. Zhang, J.; Lin, S.; Ding, L.; Bruzzone, L. Multi-scale context aggregation for semantic segmentation of remote sensing images. *Remote Sens.* **2020**, *12*, 701. [CrossRef]
46. 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]