MKANet: A Lightweight Network with Sobel Boundary Loss for Efficient Land-cover Classification of Satellite Remote Sensing Imagery

Zhiqi Zhang, Wen Lu, Jinshan Cao, Guangqi Xie

Abstract—Land cover classification is a multi-class segmentation task to classify each pixel into a certain natural or man-made category of the earth 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 down sampling or cropping them into small patches less than 512×512 pixels before sending them to a deep neural network. However, down sampling images incurs spatial detail loss, renders small segments hard to discriminate, and reverses the spatial resolution progress obtained by decades of years 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 parallel and shallow architecture to boost inference speed and friendly support image patches more than 10X larger. To enhance boundary and small segments discrimination, we also propose a method that captures category impurity areas, exploits boundary information and exerts an extra penalty on boundaries and small segment misjudgment. Both visual interpretations and quantitative metrics of extensive experiments demonstrate that MKANet acquires state-of-the-art accuracy on two land-cover classification datasets and infers 2X faster than other competitive lightweight networks. All these merits highlight the potential of MKANet in practical applications.

Index Terms—Semantic segmentation, convolutional neural network, remote sensing image, land-cover classification.

I. INTRODUCTION

NOWADAYS, various satellite constellations with shorter revisit period 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. For example, China has 30 to 50 high-resolution remote sensing satellites in orbit, by conservative estimation, several hundreds TB data is acquired every day [1]. However, with regard to the acquisition speed, rapid intelligent processing of remote sensing data still lags. 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 multi-class segmentation task to classify each pixel into a certain natural or man-made category of the earth surface, such as water, soil, natural vegetation, crops, and human infrastructure. The land cover and its change influence the ecosystem, human health, society development and economic growth. Last several decades of years have witnessed the improvement of spatial resolution of remote sensing imagery from 30m to sub-meter. With richer details and structural information of objects emerging in remote sensing imagery, land cover classification methods have shifted from discriminating in spectral or spectral-spatial information of local pixels to extracting contextual information and spatial relationship of ground objects [2]. Among them, deep neural networks (DNN) have been widely used for their strong feature extraction and high-level semantic modelling ability. However, large computational resources consumption brings slow inference speed and restricts the practical application of DNN in remote sensing imagery. Meanwhile, incapability of processing large size image patches causes cropping size too small, the resulted loss of long-range context information is detrimental to prediction accuracy.

To acquire high accuracy, conventional semantic segmen-
tation networks, such as UNet [3], FC-DenseNet [4], and DeepLabv3+ [5], usually adopt a wide and deep backbone as an encoder at the cost of large computational complexity and memory occupation. In the task of land-cover classification, limited by GPU memory capacity, most existing studies preprocessed the original remote sensing images by down sampling or cropping them into small patches less than 512x512 pixels before sending them to a deep neural network. For example, CFAMNet [6] proposes a class feature attention mechanism fused with an improved Deeplabv3+ network. To avoid memory overflow, 150 remote sensing images of 7200x6800 pixels were cropped into 20776 images of 128x128 pixels. DEANet [7] proposes a dual-branch encoder structure that depends on VGGNet [8] or ResNet [9], in their experiments, each image with a resolution of 2448x2448 pixels was compressed to half the size and then divided into subimages with a resolution of 512x512 pixels. DISNet [10] integrates the dual attention mechanism module, including the spatial attention mechanism and channel attention mechanism, into the Deeplabv3+ network. In their experiments, the original images were also cropped into small patches of 512x512 pixels before being sent into the network.

However, it takes decades of years of efforts to improve spatial resolution of remote sensing imagery, however, down sampling reverses this progress and incurs spatial detail loss. The rich details of objects, such as the geometrical shape and structural content of objects, are blurred by down sampling. 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 as less than or equal to 512x512 pixels causes a loss of long-range context information and leads to misjudgment. As shown in Figure 2, a remote sensing image with a resolution of 2048x1536 pixels is cropped into 12 small patches with a resolution of 512x512 pixels. If view the whole original image, it is clear that the water surface is a lake; however, if view 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 autonomous driving field, support of large size image input is more important for the correct semantic segmentation of remote sensing images. (In Section IV-D1 we investigate the influence of input image size on prediction accuracy, and demonstrate that the loss of long-range context information caused by small cropping size would incur misjudgement and lower accuracy). Another drawback of cropping is that cropping the images and restoring the predicted results to their original size would 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.

As presented in Figure 2, the abundant small segments, rich boundaries and small class variance in remote sensing images are all likely to cause semantic ambiguity near the boundaries and small segments. Meanwhile, areas where exist multiple land-cover categories 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 is 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, slow inference speed, incapability of processing large size image patches, and easy misjudgement on 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 Multi-branch Kernel-sharing Atrous convolution network (MKANet) and propose the Sobel Boundary Loss for efficient and accurate land-cover classification of remote sensing imagery. 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 Multi-branch Kernel-sharing Atrous convolution (MKA) module for multi-scale feature extraction.
2) For large input image size support and fast inference speed, we design a shallow semantic segmentation network (MKANet) based on MKA modules.
3) For accurate prediction on boundaries and small segments, we propose a novel boundary loss named Sobel Boundary Loss.

II. 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.

A. Lightweight Semantic Segmentation Networks

Certain semantic segmentation networks employ lightweight backbones, so compared with large networks, they consume
fewer hardware resources. For example, ENet [11] 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 [12] and BiSeNetV2 [13] propose 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) [14] that consists of two deep branches between which multiple bilateral fusions are performed. To guarantee accuracy without drastically increasing computational consumption, ABCNet [15] replaces the dot-product attention mechanism of quadratic complexity with a linear attention mechanism for global contextual information extraction. DABNet [16] adopts depthwise asymmetric convolution and dilated convolution to build a bottleneck structure for parameter reduction. DFNet [17] utilizes 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. Empirical observation shows that prediction errors are more likely to occur on boundaries and small segments [18], this observation prompts us to propose a novel boundary loss as detailed in Section III-D to bridge the accuracy gap between lightweight networks and large networks.

B. Kernel-Sharing Mechanism

The authors of KSAC [19] argued the weakness of the original ASPP [20] structure is that the kernels in the branch with small atrous rates only learn details and handle small objects well, while the kernels in the branch with large atrous rates are only able to extract features with large receptive fields. The lack of communication among branches compromises the generalizability of individual kernels. To tackle this problem, they proposed that multiple branches with different atrous rates can 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, the representation ability of the shared kernel is thus improved. KSAC adopts the architecture of DeepLabV3+, its modified ASPP structure consists 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 are various, which leads land objects of the same category have different scales. On the other hand, lands belonging to the same category have different areas. The feature of multi-scales in remote sensing images accords with the advantages of kernel-sharing mechanism, so we decide to adopt this mechanism in the basic module design.

However, direct introduction of KSAC to 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 generous images. Therefore, constructing a backbone by purely stacking the original KSAC structure is not feasible. So, we handcraft a novel multi-branch module as detailed in Section III-A the newly proposed module can be stacked in multiple stages as the backbone for semantic segmentation.

III. PROPOSED METHOD

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

A. Multi-branch 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 [4]. However, one of the costs of building a deep architecture is slow inference speed. For high efficiency and fast inference speed, we design the Multi-branch Kernel-sharing Atrous convolution (MKA) 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 a MKA module is inexpensive, 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 & pointwise convolution

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 $3 \times 3$ kernel is shared by the $M$ depthwise atrous convolutions with dilation rates of 1, 2, ... and $M$. Next, batch normalization is applied in each branch.

Compared with KSAC [19], 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 computation complexity and memory
The number of branches $M$ total parameters of the MKA module are approximately $MN$, the computational complexity of the next module.

3) Part 3: Concatenation & Pointwise Convolution: After the second part, the output features of each branch are concatenated, and then a $1 \times 1$ convolution is applied to the fused features. This part has two functions: generating new features concatenated, and then a.

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 $i$th branch ($i > 1$), a depthwise regular convolution with kernel size $(2 \times i - 1)$ is added, followed by batch normalization. Note that atrous convolution with a dilation rate of 1 is just 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) Part 3: Concatenation & Pointwise Convolution: After the second part, the output features of each branch are concatenated, and then a $1 \times 1$ 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.

4) Complexity Analysis: The number of parameters in the first part is $3 \times 3 \times N = 9N$; in the second part is $(4M^3 - M)/3 - 1) \times N$; and in the third part 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 $3 \times 3$ convolution.

B. Network Architecture

For faster inference speed and small memory occupation, based on MKA modules, we design a lightweight semantic segmentation network named MKANet, as illustrated in Figure 3. 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 (CAM) [21] 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 IV-B, this shallow but effective architecture makes MKANet capable of supporting 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 faster.

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

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

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

TABLE I

| Encoder Design. |
|------------------|
| Stages | Operation | Output Channels |
|--------|-----------|-----------------|
| Input Image | ConvS2 | $c/2$ |
| Stage 1 | ConvS2 | $c$ |
| Stage 2 | MKA | $r$ |
| Stage 3 | ConvS2 | $c \times 2$ |
| Stage 4 | ConvS2 | $c \times 4$ |
| Stage 5 | MKA | $r \times 8$ |

For faster inference speed and small memory occupation, based on MKA modules, we design 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 (CAM) [21] 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 IV-B, this shallow but effective architecture makes MKANet capable of supporting 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 faster.

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

Each stage begins with a $3 \times 3$ convolution of stride 2, followed by batch normalization and ReLU activation. MKA modules are then repeated $r$ times in each stage since 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 ($c = 128, r = 1$).

2) Decoder: The purpose of the first two stages is to extract simple, low-level features and to quickly downsize the resolution for computation reduction. 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 [12] and STDC [22], use squeeze-and-excitation (SE) attention [23] 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.
Fig. 4. Architecture of MKANet. The blue cuboid represents the $3 \times 3$ convolution with stride 2, and the green cuboid represents $3 \times 3$ convolution with stride 2 plus one or more MKA modules.

Fig. 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.

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 Modules (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 were 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, up sampled by 2X and 4X, 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 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.

C. 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 up sampled by 8X to restore them to the same resolution as the input image. The up sampled class logits are compared with the ground truth by the main semantic segmentation loss function.

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 up sampled 8X, 16X, and 32X before being sent to three auxiliary semantic segmentation loss functions and three Boundary Loss functions.

D. Sobel Boundary Loss
Sobel operator is a discrete differentiation gradient-based operator that computes the gradient approximation of 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 auxiliary boundary loss. The procedure is illustrated in Figure 7 and detailed in Algorithm 1.

For a segment with 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 building 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 larger segments to smaller 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.

\[ d \] 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 generous images in which objects have clear contours, the land boundaries in land-cover satellite images are comparatively vague. Secondly, setting buffer zones of width \( d \) benefit the network learning how to discriminate different categories. Thirdly, if \( d \) is set too small, the samples participating in 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 the 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 as 50 pixels in the illustration figures and experiments.

E. 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 \( L_b \):

### Algorithm 1 Sobel Boundary Target Generation.
**Input:** Ground Truth \( Y \), Sobel Operator \( S_x, S_y \), Dilation Rate \( d \).
**Output:** Sobel Boundary Target \( \hat{Y} \).

\[
X_b \leftarrow (|\text{Conv}(Y, S_x)| + |\text{Conv}(Y, S_y)|) > 0
\]

\[
X_d \leftarrow \text{Dilate}(X_b)
\]

\[
\hat{Y} \leftarrow Y \odot X_d
\]

return \( \hat{Y} \)

\( \odot \) means elementwise multiplication.
Fig. 8. Images, ground truths and boundary targets generated by Sobel kernels and dilation operation with \( d = 50 \) pixels.

![Image showing images, ground truths, and boundary targets generated by Sobel kernels and dilation operation.](image)

\[ L_t = w_1 \times L_{mm} + w_2 \times L_{a} + w_3 \times L_b \]  

(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 as 1 in the following experiments to avoid any hyperparameter tuning trick.

IV. EXPERIMENTAL RESULTS

Since the MKA module specializes in multi-scale 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 multi-scale remote sensing images. Then we measured inference speeds of MKANet at various image sizes to validate whether its architecture design can boost inference speed and support large image size. 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.

A. Image Classification

We add an image classification head as illustrated in Figure 9 on top of MKANet encoder for the task of image classification, and name it MKANet-Class. We compared MKANet-Class with other state-of-the-art lightweight classification networks on the RSCCN7 [24] scene classification dataset of remote sensing images (Figure 10). RSCCN7 contains seven typical scene categories of image size 400x400 pixels. For each category, 400 images are sampled on four different scales with 100 images per scale. A total of 2800 images were resized to 384x384 pixels and split into a training set, validation set, and test set at a ratio of 2:1:1.

Training details: Cross-entropy was selected as the loss function, and AdamW [25] 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 II, MKANet-Class Small outperforms other state-of-the-art lightweight classification networks with better accuracy and significantly faster inference speed, which TABLE II

| Method                  | Accuracy  | FPS   |
|------------------------|-----------|-------|
| MobileNetV2            | 90.43%    | 130.2 |
| MobileNetV3            | 90.43%    | 107.9 |
| EfficientNet-B0        | 91.28%    | 80.1  |
| ShuffleNetV2 x1.0      | 91.85%    | 123.5 |
| ResNet18               | 91.71%    | 178.8 |
| ResNet34               | 92.14%    | 97.7  |
| STDC1                  | 92.43%    | 143.8 |
| MKANet-Class Small     | 92.85%    | 314.3 |

Inference speed frames per second (FPS) were measured on a computer with CPU AMD 4800HS, GPU NVIDIA RTX 2060 Max-Q 6G, and a Pytorch environment.
The GID Fine Land Cover Classification dataset consists of RGB satellite images up to 7200×7200 pixels and is approximately 2X faster than other lightweight networks and more than 13X faster than large networks can. MKANet Small is even capable of processing images 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 [32].

Training details: For all the lightweight networks, cross-entropy was selected as the loss function, and AdamW [25] was selected as the optimizer with a batch size of 6, 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 that are defined as:

\[
\text{MIoU} = \frac{1}{N} \sum_{c=1}^{N} \frac{TP_c}{TP_c + FP_c + FN_c}.
\]

\[
\text{MF1} = \frac{1}{N} \sum_{c=1}^{N} \frac{2 \times TP_c \times TP_c + FP_c \times TP_c + FN_c}{TP_c + FP_c + TP_c + FN_c}.
\]

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 \( c \).

1) DeepGlobe Land Cover Dataset Experimental Results: The DeepGlobe Land Cover Classification dataset provides high-resolution sub-meter 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 [33] and the Zeebruges dataset [34]. The image size of this dataset is so high that only lightweight networks are able to...
TABLE IV
COMPARISON OF MKANETS WITH OTHER STATE-OF-THE-ART EFFICIENT NETWORKS ON THE DEEPGLOBE LAND COVER DATASET.

| Method                  | Proportion (%) | Per-category IoU (%) | MIoU (%) | Speed (FPS) |
|-------------------------|---------------|----------------------|---------|-------------|
|                         | Urban | Agricult | Range | Forest | Water | Barren |         |            |
| ENet [11]               | 71.14  | 80.86   | 0.42  | 75.44  | 73.10 | 49.41  | 58.39 | 6.96       |
| ABCNet [15]             | 73.52  | 82.03   | 29.03 | 75.95  | 72.50 | 55.94  | 64.83 | 7.15       |
| BiSeNetV1 (Resnet18) [12] | 72.68  | 83.91   | 28.09 | 78.97  | 74.71 | 56.71  | 65.85 | 8.41       |
| BiSeNetV2 [13]          | 72.46  | 82.23   | 23.52 | 77.86  | 69.37 | 51.03  | 62.75 | 8.42       |
| STDC1 [22]              | 74.68  | 85.05   | 31.75 | 76.82  | 75.34 | 59.11  | 67.12 | 11.81      |
| STDC2 [22]              | 73.48  | 83.79   | 30.57 | 76.27  | 73.48 | 58.11  | 65.95 | 8.76       |
| DDRNet23 [14]           | 75.01  | 84.53   | 32.62 | 77.81  | 72.79 | 58.50  | 67.63 | 7.21       |
| DABNet [16]             | 74.44  | 84.69   | 33.62 | 78.81  | 73.48 | 58.50  | 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 CPU INTEL i5-3470, GPU NVIDIA RTX 3060 12G, and a Pytorch environment.

TABLE V
COMPARISON OF MKANETS WITH LARGE NETWORKS ON THE DEEPGLOBE LAND COVER DATASET.

| Method                  | MIoU(%) | MF1(%) |
|-------------------------|---------|--------|
| UNet (Res2Net50) [3]    | 67.57   | 79.50  |
| PSPNet (Res2Net50) [30] | 69.45   | 81.07  |
| DeepLabV3 (ResNet50) [20] | 68.39 | 80.27  |
| DeepLabV3 (ResNet101) [20] | 69.94 | 80.55  |
| DeepLabV3+ (Res2Net50) [5] | 69.12  | 80.85  |
| DeepLabV3+ (ResNet101) [5] | 69.39  | 81.06  |
| EncNet (Res2Net50) [35] | 68.53  | 80.42  |
| EncNet (ResNet101) [35] | 68.60  | 80.40  |
| PSANet (ResNet50) [36]  | 68.27  | 80.13  |
| GCNet (ResNet50) [37]   | 69.09  | 80.47  |
| DEANet [7]              | 71.80  | 82.60  |
| MKANet Small            | 70.68  | 81.09  |
| MKANet Base             | 71.78  | 82.42  |
| MKANet Large            | **72.00** | **82.62** |

All the large networks employed the backbones pre-trained on ImageNet [38]. For all the large networks, the experimental results reported by Wei et al [7] were quoted.

As shown in Table IV and Table V MKANets lead other competitive lightweight networks by at least 3% and even surpass the large networks with pretrained backbones. In a previous study [7], 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 benefitted by 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.

As shown 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 accords with the per-category IoUs in Table IV. MKANets lead other networks by a large margin in the categories of small segments, such as water and range.

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 three categories are as scarce as less than 1%. We selected this rich category dataset to assess the discrimination ability of MKANet on
### TABLE VI
Comparison of MKANets with other state-of-the-art, efficient networks on the GID dataset.

| Method           | Indust Urban Rural | Traff. | Paddy | Irrig. | Crop | Gard. | Arbor | Shrub | Natur. | Artif. | River | Lake | Pond |
|------------------|--------------------|--------|-------|-------|------|-------|-------|-------|--------|--------|-------|------|------|
| Proportion (%)   | 7.26               | 13.96  | 5.59  | 4.74  | 4.38 | 36.39 | 2.86  | 0.82  | 9.04   | 0.3    | 1.68  | 0.83 | 3.05 |
| MKANet Large     | 68.82              | 71.39  | 56.83 | 71.92 | 76.18 | 83.43 | 35.57 | 13.40 | 96.00  | 12.49  | 62.28 | 86.70 | 74.46 |
| MKANet Base      | 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 |
| **Per-category IoU (%)** | | | | | | | | | | | | |
| **MKANet Large** | **65.40**          | **61.59** | **52.48** | **63.72** | **80.88** | **93.22** | | | | | | | |
| **MKANet Base**  | **64.26**          | **56.83** | **52.48** | **63.72** | **78.90** | **93.27** | | | | | | | |
| **MKANet Small** | **62.89**          | **56.83** | **52.48** | **63.72** | **78.90** | **93.27** | | | | | | | |
| **Methods**      | **60.00**          | **72.88** | **56.83** | **71.92** | **83.43** | **35.57** | 13.40 | 96.00  | 12.49  | 62.28 | 86.70 | 74.46 | 73.49 |
| **MKANet Large** | **65.40**          | **61.59** | **52.48** | **63.72** | **80.88** | **93.22** | | | | | | | |
| **MKANet Base**  | **64.26**          | **56.83** | **52.48** | **63.72** | **78.90** | **93.27** | | | | | | | |
| **MKANet Small** | **62.89**          | **56.83** | **52.48** | **63.72** | **78.90** | **93.27** | | | | | | | |
| **Methods**      | **60.00**          | **72.88** | **56.83** | **71.92** | **83.43** | **35.57** | 13.40 | 96.00  | 12.49  | 62.28 | 86.70 | 74.46 | 73.49 |

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

### TABLE VII
Comparison of MKANets with large networks on the GID dataset.

| Method           | Indust Urban Rural | Traff. | Paddy | Irrig. | Crop | Gard. | Arbor | Shrub | Natur. | Artif. | River | Lake | Pond |
|------------------|--------------------|--------|-------|-------|------|-------|-------|-------|--------|--------|-------|------|------|
| Proportion (%)   | 7.26               | 13.96  | 5.59  | 4.74  | 4.38 | 36.39 | 2.86  | 0.82  | 9.04   | 0.3    | 1.68  | 0.83 | 3.05 |
| **Per-category F1 Score (%)** | | | | | | | | | | | | | |
| **MKANet Large** | **68.77**          | **74.81** | **65.62** | **50.43** | **59.66** | **83.79** | **24.44** | **11.72** | **95.48** | **4.15** | **24.34** | **87.79** | **10.36** |
| **MKANet Base**  | **67.52**          | **71.08** | **59.64** | **52.48** | **63.72** | **80.88** | **93.27** | | | | | | |
| **MKANet Small** | **66.74**          | **70.35** | **56.83** | **71.92** | **83.43** | **35.57** | 13.40 | 96.00  | 12.49  | 62.28 | 86.70 | 74.46 | 73.49 |

For all the large networks, the experimental results reported by Ding et al. [32] were quoted.

fine and similar land-cover categories, and also evaluate its robustness with regard to highly unbalanced remote sensing imagery.

As shown in Table VI and Table VII MKANets outperform all other lightweight networks and the large networks by a large margin.

The per-category IoUs in Table VI indicate that the superiority of MKANets is mainly manifested in minor categories, small dimensional categories, and hard-discriminating categories. In these categories, MKANet Small surpasses the average IoU of other lightweight networks by more than 18%. For example, the large category of farmland (consists of paddy land, irrigated land and dry cropland) is highly skewed in distribution; the samples of irrigated land are approximately 10 times greater than those of paddy land and dry cropland; and MKANet Small exceeds the average IoU of other efficient networks by 37.7% on paddy land and 25.1% on dry cropland.

As shown in Figure 12 small-dimensional land covers, such as rivers, streets, and rows of rural houses, are better classified by MKANets. MKANet Small exceeds 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 segments discrimination and spatial detail reconstruction.
### D. Ablation analysis

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

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 VIII, 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 incur misjudgement.

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 IX compared with regular atrous convolutions, kernel-sharing atrous convolutions had stronger feature extraction ability.

3) Effectiveness of Coordinate Attention Module: To assess the effect of 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 X 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) 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 XI 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 XII). The IoUs of most categories were boosted by the two kinds of auxiliary losses, indicating that these auxiliary heads can promote spatial detail learning at lower levels and semantic context learning at higher levels.

To visualize the effect of auxiliary semantic segmentation loss and auxiliary boundary loss, the predicted labels of the above three networks are displayed in Figure XIII. 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.

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 XIII, the performance of MKANet increased by stacking additional MKA modules in each stage.

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 XIV, 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...
TABLE XI
COMPARISON OF MKANETS WITH AND WITHOUT AUXILIARY LOSSES ON THE DEEPGLOBE LAND COVER DATASET.

| Method          | Urban | Agricult | Range | Forest | Water | Barren | MIoU (%) |
|-----------------|-------|----------|-------|--------|-------|--------|----------|
| MKANet Large    | 76.53 | 87.30    | 40.36 | 81.66  | 65.08 | 72.00  |
| MKANet-NB Large | 75.71 | 87.25    | 35.51 | 81.83  | 62.19 | 70.50  |
| MKANet-NA Large | 75.97 | 86.48    | 34.22 | 80.14  | 60.99 | 69.55  |

TABLE XII
COMPARISON OF MKANETS WITH AND WITHOUT AUXILIARY LOSSES ON THE GID FINE LAND COVER CLASSIFICATION DATASET.

| Method          | Indust | Urban | Rural | Traff. | Paddy | Irrig. | Crop | Gard. | Arbor | Shrub | Natur. | Artif. | River | Lake | Pond | MIoU (%) |
|-----------------|--------|-------|-------|--------|-------|--------|------|------|-------|-------|--------|--------|-------|------|------|----------|
| MKANet Large    | 67.90  | 70.35 | 60.00 | 72.88  | 73.51 | 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  | 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 | 65.88 | 63.51 |

TABLE XIII
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 CPU INTEL i5-3470, GPU NVIDIA RTX 3060 12G, and a Pytorch environment.

TABLE XIV
COMPARISON OF MKANETS WITH A VARIABLE NUMBER OF PARALLEL BRANCHES ON THE DEEPGLOBE LAND COVER DATASET.

| Method          | Urban | Agricult | Range | Forest | Water | Barren | MIoU (%) |
|-----------------|-------|----------|-------|--------|-------|--------|----------|
| 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    |

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

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 can support even larger image size of 7200×7200 pixels on RTX 3060 and image size up to 10k×10k pixels on RTX3090.

V. CONCLUSION

Aim at the characteristics of multi-scales and large image size of top view remote sensing imagery, we merged the specialized multi-branch module and the shallow architecture design into MKANet. Through extensive experiments and ablation analysis, MKANet has reached our initial expectation and alleviated the three problems (slow inference speed, incapability of processing large size image patches, and easy misjudgement 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 can support even larger image size of 7200×7200 pixels on RTX 3060 and image size up to 10k×10k pixels on RTX3090.
Its friendly support of large subimages greatly alleviates spatial detail loss due to down sampling or long-range context information loss due to cropping. Meanwhile, MKANet Small is approximately 2X faster than other lightweight networks and 13X faster than the large networks. Both merits highlight the value of MKANet in 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 operators convolution and dilation operation were innovatively utilized to capture category impurity areas, exploit boundary information and exert an extra penalty on boundaries and small segment misjudgment. Both quantitative metrics and visual interpretations have verified that the Sobel Boundary Loss can promote spatial detail learning and boundary reconstruction.

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

In future research, we will extend the application of the MKA module as an effective and swift feature extractor in instance segmentation, change detection, and object detection tasks.

**ACKNOWLEDGMENTS**

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).

**Zhiqi Zhang** received a B.Sc. degree in Geographic Information Systems from Huazhong Agricultural University, a B.Eng. degree in Computer Science and Technology from Huazhong University of Science and Technology, an M.Eng. degree in Computer Technology from Wuhan University, and a Ph.D. degree in photogrammetry and remote sensing from Wuhan University in 2006, 2006, 2015, and 2018. He is currently an Associate Professor with the School of Computer Science, Hubei University of Technology. His research interests include system architecture, algorithm optimization, AI, and high-performance processing of remote sensing.

**Wen Lu** is a postgraduate student majoring in Computer Science and Technology in the School of Computer Science, Hubei University of Technology. His research interests include computer vision, remote sensing, machine learning, and deep learning.

**Jinshan Cao** received a Ph.D. degree in photogrammetry and remote sensing from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2012. He is currently an Associate Professor with the School of Computer Science, Hubei University of Technology, Wuhan, China. His research interests include geometric calibration, sensor orientation, and image registration of high-resolution satellite imagery.

**Guangqi Xie** received an M.Sc. degree in Remote Sensing of Resources and Environment from China University of Geosciences, Wuhan, China, in 2018. He is currently working toward a Ph.D. degree in the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China. His research interests include image matching and registration, pansharpening, and image superresolution.
