Multiscale Feature Reconstruction and Interclass Attention Weighting for Land Cover Classification

Zongqian Zhan, Zirou Xiong, Xin Huang, Chun Yang, Yi Liu, and Xin Wang

Abstract—**Land cover classification has the goal to attribute each pixel of high-resolution remoste sensing image with planimetric category labels (such as vegetation, building, and water). In recent years, many serial deep-learning architectures (features are delivered through a single path, such as in *ResNet*, *MobileNet*, and *Segformer*) based on convolutional neural networks and attention mechanisms have been widely explored in land cover classification. However, high-resolution remote sensing images typically have abundant textual details, variable scales in objects, large intraclass variance, and similar interclass correlation, which bring challenges to land cover classification. In this work, we present two pluggable modules to further boost serial learning architecture: first, to cope with ambiguous boundaries caused by lost details and fragmented segmentation stemmed from scale variances, a combination of spatial attention and channel attention is proposed for multiscale feature reconstruction (MSFR); second, to mitigate the classification error caused by intraclass variance and interclass correlation, we explore an interclass attention weighting (ICAW) module, which builds feature vectors for each category, and applies a multihead attention model to capture self-attention dependence among different categories. The experimental results demonstrate that the proposed modules are feasible to the existing serial learning architectures and can improve overall accuracy (OA) by 5.64% on the ISPRS Vaihingen two-dimensional dataset (using *ResNet50* as a backbone); in particular, the OA values are 80.68% and 86.32% before and after using the proposed modules, respectively. In addition, compared with other state-of-the-art models, our method can achieve similar or even better classification results, yet offer superior inference performance.**

Index Terms—**Interclass attention, land cover classification, multiscale feature reconstruction (MSFR), remote sensing image, semantic segmentation.**

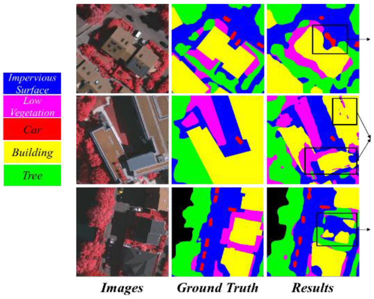

Impervious Surface
Low Vegetation
Car
Building
Tree
a. Ambiguous boundaries caused by lost details
b. Fragmented segmentation caused by scale variance
c. Classification error caused by intra-class variance and inter-class correlation
Images
Ground Truth
Results

Fig. 1. Qualitative illustration of several limitations in serial learning networks for land cover classification (this result is generated by using backbone of ResNet50 on ISPRS Vaihingen dataset).

I. Introduction

As a significant component of remote sensing image interpretation, land cover classification aims to parse each pixel into meaningful category labels [1], [2], [3]. In the early times, land cover classification was explored by index-based methods, such as normalized vegetation index (normalized difference vegetation index) [4] and built-up area presence index *PanTex* [5]. Later, traditional machine learning algorithms, such as wavelet transform [6], superpixels [7], [8], support vector machine [9], random forest [10], [11], [12], and Mahalanobis distance [13], were widely studied and applied for substantial classification performance in terms of robustness and overall accuracy [14], [15]. Over the last years, remote sensing images can be easily obtained by various sensors equipped in satellites, aircraft, and unmanned aircraft vehicles [16], [17], [18], and these images have been dramatically improved in terms of resolution and the ability of observing larger areas, which pose challenges for the mentioned traditional machine learning algorithms to deal with remote sensing images with rich details, contextual textures, variable intraclass differences, and similar interclass features [19]. On the other hand, convolutional neural networks (CNNs) and multilayer perceptron (MLP) have shown superiority on land cover classification for high-resolution remote sensing images [20], [21], [22]. And thanks to the remarkable achievements of semantic segmentation methods in computer vision, variants of fully convolutional neural networks (FCNs) based on encoder–decoder architectures have become mainstream for land cover classification [24], [25], [26].

Ample CNN variants have been studied to solve the task of land cover classification by taking high-resolution remote sensing images as input, such as these serial networks of the authors in [27], [28], [29], and [30]. However, there are still several issues that remain to be addressed, one of them is fragmented segmentation caused by scale variance of objects in high-resolution remote sensing images, as shown in Fig. 1(b). In CNN-based methods, the feature maps from the high-resolution layers typically have good focus on the tiny objects but observe

Manuscript received 26 August 2023; revised 4 November 2023 and 8 December 2023; accepted 8 December 2023. Date of publication 13 December 2023; date of current version 28 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61871295 and Grant 42301507, and in part by the Natural Science Foundation of Hubei Province, China under Grant 2022CFB727. *(Corresponding author: Xin Wang.)*

Zongqian Zhan, Zirou Xiong, Xin Huang, Yi Liu, and Xin Wang are with the School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China (e-mail: zqzhan@sgg.whu.edu.cn; xzirou@whu.edu.cn; xhuang1@whu.edu.cn; yliu@sgg.whu.edu.cn; xwang@sgg.whu.edu.cn).

Chun Yang is with the Hangzhou SensingX Technology Company Ltd., Hangzhou 310000, China (e-mail: chun.yang@sensingx.cn).

The code is available at: https://github.com/StraySparks/MSFR-and-ICAW.

Digital Object Identifier 10.1109/JSTARS.2023.3342453

large objects partly because of the smaller receptive fields, while low-resolution feature maps with larger receptive fields ignore details and have difficulties to identify adjacent objects in tiny sizes [19]. Another limitation by CNN methods is the lost details caused by downsampling operations in the encoding stages [2], [3], which negatively affects the classification of small objects as well as edges [32] [shown by Fig. 1(a)]. In addition, the intraclass differences of the same objects [36] and the interclass correlation between different objects are not fully explored by most existing deep-learning-based methods, leading to erroneous classification on corresponding objects. For example, the phenomenon of "same object with different spectra" and "same spectrum with different objects" causes serious ambiguous classification between *impervious surface* and *building* in Fig. 1(c). To reduce erroneous classification caused by intraclass differences, Yuan et al. [36] propose the object-contextual representations (OCRs) model; however, the issue caused by interclass correlations remains unsolved.

To cope with the mentioned limitations, based on serial learning networks (*ResNet* [38] for example) whose features are consecutively encoded from the high-resolution feature maps to the low-resolution feature maps through a single path, we proposed two corresponding pluggable modules: *first*, to take care of both multiscale variance and lost details, we propose a multiscale feature reconstruction (MSFR) module consisting of two models. More specifically, our previous position-sensitive attention (PSA) [3] model is run on the highest resolution feature maps, which are more sensitive to local edges; then, a new multiscale channel attention (MCA) is further applied to reconstruct multiscale feature representation for the lower resolution features. Second, to cope with erroneous classification caused by intraclass differences and interclass correlations, we present an interclass attention weighting (ICAW) module; it first generates feature vectors for each category and applies the multihead attention mechanism from *transformer* [39], [40] to investigate a weighted fusion of interclass distances based on per-class feature vectors. In summary, the contributions of this work are given as follows.

1) We present two pluggable modules for improving serial learning architecture on land cover classification.
2) A simple yet effective MSFR architecture is proposed to take care of multiscale variance and lost details.
3) Inspired by the OCR model, an ICAW architecture is designed to generate category-level features for improving land cover classification.

The rest of this article is organized as follows. Related works are reviewed in Section II. The details of our methods are illustrated in Section III. The performance of our works on different datasets and the corresponding ablation experiments are reported in Section IV. Finally, Section V concludes this article.

II. Related Work

In this section, we briefly review some studies for land cover classification, which are relevant to our works in terms of multiscale feature learning, recovering lost details, and segmentation considering regional context.

A. *Multiscale Feature Learning*

The scale variance of planimetric features is very common in high-resolution remote sensing images and brings difficulty for segmentation models to estimate features appropriately [19]. One frequently used solution is to extract multiscale feature, which perform elementwise addition or channelwise concatenation on the feature maps generated by each layer from encoder [1], [2], [24], [31]. Multiscale feature fusion based on channelwise concatenation and elementwise addition is an idiomatic approach to enhance feature stability against scale variance [24], [31] in which channelwise concatenation can preserve multiscale features and perform convolution aggregation via training, while elementwise addition can effectively reduce the cost of parameters and computations. However, the insufficient semantic information of large objects at high-resolution feature maps may lead to ambiguous prediction.

Different from multiscale feature fusion using a direct skip connection between encoder and decoder, parallel pyramid architectures expand the receptive field and fuse multiscale contextual information systematically. For example, pyramid scene parsing net obtains multiple receptive fields by convolutional kernels of different sizes [41], and atrous spatial pyramid pooling applies different dilation rates [20]. In addition, Zheng et al. [19] apply asymmetric receptive fields to build the large kernel pyramid pooling, which is able to capture multiscale correlation between long-range features [42], avoiding the large number of parameters caused by large kernel convolution and "gridding issue" caused by dilated convolutions.

Thanks to the fact that the transformer is able to capture long-range dependencies, Swin transformer (SwinT) [43] uses the shifted window strategy to estimate the self-attention within nonoverlapping local windows and dependency connections across windows from different scales. Taking the superiority of transformer, *Segformer* [44] applies a hybrid architecture to improve the representation of multiscale features; *UNetformer* [45] improves the conventional *UNet* by integrating global–local transformer block and weighted sum in the decoder part; and Gao et al. [46] use self-attention to adaptively fusion features extracted from various scales. Instead of improving the network structure, Tao et al. [47] feed images with various resolutions into a constant backbone to obtain multiscale features and perform hierarchical multiscale attention for feature fusion. Although the hierarchical multiscale attention can adaptively integrate dominant features at each scale, the reusing of backbone leads to a relatively high computational cost.

In this work, we design a new architecture to reconstruct multiscale features for serial learning networks in which the attention mechanism is modified on different channels with an effective model.

B. *Details Refinement*

The efficacy of CNN methods in integrating local texture features has been proved [20]. However, in general, these networks become less sensitive to geometric features when pooling and convolution operations impede the integrity of boundary information, resulting in ambiguous classification of adjacent categories [19]. To recover lost details, one strategy is to use dilated convolutions [32] or convolutions with a stride of 2 so as to avoid pooling operations. Other solutions are to reintroduce some details [31], [34] or revise the edges directly from the classification results [32], [35].

In order to refine details after downsampling, *UNet* [31] uses skip connections to reintroduce details from encoder, while *SegNet* [25] tends to reconstruct details with the benefit of deconvolutions. In addition, based on parallel learning architecture, *HRNet* [48] exchanges features continuously on multiresolution layers to preserve details. Some works try to refine lost details via postprocessing; the conditional random field model [35] calculates the relationship among pixels and formulates the best classification for boundaries by posterior probability; adaptive affinity fields [17], [49] build a collection of pixel-centric relations and learns the optimal boundary structures adaptively; *SegFix* [50] constructs a distance map and a direction map to improve segmentation in boundary areas. Another direction for refining details is to combine subnetworks of edge detection as multitask learning, for example, Yu et al. [51] apply the same encoder and different decoders to evaluate the loss of edge detection and semantic segmentation simultaneously. *ScasNet* [52] cascades edge detection branches to correct edge fitting residuals. Zheng et al. [19] deploy image-level dice loss to improve the sensitivity of the network to the geometric boundaries of objects. Ding et al. [53] employ transformer and propose a wide-context network for extracting features from both local and global image levels, which can benefit prediction with more details.

In general, the strategies of reintroducing details also deliver ambiguous features [42], postprocessing methods rely on coarse segmentation results, and subnetworks of edge detection increase the computation cost and require ground truth of boundary for loss evaluation. Based on our previous work on PSA model [3], we incorporate PSA model into high-resolution details refinement as a part of our MSFR architecture, which deals with the multiscale feature and lost details at the same time.

C. *Regionwise Segmentation*

Contextual information around the local region where each pixel is located is vital for land cover classification. Caesar et al. [54] propose a free-form pooling layer for region-of-interest (RoI), which in particular takes the representations of the effective pixels in the region into account. Based on the discriminative regions learned by CNNs on the task of image classification, Wei et al. [55] improve an adversarial erasing approach for regionwise segmentation progressively. Based on the predetected RoI, Ye et al. [56] propose a semantic relation learning module, which has two paths incorporated with bottleneck structures and skip connections. In order to improve the efficiency of regional representation aggregated by regular receptive fields, Fu et al. [57] propose dual attention to weight regional representations, and Dai et al. [37] apply the deformable convolution to construct adaptive receptive field that is consistent with objects inside the region. Tang et al. [22] propose a hybrid solution via combining the traditional segmentation method and MLP, which clusters similar pixels into superpixels and identifies them with a classifier trained via MLP. Moreover, Yuan et al. [36] propose the OCR model as well as Zhang et al. [58] propose attentional class feature network (ACFNet) to generate per-class feature vectors instead of fragmented regional features, which is supposed to address the issue caused by intraclass differences. However, the problem caused by interclass correlations remains unsolved.

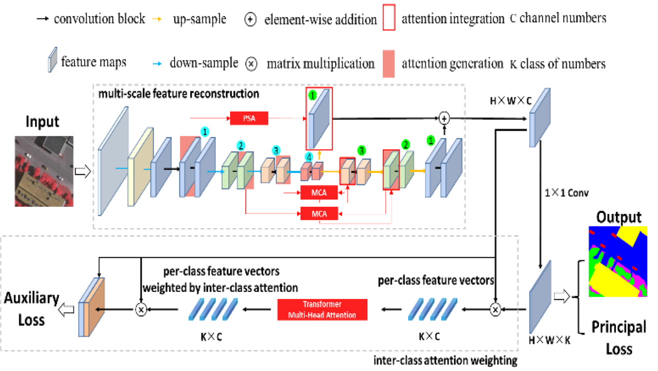


Fig. 2. Overall architecture of the proposed model. PSA means our spatial attention, MCA is the channel attention, numbers 1–4 in blue circles represent different resolution layers from encoder and numbers in green circles denote different resolution layers from decoder. H denotes the height, W denotes the width, C denotes the number of channels, and K denotes the number of categories.

In this article, we improve the per-class feature vector generation in the same way as in the OCR model and ACFNet with ICAW, which is expected to reduce erroneous classification caused by interclass correlations.

III. Methodology

In this section, we illustrate our framework for land cover classification in detail to address the problems of object scale variance, lost details, ambiguous classification caused by intraclass differences, and interclass correlation. In particular, two pluggable modules composed of the MSFR architecture and the ICAW architecture will be explained.

A. *Overview of Our Architecture*

The workflow of our method is shown in Fig. 2. It mainly consists of two parts, in which the first part is for MSFR and the second part is to perform ICAW. The MSFR architecture takes an image as input and outputs a quarter-sized feature map, which will act as the input for the ICAW architecture. MSFR can be deployed in different serial encoder and decoder structures in which we generate the PSA with the feature maps from encoder at the highest resolution (number 1 in blue circle) [3] and the MCA at the rest two resolutions (numbers 2 and 3 in blue circles). Based on PSA and MCA, we weigh the corresponding feature maps from the decoder to provide multiscale information with expectation of recovering more details. ICAW generates feature vectors for each category and applies the multihead attention mechanism [39] for weighting interclass correlation. Analogous to the OCR model [36], ICAW is only activated when network is training, providing an auxiliary loss. With the help of ICAW,

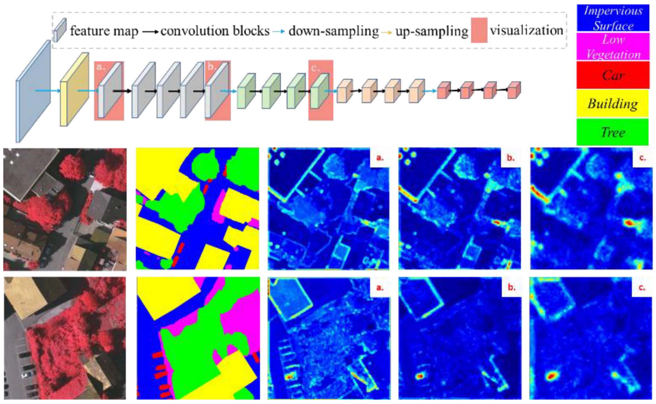


Fig. 3. Example for feature map visualizations in ResNet50. The input images contain NIR, red, and green bands, and we use pink, red, green, yellow, and blue to represent the corresponding ground truth categories. a, b, and c denote the visual feature maps extracted from different layers, and we upsample them to the same size as the input image (i.e., 512 × 512). The visualization strategy is to build a pseudocolor map with the corresponding response peaks on the channel dimension for each pixel, and the color is warmer where the response is higher.

guidance for object-level feature learning is supposed to be provided. In the following sections, more details of both modules are introduced.

B. Multiscale Feature Reconstruction

In order to mitigate the poor impact of object scale variance, we propose the MSFR architecture, whose detail is illustrated in this section.

1) Qualitative Analysis of CNN Feature Maps With Various Resolutions: First of all, we investigate the architecture of serial learning networks and explore the extraction of feature maps (*ResNet50* for example). As shown in Fig. 3, high-resolution (one-quarter of the original image size) feature maps are always extracted from the starting layers and then downsampled and convolved, resulting in lower resolution feature maps. In other words, features are always delivered from the high-resolution feature maps to the low-resolution feature maps, and it is supposed to be able to find multiscale features on various resolution feature maps, which provides the possibility to gradually reconstruct multiscale features. Meanwhile, details are gradually lost during downsampling, and this means that the highest resolution feature maps contain more abundant details (as shown in Fig. 3, the corresponding response peaks on channel dimension are visualized). Therefore, lost details can also be refined in the MSFR architecture as a part of the highest resolution features.

In Fig. 3, feature map b shows both clear boundaries and low detail redundancy inside objects, which might be the best candidate for generating PSA. Similar visualization results of feature maps can be found in other serial learning networks as well.

2) Position-Sensitive Attention: To refine spatial details, we apply the PSA module [3] to generate spatial attention, which contributes to the reconstruction of the highest spatial resolution features (feature map b in Fig. 3). Two 3 × 3 convolutions are used to aggregate local context without changing the number of channels, and then the 1 × 1 convolution is deployed to compress the middle result into one channel. As shown in Fig. 4, the Sigmoid activation function is inserted in the end to normalize the values within the range of [0, 1]. With this structure, we are able to assign a weight to each spatial element. According to the visualization in Fig. 3, the highest spatial resolution features contain abundant details around boundaries; thus, the PSA model generated by these features is expected to boost the representations near the boundary areas.

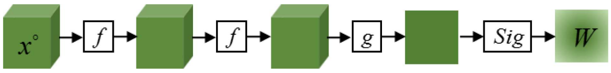


Fig. 4. Workflow of attention generation for PSA module, where x_1° presents the feature map, f is 3 × 3 convolution, g is 1 × 1 convolution, Sig means Sigmoid activation function, and W is the attention.

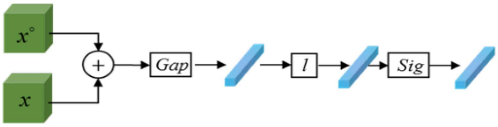


Fig. 5. Workflow of attention generation for the MCA module. x° and x present the corresponding feature maps from encoder and decoder, while Gap denotes the 3 × 3 convolution, l is the 1 × 1 convolution, Sig denotes the Sigmoid activation function, and $\oplus$ is the elementwise addition.

We can formulate the PSA module as follows:

$$\text{PSA} = \varphi_p \ (x_1^\circ) = \text{Sigmoid}\,(x_1^\circ \otimes f \otimes f \otimes g) \quad (1)$$

where x_1° represents the feature map for attention generation, while f denotes the 3 × 3 convolution, g is the 1 × 1 convolution, Sigmoid means the Sigmoid activation function, φ_p denotes the generation for PSA, and $\otimes$ denotes the convolution operation.

3) Multiscale Channel Attention: To obtain an effective integration of dominant features from various scales, we propose the MCA module to estimate two-channel attentions, improving the low-resolution features integration. For two lower spatial resolution maps, we add two corresponding feature maps from encoder and decoder, and then global pooling contributes to the global context for each dimension together with linear transformation. Finally, the Sigmoid activation function is applied for ranging the value between 0 and 1, as shown in Fig. 5. Similar to the SE attention [59], the MCA module explores more expressive feature dimensions. By applying the MCA module, we assign a weight to each dimension for selecting more representative features globally.

The MCA module is formulated as follows:

$$\text{MCA} = \varphi_m \ (x, x^\circ) = \text{Sigmoid}\,(\text{Gap}\,(x + x^\circ) \otimes l) \quad (2)$$

where φ_m denotes the attention generation for MCA, $\otimes$ is the convolution operation, and $+$ is the elementwise addition.

4) Attention Integration: As shown in Fig. 6, we integrate the above two kinds of attention into the corresponding feature maps from decoder. Specifically, the integration for the MCA module is performed by progressively upsampling, and for the PSA module, we upsample the lowest resolution feature map (generated by encoder) directly to the highest resolution for

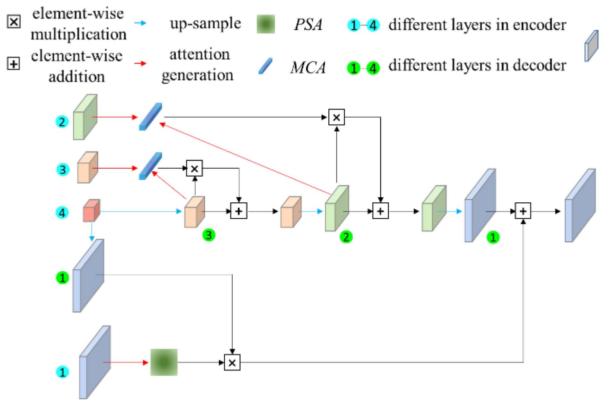

element-wise multiplication
element-wise addition
up-sample
attention generation
PSA
MCA
1-4 different layers in encoder
1 4 different layers in decoder
feature maps
2
3
4
1
1
3
2
1

Fig. 6. Workflow of attention integration for the PSA module and the MCA module, whereby numbers in blue circles and green circles mean different resolution layers from encoder and decoder, the same as Fig. 2.

integration. During each integration, we expand the attention to the same size as the corresponding feature map and then perform the elementwise multiplication between them to get a weighted feature map. Eventually, we apply the elementwise addition between the original feature map and the weighted feature map, enhancing multiscale dominant features without losing information or increasing ambiguous information.

We formulate the attention integration in the MSFR module as follows:

$$\begin{aligned}\hat{x}_1 &= \omega_p\left(x_1, x_1', x_1^{\circ}\right) = x_1 + x_1' \cdot \mathrm{PSA} = x_1 + x_1' \cdot \varphi_p\left(x_1^{\circ}\right)\\ \hat{x}_i &= \omega_m\left(x_i, x_i^{\circ}\right) = x_i + x_i \cdot \mathrm{MCA} = x_i + x_i \cdot \varphi_m\left(x_i, x_i^{\circ}\right)\\ x_{i-1}^{\circ} &= \mathrm{up}\left(x_i^{\circ}\right),\ i \in \{2, 3\} \qquad (3)\end{aligned}$$

where x_1 is the feature map with the largest size from decoder (green circle 1 in Fig. 6), x_1' is the result with the corresponding resolution upsampled from the input feature map of decoder (blue circle 4 in Fig. 6), x_1° is the corresponding feature map from encoder (blue circle 1 in Fig. 6), $\hat{x}_1$ is the highest resolution feature map integrated by PSA, ω_p is the attention integration of PSA, φ_p is the attention generation of PSA, $+$ is the elementwise addition, and $\cdot$ is the elementwise multiplication; x_i denotes the low-resolution feature maps from decoder (green circles 2 and 3 in Fig. 6, i means the number for different layer), x_i° represents the corresponding feature maps form encoder (blue circles 2 and 3 in Fig. 6), $\hat{x}_i$ are the feature maps integrated by MCA, ω_m is the attention integration of MCA, φ_m is the attention generation of MCA, up denotes the upsampling unit composed of convolution, batch normalization, ReLU activation function, and bilinear interpolation.

Assigning each pixel with a corresponding weight on the space, the spatial attention is supposed to have a good preservation of geometrical information, and this is important for the refinement of spatial details. Moreover, channel attention aims to identify the effective dimensions of features, which can be used to select the dominant features at each scale. In addition, self-attention is expected to describe the distance between feature elements so as to construct the correlation between the feature vectors of each category.

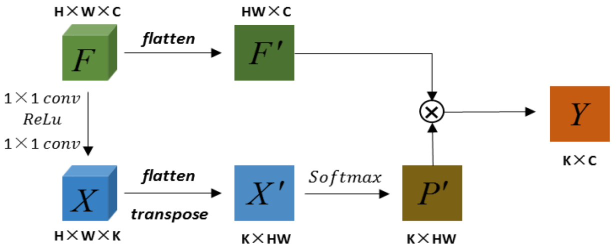


Fig. 7. Workflow for generating per-class feature vectors, where F denotes the feature map, X denotes the correlation map, F', X', and P' mean to be the corresponding matrices, $\otimes$ means the matrix multiplication, C is the number of channels, K is the number of classes, and Y is the per-class feature vectors.

C. Interclass Attention Weighting

In order to reduce the erroneous classification caused by large intraclass discrepancy and strong interclass correlation, we propose the ICAW architecture based on the OCR model [36] and ACFNet [58]; more details are explained in the following text.

1) Per-Class Feature Vectors: Taking intraclass changes and the association between pixels and different categories into account, we build per-class feature vectors to represent the corresponding holistic features, which is similar to the OCR model [36] and ACFNet [58].

In general, a series of 1 × 1 convolution and activation functions are applied at the end of the network to generate a tensor (it is named a correlation map for simplicity) with the number of channels equal to the number of categories, in which the response value on each channel represents the association between pixels and the corresponding categories. After flattening, transposing, and Softmax activation (as shown in Fig. 7), the correlation map can be transferred into a matrix, where each row denotes a category and the possibility of each pixel belonging to it (indicated by P' in Fig. 7). On the other hand, the matrix flattened by the feature map indicates the multidimensional features for each pixel (indicated by F'in Fig. 7). In addition, it is obvious that the multiplication between the above two matrices leads to the per-class feature vectors (indicated by Y in Fig. 7), which is in fact a sum of the pixel features with categorical probabilities as weights. Equation (4) shows the generation of per-class feature vectors

$$\begin{aligned}{F'}^{HW\times C} &= \mathrm{flatten}\left(F^{H\times W\times C}\right)\\ X^{H\times W\times K} &= g'\left(F^{H\times W\times C}\right),\quad g' = \mathrm{conv}\left(\mathrm{relu}\left(\mathrm{conv}\left(.\right)\right)\right)\\ {X'}^{K\times HW} &= \mathrm{flatten}\left(X^{H\times W\times K}\right)^{\mathrm{Tran}}\\ {P'}^{K\times HW} &= \mathrm{softmax}\left(\mathrm{flatten}\left(g'\left(F^{H\times W\times C}\right)\right)\right)\\ Y^{K\times C} &= {P'}^{K\times HW} \otimes {F'}^{HW\times C} \qquad (4)\end{aligned}$$

where g'consists of two 1 × 1 convolutions and ReLU activation.

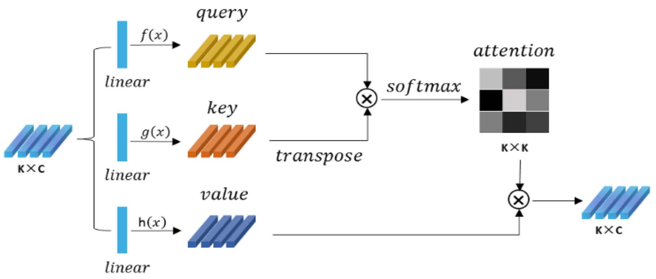
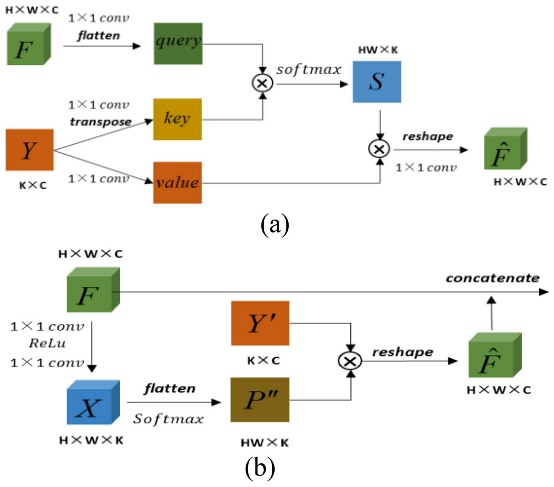


Fig. 8. Workflow for generating interclass attention. f, g, and φ represent the different linear layers, and $\otimes$ means the matrix multiplication.

2) Interclass Attention: For the interclass association, we perform the ICAW on the per-class feature vectors from Section III-C1 with the multihead attention mechanism [39]. As shown in Fig. 8, different linear transformations are deployed to derive different matrices for estimating corresponding self-attention, in which the query aims to match other categories, key is designed to be matched, and value means the information that is expected to be extracted. The matrix multiplication between query and key results in the interclass covariance matrix as the attention, which represents the similarity measure between categories. Finally, features for each class weighted by the interclass attention can be obtained after the matrix multiplication between the covariance matrix and value. Moreover, multihead strategy makes it possible to get multiple weighted results, each head may focus on a different range of information. In particular, skip connection is applied to deliver the original per-class feature vectors to prevent the information from vanishing or mutating. Our interclass attention is formulated as (5), where three linear projection layers (linear_*f*, linear_*g*, and linear_*h*) are used for query, key, and value

$$\text{query} = \text{linear\_}f\left(Y^{K\times C}\right), \text{key} = \text{linear\_}g\left(Y^{K\times C}\right)^{\text{Tran}}$$
$$\text{attention}^{K\times K} = \text{softmax}\ (\text{query},\ \text{key})$$
$$Y'^{K\times C} = \text{attention}^{K\times K} \otimes \text{linear\_}h\left(Y^{K\times C}\right). \quad (5)$$

3) Feature Augmentation: Considering category-level features and reinforcing the representation for pixel-level classification, the final step in ICAW is to integrate the per-class feature vectors with the original features extracted by the network. As shown in Fig. 9, the cross attention is applied in the OCR model [36] to obtain OCRs, in which query is extracted from the original feature map and the per-class feature vectors contribute to key and value, the association between pixels and categories is explored, similar to the decoder in Transformer [39]. Inspired by the OCR model, a similar yet simple solution is proposed, specifically, we directly use the matrix multiplication between per-class feature vectors (indicated by Y' in Fig. 9) and the probability that each pixel belongs to it (indicated by P'' in Fig. 9), analogous to the generation of per-class feature vectors. In addition, both methods concatenate the original feature map and the corrected one on the channel dimension with skip connection, improving the robustness of this model.



(a)

(b)

Fig. 9. Workflow for feature augmentation in the (a) original OCR model and (b) in our method, where F is the input feature map, X is the correlation map, $\hat{F}$ is the output feature map, S is the covariance matrix, P'' is the probability matrix, $\otimes$ means the matrix multiplication, Y is the original per-class feature vectors, and Y' is the per-class feature vectors weighted by the interclass attention.

D. Loss Function and Prediction

In essence, the fundamental categorical cross-entropy loss that measures the discrepancy between predicted results and ground truth labels is used, as (6) illustrates, K is the number of categories, I is the input image whose pixel number is N, $p_i = [p_1, \ldots, p_K]$ indicates the inferenced probability distribution, $t = [t_1, \ldots, t_K]$ is the one-hot label of the corresponding pixel, and $t_i = 1$ if the related pixel belongs to class i, otherwise $t_i = 0$

$$\text{Loss}\ (p_i, t_i) = -\frac{1}{N}\sum_{p\in I}^{N}\sum_{i=1}^{K} t_i \log\left(p_i\right) \quad (6)$$

$$L = \text{Loss}_{\text{PL}}\ \left(p_i^{\text{PL}}, t_i\right) + \alpha\text{Loss}_{\text{AL}}\ \left(p_i^{\text{AL}}, t_i\right). \quad (7)$$

In this article, to better guide our model training, we employed (7) as overall Loss L, which is linearly summed by two categorical cross-entropy losses: $\text{Loss}_{\text{PL}}(p_i^{\text{PL}}, t_i)$—principal loss, measuring the discrepancy between coarse segmentation and ground truth before applying ICAW; and $\text{Loss}_{\text{AL}}(p_i^{\text{AL}}, t_i)$—the auxiliary loss, estimating the discrepancy between segmentation results and labels after using ICAW. More specifically, α is the balance parameters, which are set to be 0.5 in this work, p_i^{PL} and p_i^{AL} are the predicted probabilities of each pixel before and after using ICAW, respectively.

In order to reduce the influence of random errors on the prediction results, we vote the predictions from three independent training for testing precision metrics and analyzing visualization. However, for efficiency metrics, we evaluate them through one training process, taking ICAW into consideration. For a single prediction, as shown in Fig. 10, we apply the 1 × 1 convolution to generate the correlation map from the last feature map in the network, and the index of the response peaks on the channel dimension is the forecasted category for each pixel in the correlation map. In the voting strategy, we first apply Softmax

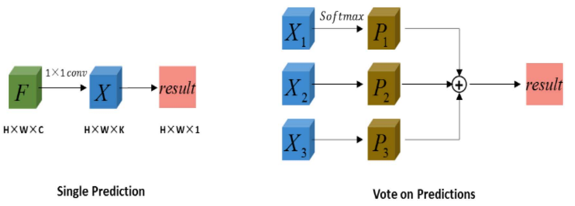


Fig. 10. Workflow for voting predictions. F denotes the feature map, X, X_1, X_2, and X_3 represent the different correlation maps and P_1, P_2, and P_3 means the different possibility maps, and $\oplus$ is the elementwise addition.

activation to obtain the probability map from the correlation map in which the value on each channel represents the probability that pixels belong to the corresponding categories. Then, we use elementwise addition for the three probability maps, which leads to the final result. Consequentially, the voting strategy is able to effectively evaluate different methods.

IV. Experiments

To demonstrate the performance of our work, two ablation studies are first conducted: first, experiments on the ISPRS Vaihingen two-dimensional (2-D) dataset [60] and LoveDa dataset [61] are tested to verify the synergy and efficacy of the proposed modules; second, various serial learning backbones (e.g., *ResNet*, *SegFormer*, *MobileNet,* etc.) are deployed with the proposed modules and the results of 2020 National Artificial Intelligence Competition (NAIC) in China are reported. In addition, we compare our method to several state-of-the-art methods on the ISPRS Potsdam 2-D dataset [60]. In the following sections, we will introduce more details in regard to the datasets, experimental settings, and our experimental results.

A. Experimental Datasets

1) ISPRS 2-D Challenge Dataset: It is a widely studied benchmark dataset in the ISPRS community, including high-resolution true orthophoto (TOP) images from two different regions acquired by aircraft, Vaihingen, and Potsdam [60]. Both of them contain a normalized digital surface model (NDSM), and six land cover classes are investigated, including *impervious surface* (*imp. surf.*), *building* (*build.*), *low vegetation* (*low veg.*), *tree*, *car,* and *clutter*. In particular, Vaihingen is a rural area containing three spectral bands of NIR, red, and green, and has 33 images with size of about 2500 × 2000 pixels (11 images are selected as training data, 5 are used as validation sets, and 17 are divided into test sets), and the areas of *low veg.* and *tree,* which have similar spectral response, are relatively higher than that of *car* and *clutter*. On the contrary, Potsdam is basically an urban region containing four spectral bands of NIR, red, green, and blue, where the areas covered by *low veg.* and *tree* (most of which are branches with few leaves) are relatively small; there are 38 images of 6000 × 6000 pixels, which are divided into 17 training samples, 7 validations, and 14 test sets.

In our experiments, only TOP images are exploited. For Vaihingen, we crop training samples into images with size of 512 × 512 pixels and generate 4326 training images and 111 validation images through augmentations (such as rotation and flip); the performance of the proposed modules is ablatively analyzed and evaluated based on the whole image from the 17 test samples. For Potsdam, to cope with the imbalanced allocation of training and testing samples, the original training and validation images are cut into 8664 images with the size of 512 × 512 pixels for our training. Four images are randomly selected from the original test images, which are then cropped into 1444 images for validation, and the remaining ten original test images are cropped into 3610 images as our test images. The Potsdam dataset is employed to demonstrate the efficacy of our whole framework composed of MSFR and ICAW when compared with other relevant existing methods.

2) LoveDa Dataset: Deriving from Google Earth, the LoveDa dataset [61] constitutes 5987 high-resolution remote sensing images of urban and rural scenes in Nanjing, Changzhou, and Wuhan, whose spatial resolution is 0.3 m and image size is 1024 × 1024 pixels; the original image number for training, validation, and testing is 2522, 1669, and 1796, respectively. Similar to the ISPRS challenge dataset, after data augmentation and cropping into size of 512 × 512 pixels, 28 288 images are input for training, 1053 images are validated during training, and 7739 images are tested. Seven land cover categories are studied, including *background* (*B.G.*), building (*build.*), *road*, *water*, *barren*, *forest,* and *agriculture* (*Agr.*) [61]. In this dataset, the characteristics of ground objects in urban and rural areas are quite different, including spatial distribution and spectral response; the scales of ground objects vary a lot as well; furthermore, the contents of *B.G.* are very complicated, which results in high intraclass variance. Therefore, this dataset is also tailored to be applied for ablatively proving the efficacy of the proposed two pluggable modules (MSFR and ICAW).

3) NAIC Dataset: The NAIC dataset is a hybrid imagery benchmark for land cover classification competition,[1] it is composed of 100 000 RGB images collected from satellites and aircraft in China, and the spatial resolution is between 0.1 and 4 m. Each image has been cropped to the size of 256 × 256 pixels. As the image sources are complex in NAIC dataset, the differences in spatial resolution and spectral radiation are very explicit among images, resulting in large-scale variance as well as different spectral responses inside each category. We randomly use 89 000 images for training, 1000 images for validation, and 10 000 images for testing. The eight categories to be discerned in this dataset are: *waters*, *buildings* (*build.*), *transportation* (*trans.*), *arable land* (*ara.*), *grass*, *forest*, *bare soil* (*soil.*), and *others*. This dataset is used to test the feasibility when MSFR and ICAW are both deployed on different backbones.

B. Implementation Details

1) Experimental Settings: Our method is implemented with the Pytorch framework. The base learning rate is set to 0.001. A poly learning rate policy is employed in which the initial learning rate is multiplied by $(1 - \frac{\text{epoch}}{\text{total\_epoch}})^{0.9}$ during each epoch. All models in the reported experiments are trained with the SGD

[1] More details related to NAIC can be found at https://naic.pcl.ac.cn/contest/6/track/24

optimizer on NVIDIA GTX 3090. The momentum value is 0.9 and the weight decay value is 1e-4. For each experiment, the training procedure is with 300 epochs and the validation is conducted every five epochs. During each training epoch, we use all training images in the ISPRS Vaihingen dataset, while half of the images are taken randomly for training in the Potsdam dataset and quarter in the LoveDa dataset and the NAIC dataset. The number of all iterations for each epoch is determined by the batch size accordingly; more training information is introduced in the following sections. For the experiment with SwinT [43], we use pretrained models based on the COCO dataset [62], and for other experiments, we apply corresponding pretrained weights from ImageNet dataset [63], except the experiments with SETR [40] and ABCNet [64].

2) Evaluation Metrics: To evaluate the results, we use several common metrics of intersection over union (IoU), precision, recall, $F1$-score, overall accuracy (OA), and Kappa. The evaluation was based on an accumulated confusion matrix from all test images, whereby IoU, precision, recall, $F1$-score, OA, and Kappa can be derived

$$\mathrm{IOU}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k} \tag{8}$$

$$\mathrm{Precision}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k},\ \mathrm{Recall}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FN}_k} \tag{9}$$

$$F1 - \mathrm{score}_k = 2 \cdot \frac{\mathrm{Precision}_k \cdot \mathrm{Recall}_k}{\mathrm{Precision}_k + \mathrm{Recall}_k} \tag{10}$$

$$\mathrm{OverAccuracy} = \frac{\sum_{k=1}^{N} \mathrm{TP}_k}{\sum_{k=1}^{N} \mathrm{TP}_k + \mathrm{FN}_k} \tag{11}$$

$$pe = \frac{\sum_{k=1}^{N} (\mathrm{TP}_k + \mathrm{FP}_k) \cdot (\mathrm{TP}_k + \mathrm{FN}_k)}{\left(\sum_{k=1}^{N} \mathrm{TP}_k + \mathrm{FN}_k\right)^2},$$

$$\mathrm{Kappa} = \frac{\mathrm{OverAccuracy} - pe}{1 - pe}. \tag{12}$$

TP_k, FP_k, and FN_k denote the true positive, false positive, and false negative pixels, respectively, and k is the category index. We use mean intersection over union (mIoU) and average $F1$-score (avg. $F1$) to represent mean results for all classes. For evaluating the efficiency of methods, we apply giga floating-point operations per second (GFLOPS) to measure the cost of computation and capacity of weights (#Params.) to measure the cost of parameters. In addition, we compute the average time of eight inferences (Time) on one image using the GTX 3090 GPU during training, which assesses the inference time of each method.

C. Internal Ablation Studies for the Proposed Modules

In this section, experiments are conducted on the Vaihingen and LoveDa datasets for testing modules of MSFR (PSA and MCA) and ICAW individually and mutually. We use ResNet50 [38] as the backbone (upsampling and convolution layer-by-layer), and the batch size is set to 16 (360 iterations per epoch on the Vaihingen dataset and 442 iterations per epoch on the LoveDa dataset).

1) Results on Vaihingen Dataset: For Vaihingen, three spectral bands of NIR, red, and green are used; the classification accuracy and inference efficiency with/without the proposed pluggable modules are quantitively shown in Table I (eight variants are compared). The configuration of baseline is to simultaneously switch OFF our MSFR and ICAW modules. As expected, the full variant with the two modules of MSFR and ICAW plugged is always the best in terms of Kappa, mIoU, avg.$F1$, and OA. In particular, comparing with the baseline, the improvements of Kappa, mIoU, avg.$F1$, and OA are 7.45%, 10.46%, 8.39%, and 5.64%, respectively. In addition, the superiority of each proposed module can be numerically demonstrated: the variant with MCA improves the OA by 3.6%, the one with PSA can make an improvement of 3.97%, and the one with ICAW increases OA by 1.28% inc. On the other hand, as Table I lists, there are extra overheads in terms of "GFLOPS" and "#Params." after embedding our modules. However, comparing with the baseline, the full variant with MSFR and ICAW only needs about 3 ms more for inference, which is almost negligible. Therefore, the proposed modules can significantly improve the classification performance of baseline with backbone ResNet50, yet with negligible extra inference time.

In order to qualitatively demonstrate the effect of PSA and MCA, we visualize these two attentions in Fig. 11. Specifically, the visualization of the MCA model is applied after the multiplication with the corresponding feature map. The PSA model can observe more details on *Build.* and *Imp. Surf.*, which might predict clearer boundaries; however, the PSA model also captures redundant details, the edge of shadows for example (shown in black circle of Fig. 11). As for the MCA model, MCA(1/8) of higher resolution is more sensitive to *Tree* and the van (shown in red rectangular of Fig. 11), while MCA(1/16) of lower resolution pays more attention to the cars (shown in white rectangular of Fig. 11). MCA model of various resolutions also shows some global consistency and makes responses to *Car* at different scales; this reflects the capability against multiscale variation. In general, the attention from MSFR (including PSA and MCA) focuses on objects of various sizes, leading to abundant features from the image.

As shown in Fig. 12, we plot bar graphs with avg.$F1$ and visualize corresponding classification results to illustrate the effectiveness of PSA, MCA, and ICAW models when dealing with the problems of inaccurate edges, fragmented segmentation, and erroneous classification. Fig. 12(a) indicates the improvement after deploying PSA, and the ambiguous boundaries on *Car* (white box) basically do not exist anymore. Fig. 12(b) shows the improvement from MCA, the baseline generates some incorrect fragmented segments (as the black circles show), which are refined by plugging our MCA model. Combing PSA and MCA, the generated MSFR module is supposed to be capable to deal with the ambiguous edge prediction and fragmented segmentation, Fig. 12(c) depicts that the boundaries of *Car* (black circle) are more accurate and the fragmentation noise inside *Build.* (black box) is erased; however, the erroneous classification on *Build.* (red box) still exists, i.e., the shadow increases feature discrepancy inside *Build.* and makes features similar between *Build.* and *Imp. Surf.* Fig. 12(d) indicates the improvement of

TABLE I
QUANTITATIVE RESULTS OF LAND COVER CLASSIFICATION FOR ABLATION STUDY ON THE VAIHINGEN DATASET

ResNet50	*MCA*	*PSA*	*ICAW*	Kappa[%]	mIoU[%]	avg. F1 [%]	OA[%]	GFLOPS[G]	#Params.[M]	Time[ms]
√	-	-	-	74.45	56.54	70.52	80.68	**24.50**	**28.67**	**12.09**
√	√	-	-	79.19	61.51	74.54	84.28	32.06	31.96	13.22
√	-	√	-	79.69	62.74	75.67	84.65	28.44	29.08	12.59
√	-	-	√	76.12	57.75	71.41	81.96	24.81	29.80	13.46
√	√	√	-	81.40	64.62	77.02	85.95	33.27	32.04	14.09
√	√	-	√	80.87	63.62	76.20	85.56	32.36	33.10	14.46
√	-	√	√	81.00	63.68	76.17	85.66	28.74	30.21	14.08
√	√	√	√	**81.90**	**66.90**	**78.91**	**86.32**	33.57	33.17	14.96

The baseline model is *ResNet50*, and √ indicates that the relevant module is applied. Best scores are in bold font.
The best results are highlighted in bold.

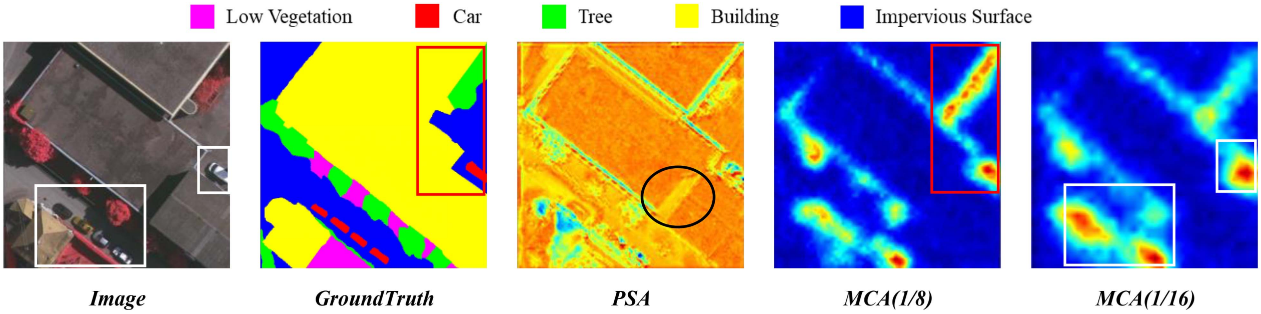


Fig. 11. Visualization of the attentions from the MSFR architecture, in which MCA(1/8) denotes the heatmap of one-eighth the size of the input image, and MCA(1/16) denotes the heatmap of 1/16 the size of the input image. Black circle contains the shadow captured by PSA, white rectangular contains cars captured by MCA(1/16), and red rectangular contains the trees and the van captured by MCA(1/8).

ICAW on erroneous classification, e.g., more correct classification results are obtained in spite of large intraclass differences inside *Build.* and similar interclass distances between *Build.* and *Imp. Surf.* (black box). Note that the ICAW model does not completely eliminate the erroneous classification; significant improvement is made over the baseline method.

Furthermore, a more comprehensive internal ablation study is provided with detailed qualitative results, i.e., more classification results are visualized in Fig. 13. The baseline model performs smooth predictions with ambiguous boundaries, while the PSA model shows clear edges, especially on cars (shown in white rectangular of Fig. 13). However, the PSA model also generates noisy fragmented details inside objects (shown in white circle of Fig. 13), as well as jagged edges on buildings due to shadows (shown in black rectangular of Fig. 13). On the contrary, the MCA model generates less noise with poor boundaries and tends to be more representative in global due to the reconstructed multiscale features. Moreover, the proposed ICAW works well with shadows, which is probably because the interclass correlation and intraclass variance are considered. In addition, the classification results after deploying both MSFR and ICAW are rewarding, in which the global segmentation looks more reasonable and the local boundaries are basically improved; this in turn demonstrates that the combination of the two proposed modules is effective.

From Table I and Fig. 13, the proposed modules can significantly improve the baseline with just slightly extra computation cost. Nevertheless, ambiguity still exists between *Tree* and *Low. Veg.,* especially shrubs (shown in the black circle of Fig. 13), which are challenging without using the information of NDSM.

2) Results on LoveDa Dataset: To further explore the generalizability of our proposed modules and avoid the coincident superior performance obtained just on Vaihingen, we also conduct the internal ablation study using LoveDa dataset, which is composed of both satellite remote sensing images and aerial images. The image resolution of LoveDa dataset is generally lower than that of Vaihingen and two different areas are observed, urban and rural, making it more challenging for ambiguous boundaries, fragmented segmentation, and erroneous classification. The number of training batches is set as the same as Vaihingen in this experiment.

Table II provides the classification results and time efficiency, similar to Vaihingen; all the proposed modules are able to improve the classification performance, while the magnitude of improvements is in general slightly smaller than that of Vaihingen. Specifically, comparing with baseline, PSA, MCA, and ICAW are able to improve the avg.$F1$ by 1%, 1.01%, and 0.87%, respectively. When investigating the performance of combinations between various modules, it can be found that MSFR consisting of PSA and MCA can further improve avg.$F1$

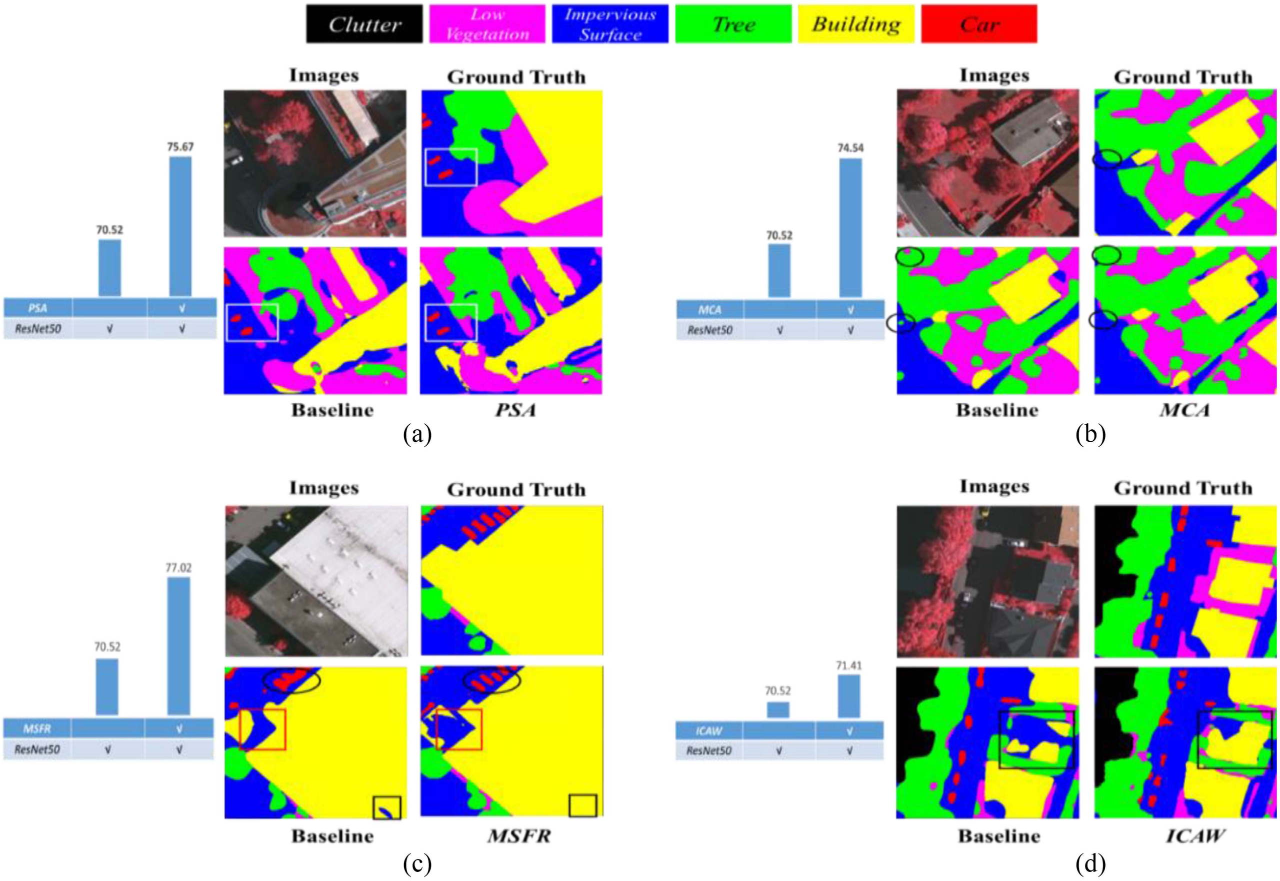


Fig. 12. Visualization of results on the ablation study with the Vaihingen dataset.

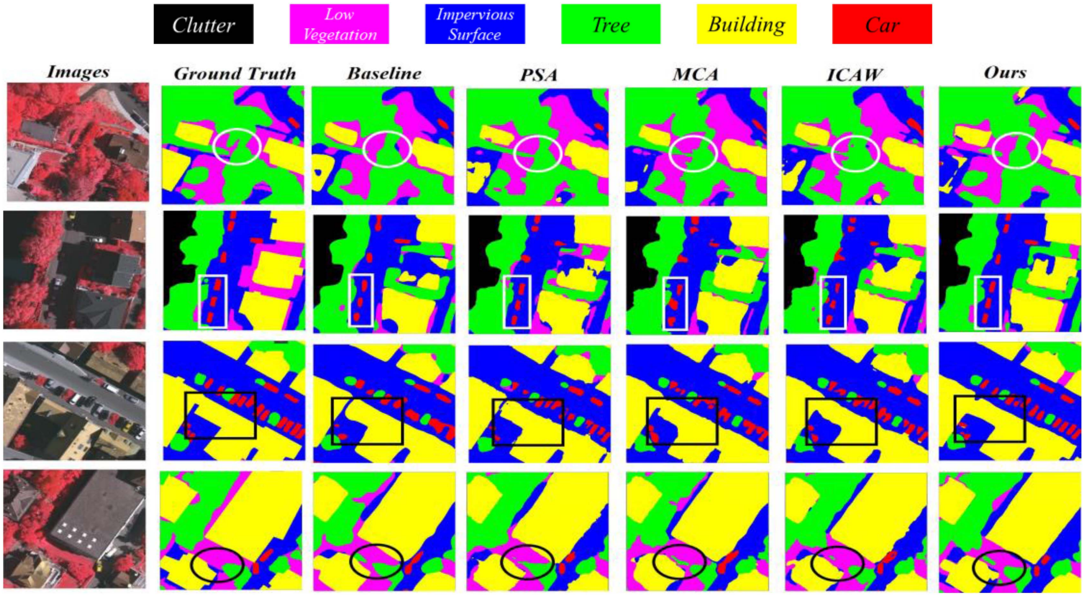


Fig. 13. Examples for classification results of ablation studies on the Vaihingen dataset. We visualize the results of PSA, MCA, and ICAW via deploying on the baseline individually and together (*ours* for together). White rectangular contains cars, white circle contains noise, black rectangular contains edges of buildings, and black circle contains ambiguity between trees and low vegetation.

TABLE II
QUANTITATIVE RESULTS OF LAND COVER CLASSIFICATION FOR ABLATION STUDY ON THE LOVEDA DATASET

ResNet50	*MCA*	*PSA*	*ICAW*	Kappa[%]	mIoU[%]	avg. F1 [%]	OA[%]	GFLOPS[G]	#Params.[M]	Time[ms]
√	-	-	-	68.99	59.53	74.23	76.67	**24.50**	**28.67**	**12.34**
√	√	-	-	70.24	60.78	75.24	77.59	32.06	31.96	13.46
√	-	√	-	69.35	60.74	75.23	76.89	28.44	29.08	12.84
√	-	-	√	70.13	60.64	75.15	77.54	24.81	29.80	13.84
√	√	√	-	71.14	62.13	76.32	78.20	33.27	32.04	13.96
√	√	-	√	70.44	61.05	75.46	77.70	32.36	33.10	14.84
√	-	√	√	71.02	62.19	76.38	78.14	28.74	30.21	14.46
√	√	√	√	**71.21**	**62.38**	**76.54**	**78.28**	33.57	33.17	15.34

The best results are highlighted in bold.

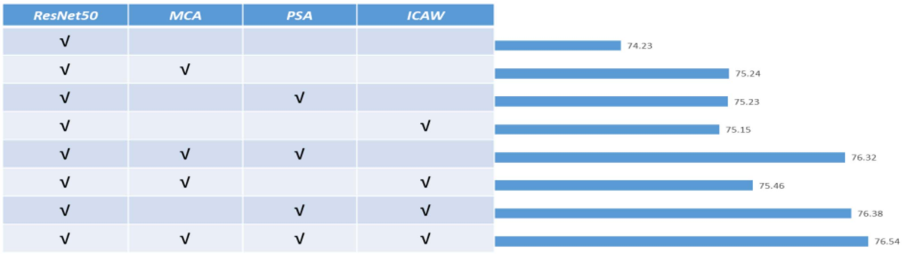


Fig. 14. Bar chart of the avg.$F1$ for ablation study on the LoveDa dataset.

by 1.09% and 1.08%, and the integration of PSA and ICAW improves the avg.$F1$ by 1.15% and 1.23%. Plugging in both MSFR and ICAW is always the best, whose avg.$F1$ is 1.08% higher than only applying MCA and ICAW. For the other evaluation metrics, comparing with the baseline, utilizing proposed MSFR and ICAW can improve Kappa, mIoU, OA, and avg.$F1$ by 2.22%, 2.85%, 1.61%, and 2.31%, respectively. As for the computational cost, the metrics of "GLFOPS" and "#Params." in Table II are consistent with the corresponding values in Table I, whereas "Time" only varies very slightly, this is because the experimental settings are basically the same (constant running machine and tested models) and the experimental image size (512 × 512) is identical with Vaihingen.

To explicitly highlight the ablation results, the avg.$F1$ of various variants with different modules is plotted in this section as a bar chart, as shown in Fig. 14; it can clearly depict that all the proposed modules and their corresponding combinations can apparently improve the baseline, in which PSA is of the best portability, as the avg. $F1$ scores typically show a significant increase after embedding PSA into other variant models.

Similar to Fig. 13, the classification results of LoveDa using various model configurations are qualitatively compared in Fig. 15. Akin to Vaihingen, the baseline method obtains smoother edges as a whole, and PSA predicts more accurate boundaries while adding fragmented segmentations, redundant internal features, and jaggedness around boundaries, as shown in Fig. 15(black boxes). Both MCA and ICAW are capable to eliminate some incorrect fragmented segmentations and erroneous classifications (as shown by the black and white circles in Fig. 15), but boundaries are yet not refined. From the last column of Fig. 15, deploying all the proposed modules together into the baseline can basically maintain their original superiority, which results in more accurate boundaries and less fragmentation segmentation, and erroneous classification.

Comparing the quantitative and qualitative results between Vaihingen and LoveDa, the LoveDa dataset contains very rich internal features of objects, especially inside *Build.* in urban areas, as shown by the white boxes in Fig. 15, and PSA may pay more attention to some redundant but useless details, which might lead to that the corresponding improvement on LoveDa is not as good as Vaihingen. In addition, *Barren*, *Forest*, *B.G.,* and some *Agr.* in LoveDa show a similar spectral response and texture, and *Road* is also much narrower and longer than *Imp. Surf.* in the Vaihingen, which is a very challenging task for land cover classification. Therefore, the improvement of our proposed modules (including each individual module or their combinations) on LoveDa is a little bit lower than that on Vaihingen. Nevertheless, the efficacy and superiority of embedding the proposed MSRF and ICAW are successfully demonstrated on both LoveDa and Vaihingen; we can expect a high possibility for effectively generalizing our method on other datasets.

D. Ablation Study on Different Backbones

To validate the effectiveness and feasibility of the proposed MSFR and ICAW modules on various backbones, based on the NAIC dataset, the performance of *MobileNet* [65], *GoogleNet* [66], *Xception* [67], *HRNet* [48], *SETR* [40], and *SegFormer* [44] are explored in which the first one is a relatively lightweight model, *HRNet* is parallel architecture, and the last two are transformer-based architectures. In addition, each baseline applies the multiscale feature fusion architecture proposed by FCN [24]. In this section, the batch size is set to 12 (1854

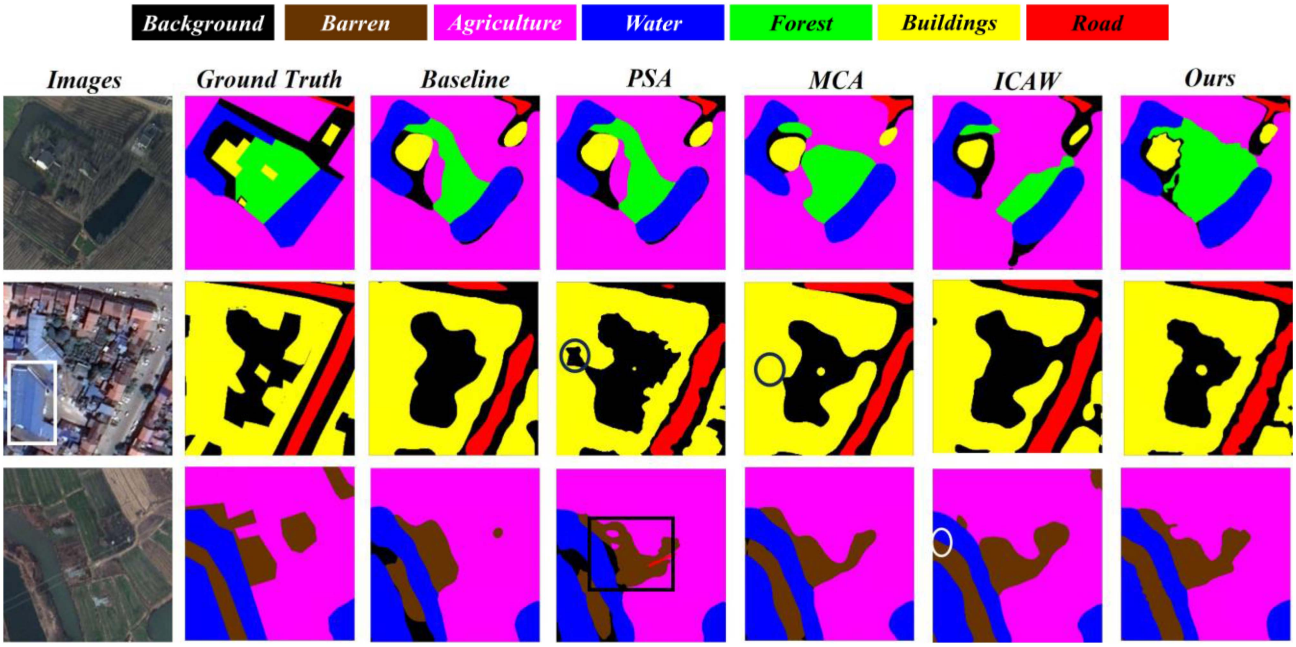


Fig. 15. Examples for classification results of qualitative ablation studies on LoveDa dataset.

TABLE III
Classification Results on the NAIC Dataset

Networks	F1 [%]								Kappa[%]	mIOU[%]	avg. F1 [%]	OA[%]
	Waters	*Trans.*	*Build.*	*Ara.*	*Grass*	*Forest*	*Soil.*	*Others*				
MobileNet	88.10	59.92	79.52	72.49	61.60	80.15	60.50	63.56	68.25	55.71	70.73	73.00
MobileNet +ours	**91.83**	**66.56**	**84.32**	**79.02**	**70.56**	**83.82**	**71.21**	**70.91**	**75.18**	**63.73**	**77.28**	**78.86**
ResNet	92.31	74.15	87.77	81.66	76.52	87.51	76.15	75.97	79.70	69.29	81.50	82.68
ResNet+ours	**93.55**	**77.58**	**89.58**	**82.58**	**79.13**	**88.76**	**79.73**	**78.03**	**81.81**	**72.28**	**83.62**	**84.48**
Xception	92.59	78.73	89.51	82.42	79.98	88.82	80.58	78.30	81.89	72.57	83.87	84.54
Xception +ours	**93.32**	**79.94**	**90.81**	**82.73**	**80.63**	**90.21**	**81.27**	**79.28**	**83.00**	**73.95**	**84.77**	**85.87**
GoogleNet	92.83	72.74	87.30	81.17	74.99	85.98	75.60	74.62	78.69	68.15	80.65	81.82
GoogleNet+ours	**93.02**	**74.42**	**88.31**	**81.69**	**76.31**	**86.97**	**77.26**	**75.55**	**79.78**	**69.58**	**81.69**	**82.74**
HRNet	**94.06**	80.34	90.85	83.30	80.52	90.01	81.62	79.83	83.32	74.39	85.06	85.77
HRNet+ours	93.85	**80.67**	**91.12**	**83.65**	**80.92**	**90.43**	**81.75**	**79.95**	**83.61**	**74.73**	**85.29**	**86.01**
SETR	90.59	63.38	82.51	75.72	69.50	**84.07**	69.41	67.64	72.90	61.29	75.35	76.92
SETR+ours	**90.96**	**64.40**	**83.43**	**76.16**	**69.94**	83.80	**69.94**	**68.66**	**73.52**	**61.99**	**75.91**	**77.43**
SegFormer	**93.56**	76.02	**89.81**	**83.24**	79.61	88.98	79.37	77.87	81.90	72.23	83.56	84.56
SegFormer+ours	92.92	**77.05**	89.79	83.01	**80.06**	**89.69**	**80.27**	**78.42**	**82.21**	**72.68**	**83.90**	**84.83**

The best results are highlighted in bold.

iterations per epoch) for *Xception* and 16 (1391 iterations per epoch) for the rest. Table III present the numerical results, and Table IV gives details of the computation costs.

According to Table III, comparing with the original baseline, our proposed modules can generally lead to classification improvement. Over all the backbones, *MobileNet* obtains the most explicit improvements, in which Kappa, mIoU, avg.$F1$, and OA are increased by $+6.93\%$, $+8.02\%$, $+6.55\%$, and $+5.86\%$, respectively. The reason could be that the number of unknowns in *MobileNet* is relatively low; thus, the ability to capture more powerful representations might be enhanced after integrating with the proposed modules. For the other backbones, various degrees of improvement are made by embedding our MSFR and ICAW. For instance, comparing with the original baseline of *ResNet*, *Xception,* and *GoogleNet*, the corresponding improvements of kappa, mIoU, avg.$F1$, and OA are between 1.11% and 2.11%, 1.38% and 2.99%, 0.9% and 2.12%, and 0.92% and 1.8%, respectively. However, the overall evaluation metrics of *HRNet*, *SETR,* and *SegFormer* show that only limited improvements can be made (up to $+0.5\%$ and $+0.7\%$ for *HRNet* and transformer-based models, respectively), and the $F1$ scores of some categories get slightly worse; this could be explained by the fact that *HRNet* deploys parallel learning architecture and our methods are more tailored for serial learning networks, and the transformer-based methods themselves have already applied attention mechanism to improve the model (e.g., enlarge the reception field, learning to connect encoder and decoder, etc.), which might make extra attention operations be a little superfluous. In addition, our modules yield extra computation cost, i.e., +0.46–3.93G of "GFLOPS," +0.48–2.36M of "Params.," and

TABLE IV
COMPUTATION COST OF ALL MODELS ON THE NAIC DATASET

Networks	GFLOPS[G]	Params.[M]	Time[ms]	Networks	GFLOPS[G]	Params.[M]	Time[ms]
MobileNet	0.72	2.41	8.73	*MobileNet +ours*	1.46	3.88	10.48
ResNet	7.94	31.77	12.72	*ResNet+ours*	8.40	33.17	14.09
Xception	41.27	44.82	27.18	*Xception +ours*	44.89	44.34	29.80
GoogleNet	3.90	10.29	16.21	*GoogleNet+ours*	6.94	12.31	17.08
HRNet	23.41	65.85	73.80	*HRNet+ours*	25.48	67.50	75.43
SETR	23.95	93.63	10.72	*SETR+ours*	27.88	95.99	13.59
SegFormer	5.12	26.72	19.83	*SegFormer+ours*	6.03	28.23	25.68

The best results are highlighted in bold.

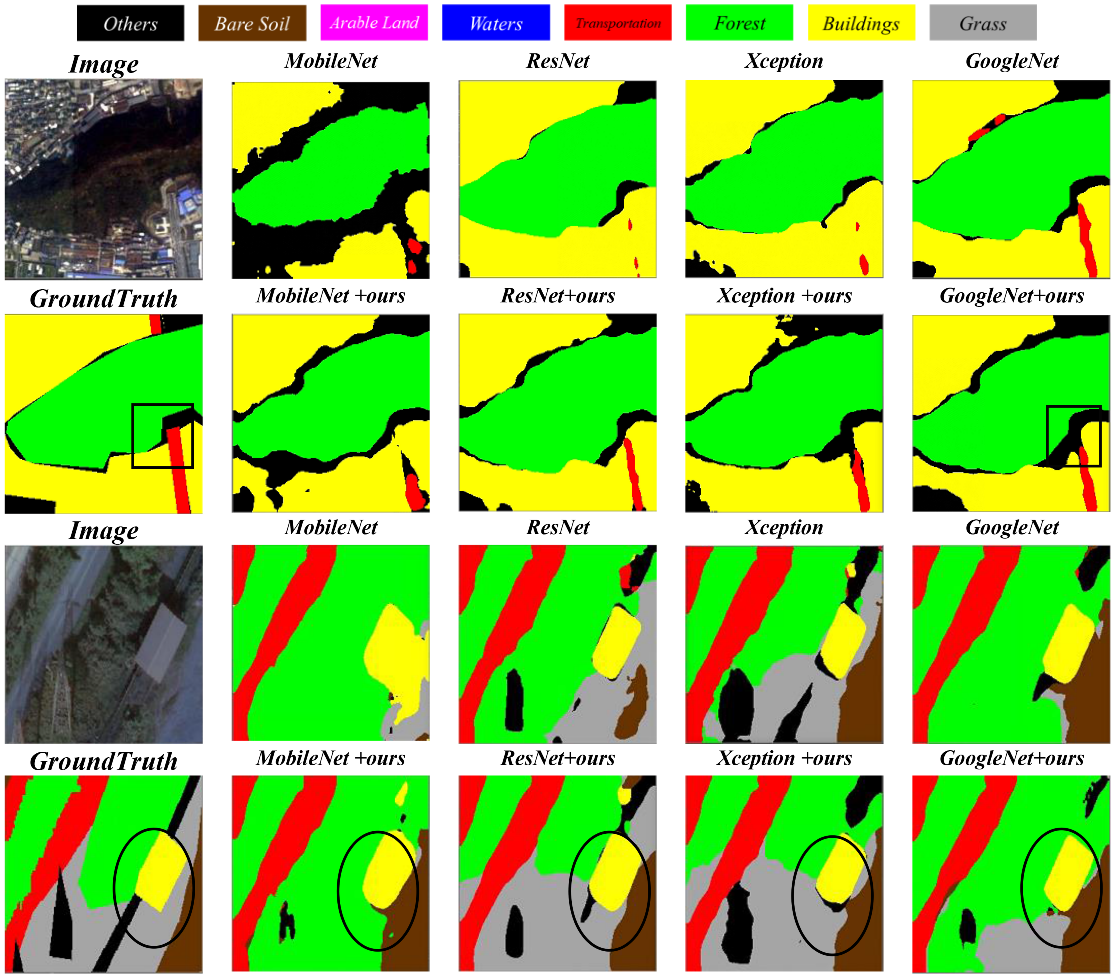


Fig. 16. Examples for classification results of the NAIC dataset using various backbones, i.e., *MobileNet*, *ResNet*, *Xception,* and *GoogleNet* as well as the variants deployed with our proposed modules.

0.87–5.85 ms of "Time," which is in fact negligible for general land cover classification task.

In general, the proposed modules can be feasibly plugged into various serial learning networks and improve their corresponding land cover classification performance; in particular, the lightweight model can typically get better improvement.

Fig. 16 shows some visualization of classification results on several serial convolution networks, and we can find that the experiments on *ResNet* and *Xception* are the best. Our modules can significantly improve boundaries, especially on *Forest*, *Trans.*, and *Build.* (black box in Fig. 16), mainly due to the details refined by the proposed PSA model. Moreover, our methods also show advantages in correcting some wrong segmentations for *Soil.*, *Grass*, *Waters*, etc. (black circles), which could be attributed to the consistent multiscale features generated by our MCA model and identifiability for various categories improved by our ICAW architecture. However, some noises inside the predictions exist for the proposed method; this is possibly caused by redundant details from the PSA model or ambiguous category features.

TABLE V
QUANTITATIVE RESULTS COMPARED WITH STATE-OF-ARTS ON POTSDAM DATASET

method	F1 [%]						Kappa [%]	mIoU [%]	avg. F1 [%]	OA[%]	GFLOPS[G]	Params.[M]	Time[ms]
	Imp. Surf.	*Clutter*	*Car*	*Tree*	*Low. Veg.*	*Build.*							
ours	89.76	48.78	89.19	86.49	83.95	95.72	84.55	72.42	82.32	88.27	33.57	33.17	15.21
ABCNet	84.14	31.02	81.31	78.01	77.17	90.68	75.96	61.53	73.72	81.76	15.58	13.39	9.23
HRNet+OCR	**90.64**	**51.70**	**90.01**	**86.65**	84.00	**96.08**	**85.17**	**73.48**	**83.18**	**88.71**	106	66.94	76.05
Segformer	90.03	47.52	88.31	86.35	83.92	95.57	84.50	71.98	81.95	88.20	20.48	26.72	20.57
SwinT	90.44	51.36	88.32	86.04	**84.41**	95.70	84.88	72.74	82.71	88.49	29.65	30.85	17.83
ConvNeXt	89.74	48.54	88.50	85.37	82.47	95.03	83.45	71.33	81.61	87.38	28.29	31.17	9.98

The best results are highlighted in bold.

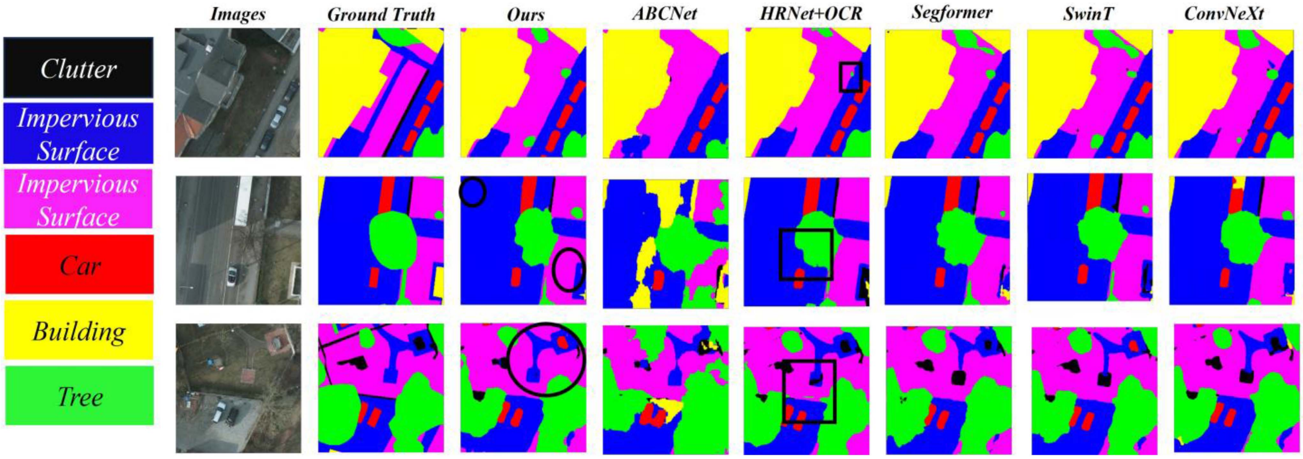


Fig. 17. Examples for classification results of our method and the state-of-the-art methods on the Potsdam dataset.

E. Comparison With the State-of-Arts

In this section, based on Potsdam dataset, several state-of-the-art methods are compared, i.e., *SwinT* [43], *HRNet+OCR* [36], *SegFormer* [44], *ABCNet* [64], and *ConvNeXt* [1], and the backbone (*ResNet50* [38] is selected to see how far we can benefit on a middle-level backbone when comparing with other methods) integrated with our proposed MSFR and ICAW is indicated as our method. The training batch size is set to 8 (540 iterations per epoch) for *HRNet+OCR* and *SwinT*, and 16 (270 iterations per epoch) for all the others.

The classification results and relevant computation costs are listed in Table V; we can easily figure out that *HRNet+OCR* basically performs the best, whose Kappa, mIoU, avg.$F1$, and OA are 0.62%, 1.06%, 0.86%, and 0.44% than our method. Nevertheless, *HRNet+OCR* is more costly, which needs three times "GFLOPS," twice the "Params.," and about five times inferencing "Time" as our method does. *ABCNet* shows the worst results, e.g., the corresponding avg. $F1$ is 8.6% poorer than our method; this is mainly due to that *ABCNet* is trained from scratch without pretrained model and still underfitting when implementation settings are as the same as other methods, whereas the cost of this method is the least. In addition, our method also outperforms *ConvNeXt* 0.71% and *SegFormer* 0.37% in avg.$F1$, and is 0.39% inferior to *SwinT*. In general, the discrepancies of classification performance among compared methods (except for *ABCNet*) are close; this is probably due to some inherent characteristics of Potsdam, e.g., number of training, validation and testing images, sample ground object distribution, etc., which might lead to limited classification improvement even if a larger model is trained. When investigating the computation cost, our method needs more parameters and computation with less inference time than *SegFormer* and *SwinT* do, this is because GPU offers algorithmic acceleration for kernel-based convolutional operations, while linear layers are the fundamental units in transformer-based methods.

In general, it can be found that our method takes the middle land on both accuracy and efficiency, and obtains the best balance when similar performance is achieved among the methods, especially comparing with *HRNet+OCR* of the highest accuracy with the lowest efficiency and the opposite *ABCNet*. In summary, our method deployed with MSFR and ICAW on a simpler backbone achieves a reasonable balance between model accuracy and inference efficiency compared with other methods.

Classification results are visualized in Fig. 17. *HRNet+OCR* shows clear boundaries on *Car* and *Build.*, but there is also quite a lot of noise and fragmented segmentations (black boxes), which result from the redundant and incomplete features exchanged among the parallel branches at different scales. Our method predicts refined boundaries that are close to *HRNet+OCR* and *SwinT*; however, there is still some noise and erroneous classification in the results (black circles), especially on *Build.* and *Imp. Surf.* with extremely similar features, which means that the robustness of ICAW needs to be further strengthened. In general, our method provides acceptable results, and the limitation is also clear that our method cannot identify objects with extremely similar spectral features, such as tiny buildings, a part of *Imp. Surf.* and *Clutter* built by cement, as well as noises in objects with large coverage areas.

V. Conclusion

In this article, two pluggable modules are proposed to improve the serial learning architectures on land cover classification task, i.e., the MSFR and the ICAW module. The first module focuses on solving the problems of multiscale variance and lost spatial details, and the second one is expected to mitigate the negative influence of large intraclass difference and strong interclass correlation. The extensive experiments on different datasets have shown that our proposed modules are able to improve the land cover classification in a remarkable degree. In particular, compared with the original backbone of ResNet50, the improvements are 7.45% in Kappa, 10.36% in mIoU, 8.39% in avg. $F1$, and 5.64% in OA, yet with only 9.07 GB extra GFLOPS and 4.7M more parameters. Deployed on different serial backbones, our methods show superior portability. Compared with state-of-the-art methods, our methods achieve comparable results, yet offer a better balance between inference time efficiency and accuracy. In future work, we would like to explore the possibility of integrating our modules into the transformer-based networks. In addition, we also want to improve our modules for better performance via investigating more datasets, such as RSIPAC[2] and Ali Tianchi.[3]

References

[1] C. Yang, F. Rottensteiner, and C. Heipke, "Towards better classification of land cover and land use based on convolutional neural networks," *ISPRS Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W13, pp. 139–146, 2019.

[2] C. Yang, F. Rottensteiner, and C. Heipke, "Investigations on skip-connections with an additional cosine similarity loss for land cover classification," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 339–346, 2020.

[3] Z. Xiong, Z. Zhan, and X. Wang, "Position-sensitive attention based on fully convolutional neural networks for land cover classification," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 281–288, 2022.

[4] R. S. Defries and J. R. G. Townshend, "NDVI-derived land cover classifications at a global scale," *Int. J. Remote Sens.*, vol. 15, no. 17, pp. 3567–3586, 1994.

[5] M. Pesaresi, A. Gerhardinger, and F. Kayitakire, "A robust built-up area presence index by anisotropic rotation-invariant textural measure," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 1, no. 3, pp. 180–192, Sep. 2008.

[6] S. W. Myint, N. Lam, and J. M. Tyler, "Wavelets for urban spatial feature discrimination," *Photogramm. Eng. Remote Sens.*, vol. 70, no. 7, pp. 803–812, 2004.

[7] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. IEEE 9th Int. Conf. Comput. Vis.*, 2003, pp. 10–17.

[8] V. Machairas, M. Faessel, D. Cardenas-Pena, T. Chabardes, T. Walter, and E. Decencière, "Waterpixels," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3707–3716, Nov. 2015.

[9] M. Pal and G. M. Foody, "Evaluation of SVM, RVM and SMLR for accurate image classification with limited ground data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 5, pp. 1344–1355, Oct. 2012.

[10] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.

[11] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. Atli Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 45–54, Jan. 2014.

[12] P. Teluguntla et al., "A 30-m Landsat-derived cropland extent product of Australia and China using random forest machine learning algorithm on Google Earth engine cloud computing platform," *ISPRS J. Photogramm. Remote Sens.*, vol. 144, pp. 325–340, 2018.

[13] S. Talukdar et al., "Land-use land-cover classification by machine learning classifiers for satellite observations—A review," *Remote Sens.*, vol. 12, no. 7, 2020, Art. no. 1135.

[14] E. Adam, O. Mutanga, J. Odindi, and E. M. Abdel-Rahman, "Land-use/cover classification in a heterogeneous coastal landscape using Rapid-Eye imagery: Evaluating the performance of random forest and support vector machines classifiers," *Int. J. Remote Sens.*, vol. 35, no. 10, pp. 3440–3458, 2014.

[15] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: An applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018.

[16] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.

[17] Z. Zhan, X. Zhang, Y. Liu, X. Sun, C. Pang, and C. Zhao, "Vegetation land use/land cover extraction from high-resolution satellite images based on adaptive context inference," *IEEE Access*, vol. 8, pp. 21036–21051, 2020.

[18] X. Deng, Y. Zhu, Y. Tian, and S. Newsam, "Scale aware adaptation for land-cover classification in remote sensing imagery," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 2159–2168.

[19] X. Zheng, L. Huan, G.-S. Xia, and J. Gong, "Parsing very high resolution urban scene images by learning deep ConvNets with edge-aware loss," *ISPRS J. Photogramm. Remote Sens.*, vol. 170, pp. 15–28, 2020.

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[21] B. Ayhan et al., "Vegetation detection using deep learning and conventional methods," *Remote Sens.*, vol. 12, no. 15, 2020, Art. no. 2502.

[22] X. Tang, H. Huang, Z. Xiong, X. Wang, and Z. Zhan, "An adaptive superpixels for vegetation detection on high resolution images based on MLP," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLIII-B3-2022, pp. 187–195, 2022.

[23] X. X. Zhu et al., "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[25] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[26] I. Demir et al., "DeepGlobe 2018: A challenge to parse the Earth through satellite images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 172–181.

[27] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1561–1570.

[28] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "High-resolution aerial image labeling with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7092–7103, Dec. 2017.

[29] W. Luo, W. Yang, X. Yu, Y. Wang, and K. Tan, "Lightweight convolutional neural network for high-spatial-resolution remote sensing scenes classification," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, 2020, pp. 104–108.

[30] J. Huang, L. Weng, B. Chen, and M. Xia, "DFFAN: Dual function feature aggregation network for semantic segmentation of land cover," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 3, 2021, Art. no. 125.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 31. Cham, Switzerland: Springer, 2015, pp. 234–241.

[2] More details related to RSIPAC can be found at http://rsipac.whu.edu.cn/subject_one.

[3] More details related to Ali Tianchi can be found at https://tianchi.aliyun.com/competition/entrance/531860/information.

[32] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection,” *ISPRS J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, 2018.

[33] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *Proc. Int. Conf. Legal Regulators Comput. Vis. Pattern Recognit.*, 2016, pp. 1–6.

[34] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528.

[35] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. Van-Den Hengel, “Effective semantic pixel labelling with convolutional networks and conditional random fields,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 36–43.

[36] Y. Yuan, X. Chen, and J. Wang, “Object-contextual representations for semantic segmentation,” in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 173–190.

[37] J. Dai et al., “Deformable convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[39] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[40] S. Zheng et al., “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6877–6886.

[41] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[42] P. Wang et al., “Understanding convolution for semantic segmentation,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1451–1460.

[43] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.

[44] E. Xie et al., “SegFormer: Simple and efficient design for semantic segmentation with transformers,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 12077–12090.

[45] L. Wang, R. Li, and C. Zhang, “UNetFormer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 190, pp. 196–214, 2022.

[46] L. Gao et al., “STransFuse: Fusing Swin transformer and convolutional neural network for remote sensing image semantic segmentation,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10990–11003, Oct. 2021.

[47] A. Tao, K. Sapra, and B. Catanzaro, “Hierarchical multi-scale attention for semantic segmentation,” 2020, *arXiv:2005.10821*.

[48] K. Sun et al., “High-resolution representations for labeling pixels and regions,” 2019, *arXiv:1904.04514*.

[49] T.-W. Ke et al., “Adaptive affinity fields for semantic segmentation,” 2018, *arXiv:1803.10335*.

[50] Y. Yuan, J. Xie, X. Chen, and J. Wang, “SegFix: Model-agnostic boundary refinement for segmentation,” 2020, *arXiv:2007.04269*.

[51] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Learning a discriminative feature network for semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1857–1866.

[52] Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, and C. Pan, “Semantic labeling in very high resolution images via a self-cascaded convolutional neural network,” *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 78–95, 2018.

[53] L. Ding et al., “Looking outside the window: Wide-context transformer for the semantic segmentation of high-resolution remote sensing images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Apr. 2022, Art. no. 4410313.

[54] H. Caesar, J. Uijlings, and V. Ferrari, “Region-based semantic segmentation with end-to-end training,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 381–397.

[55] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6488–6496.

[56] W. Ye, Wei Zhang, W. Lei, W. Zhang, X. Chen, and Y. Wang, “Remote sensing image instance segmentation network with transformer and multi-scale feature representation,” *Expert Syst. Appl.*, vol. 234, 2023, Art. no. 121007.

[57] J. Fu et al., “Dual attention network for scene segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3141–3149.

[58] F. Zhang, Y. Chen, and Z. Li, “ACFNet: Attentional class feature network for semantic segmentation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6797–6806.

[59] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[60] J. D. Wegner, F. Rottensteiner, M. Gerke, and G. Sohn, “The ISPRS labelling challenge,” *ISPRS Semantic Labeling Contest*, vol. 31, 2017, Art. no. 1.

[61] J. Wang et al., “LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation,” 2021, *arXiv:2110.08733*.

[62] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[63] O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[64] R. Li, S. Zheng, C. Zhang, C. Duan, L. Wang, and P. M. Atkinson, “ABCNet: Attentive bilateral contextual network for efficient semantic segmentation of fine-resolution remotely sensed imagery,” *ISPRS J. Photogramm. Remote Sens.*, vol. 181, pp. 84–98, 2021.

[65] A. G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, *arXiv:1704.04861*.

[66] C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[67] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1800–1807.

[68] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11966–11976.

Zongqian Zhan received the M.A.Eng. and Ph.D. degrees in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2003 and 2007, respectively.

He is currently a Full Professor with the School of Geodesy and Geomatics, Wuhan University. His research interests include camera calibration, close-range and UAV photogrammetry, oblique photogrammetry, deep learning, and remote sensing.

Zirou Xiong received the B.Eng. degree in surveying and mapping engineering in 2020 from Wuhan University, Wuhan, China, where he is currently working toward the M.Eng. degree in photogrammetry and remote sensing.

His research interests include semantic segmentation, object detection based on deep learning, photogrammetry, and remote sensing.

Xin Huang received the B.Eng. degree in surveying and mapping engineering in 2021 from Wuhan University, Wuhan, China, where she is currently working toward the M.Eng. degree in photogrammetry and remote sensing.

Her research interests include landcover classification based on deep learning, superpixel segmentation, photogrammetry, and remote sensing.

Chun Yang received the M.S. degree in communication technology from Ruhr University Bochum, Bochum, Germany, in 2011, and the Doctor of Engineering degree in photogrammetry and remote sensing from the Leibniz University Hanover, Hanover, Germany, in 2021.

In 2022, he co-founded Hangzhou SensingX Technology, Co., Ltd., which aims to provide full-cycle solutions for surveying urban underground space and digital twins for power plants.

Yi Liu received the B.S. degree in computer science and technology from the China University of Geosciences, Wuhan, China, in 2003, and the Ph.D. in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2009.

She is currently an Associate Professor with the School of Geodesy and Geomatics, Wuhan University. Her research interests include remote sensing image processing and deep learning.

Xin Wang received the B.Eng. and M.Eng. degrees in surveying and mapping from the School of Geodesy and Geomatics, Wuhan University, Wuhan, China, in 2013 and 2016, respectively, and the Doctor of Engineering degree in photogrammetry and remote sensing from Leibniz University Hannover, Hannover, Germany, in 2021.

He is currently an Assistant Professor with Wuhan University. His research interests include computer vision and deep leaning in applied photogrammetry.