DOI: 10.1111/cgf.14978

Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

MAPMaN: Multi-Stage U-Shaped Adaptive Pattern Matching Network for Semantic Segmentation of Remote Sensing Images

T. Hong[1], X. Ma[1], X. Wang[1], R. Che[1], C. Hu[1], T. Feng[†1], W. Zhang[1,2]

[1]School of Software Technology, Zhejiang University, Hangzhou, China
[2]Innovation Center of Yangtze River Delta, Zhejiang University, Jiaxing, China

Abstract

Remote sensing images (RSIs) often possess obvious background noises, exhibit a multi-scale phenomenon, and are characterized by complex scenes with ground objects in diversely spatial distribution pattern, bringing challenges to the corresponding semantic segmentation. CNN-based methods can hardly address the diverse spatial distributions of ground objects, especially their compositional relationships, while Vision Transformers (ViTs) introduce background noises and have a quadratic time complexity due to dense global matrix multiplications. In this paper, we introduce Adaptive Pattern Matching (APM), a lightweight method for long-range adaptive weight aggregation. Our APM obtains a set of pixels belonging to the same spatial distribution pattern of each pixel, and calculates the adaptive weights according to their compositional relationships. In addition, we design a tiny U-shaped network using the APM as a module to address the large variance of scales of ground objects in RSIs. This network is embedded after each stage in a backbone network to establish a Multi-stage U-shaped Adaptive Pattern Matching Network (MAPMaN), for nested multi-scale modeling of ground objects towards semantic segmentation of RSIs. Experiments on three datasets demonstrate that our MAPMaN can outperform the state-of-the-art methods in common metrics. The code can be available at `https://github.com/INiid/MAPMaN`.

CCS Concepts

• ***Computing methodologies*** → ***Neural networks; Image segmentation;***

1. Introduction

As a pixel-wise classification problem, semantic segmentation is a fundamental task in the field of computer vision, and significantly contributes to the analysis of remote sensing images (RSIs) and relevant applications, including land use classification [DDS22], agricultural production estimation [LYY*23], and environmental monitoring [YSL*20]. Compared to natural images, there are some considerable difficulties in RSIs as illustrated in Figure 1. Firstly, RSIs often possess obvious background noises, which are represented by disturbing elements (e.g., shadows, blurred edges, and ambiguous backgrounds) [WZC*20] [CNG*23] and unrelated remote ground objects [DTB21]; beside, RSIs tend to exhibit a multi-scale phenomenon that both inter-class and intra-class ground objects have significant scale variations; furthermore, ground objects in RSIs frequently exist within specific spatial distribution pattern (e.g., buildings are individually placed, whereas cars, roads, and houses jointly represent a neighborhood); even within the same pattern, ground objects may exhibit complex compositional relationships and geometric variations (e.g., buildings are likely to have different orientations and shapes).

Convolutional Neural Networks (CNNs) have been adopted as a preferred solution to semantic segmentation of images, including RSIs, given their outstanding capability to extract feature [PZY*17, ZSQ*17, ZLDB20]. However, most convolutional kernels used in the CNN-based semantic segmentation methods are actually a type of fixed filters, demonstrating two obvious limitations in the applications with RSIs: (1) The fixed weights cause sub-optimal performance for modeling compositional relationships with significant differences; (2) the fixed shapes lead to unsatisfactory adaptability to geometric variations. Recently, deformable convolutions [DQX*17, ZHLD19] have been proposed to address the issue on geometric variations, which support irregular sampling and emphasize the parts of interest within the sampled pixels by modulation scalars. However, they are yet to consider the compositional relationships among the sampled points.

To resolve the above-discussed limitations, recent studies have exploited the attention mechanism to establish Vision Transformers (ViTs) [DBK*20, WLZ*22]. The attention mechanism calculates the pixel-level similarity of the input image for weights aggregation, and thus establishes adaptive long-range dependence. However, it introduces excessive background noises and has a quadratic time complexity due to dense matrix multiplications, limiting the application scenarios in practice. Although local atten-

[†] Corresponding author (Email: t.feng@zju.edu.cn).

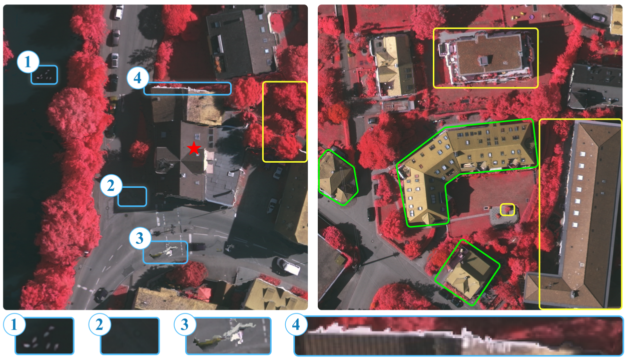


Figure 1: *Challenges in semantic segmentation of RSIs, including (1) background noises from disturbing elements and unrelated remote objects (in blue bounding boxes) for the reference pixel (marked by the red star); (2) multi-scale phenomenon in inter-class and intra-class objects (in yellow bounding boxes); and (3) objects in same pattern with complex compositional relationships and geometric variations (in green bounding boxes).*

tion [DTB21, LLC*21] and sparse attention [HWH*19, CGRS19] have attempted to resolve these issues, it is noteworthy that their capabilities to capture long-range dependence and adapt to geometric variations can be negatively impacted.

Recent studies have been devoted to investigating the use of convolutions as an alternative to attention. [HFD*21] finds that the superior performance of local attention comes from sparse connectivity, weight sharing and dynamic weight, which are also achievable by dynamic depth-wise convolutions. Meanwhile, ConvNeXt [LMW*22] is proposed to gradually transform a standard ResNet into the design of ViTs and obtain several critical components filling the performance gap between ViTs and CNNs. Furthermore, Conv2Former [HLCF22] is presented with a convolutional modulation operation for simplifying the attention mechanism, which achieves the outperformance over ViTs. These studies on the composition of the Transformer-based methods and the rethinking about CNN-based methods inspire us to devise a novel method for semantic segmentation of RSIs following the advantageous design of both ViTs and convolutions.

In this paper, we propose a compact and light-weight method for long-range adaptive weight aggregation, named Adaptive Pattern Matching (APM), where convolutions are exclusively adopted to explore the upper bound of their capability to extract features. Specifically, the APM comprises two processes, that is, Adaptive Pattern Sampling (APS) and Adaptive Feature Modulation (AFM). To begin with, the APS searches for a group of pixels belonging to the same spatial distribution pattern of ground objects for each pixel from its local features while identifying geometric variations of ground objects and filtering background noises; afterwards, the AFM performs point-wise convolutions on the group of pixels along the spatial and channel dimensions to recognize the compositional relationships of ground objects in the pattern. To address the multi-scale phenomenon of ground objects in RSIs, we design the UAPM, a tiny U-shaped network using the APM as a module. The UAPM is embedded in the backbone network to establish a Multi-stage U-shaped Adaptive Pattern Matching Network (MAPMaN) for semantic segmentation of RSIs, which enables the modeling of ground objects in a nested multi-scale way.

The contributions of this work can be summarized as follows:

- We design a novel long-range adaptive weight aggregation method to recognize spatial distribution patterns of ground objects in RSIs.
- We present a nested multi-scale architecture that embeds a tiny U-shaped network after each stage of the backbone.
- We propose a Multi-Stage U-Shaped Adaptive Pattern Matching Network for semantic segmentation of RSIs, suggesting the superiority to other state-of-the-art methods on three datasets in our experiments.

2. Related work

CNN-based methods. Drastically increasing computational resources and data have driven researchers to propose numerous CNN-based methods for computer vision tasks. To solve the problem of semantic segmentation, which is a dense prediction task, [LSD15] proposes Fully Convolutional Network (FCN) enabling the end-to-end training for the first time. Most of the subsequent methods adapt an encoder-decoder architecture, where the decoder incorporates various feature enhancement techniques, such as spatial pyramid modeling [ZSQ*17, CZP*18, YWP*18], symmetric structure-based multi-scale modeling [RFB15, LMSR17, KGHD19], and attention mechanism [WGGH18, WPLK18, HWH*19].

Recent studies have presented further advancements on convolutional kernels. DCNv1 [DQX*17] employs irregular sampling to handle visual object deformations; DCNv2 [ZHLD19] instigates modulation scalars to emphasize the parts of interest within the sampled pixels; DCNv3 [WDC*23] optimizes the modulation scalars with a multi-head mechanism. However, these methods still use fixed weights and can hardly identify the various compositional relationships among the sampled pixels.

Vision Transformers. Given to the excellent performance for NLP tasks, Transformer [VSP*17] has been widely adopted to form a series of Vision Transformers (ViTs) towards computer vision tasks. In particular, the vanilla ViT [DBK*20] slices the input image into fixed-size tokens and then projects them as patches, which is the first purely Transformer-based method achieving impressive results. Besides, Swim Transformer [LLC*21] adopts the attention mechanism with shifted windows to introduce the convolution-like inductive bias while reducing the computational cost. Pyramid Vision Transformer (PVT) [WXL*21] employs a pyramid structure similar to CNNs and brings suitability for dense prediction tasks, including semantic segmentation. In addition, a series of decoders [ZZT*20, ZTT*22, SGLS21] have emerged to enhance dense prediction via multi-scale design and the revision of the attention mechanism for ViTs.

Composition of ViTs. More recently, several studies attempt to replace the token mixers represented by the attention mechanism

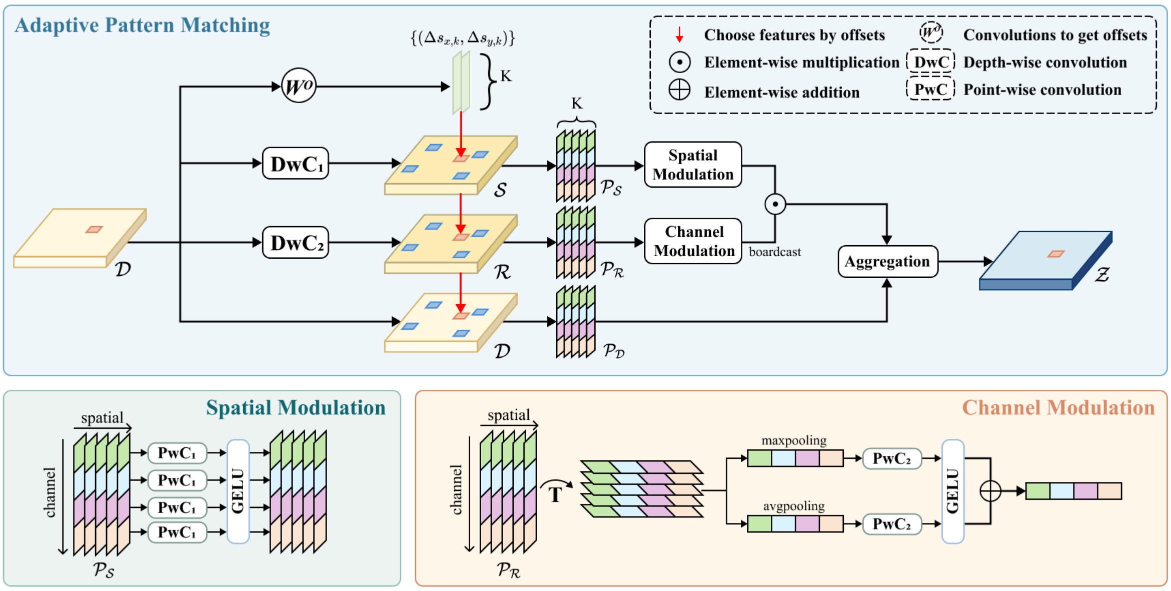

Adaptive Pattern Matching
$\{(\Delta s_{x,k}, \Delta s_{y,k})\}$
K
W^o
DwC₁
DwC₂
$\mathcal{S}$
$\mathcal{R}$
$\mathcal{D}$
K
$\mathcal{P}_\mathcal{S}$
$\mathcal{P}_\mathcal{R}$
$\mathcal{P}_\mathcal{D}$
Spatial Modulation
Channel Modulation
boardcast
Aggregation
$\mathcal{Z}$
Choose features by offsets
W^o Convolutions to get offsets
⊙ Element-wise multiplication
DwC Depth-wise convolution
⊕ Element-wise addition
PwC Point-wise convolution
Spatial Modulation
spatial
channel
PwC₁
PwC₁
PwC₁
PwC₁
GELU
$\mathcal{P}_\mathcal{S}$
Channel Modulation
spatial
channel
T
maxpooling
avgpooling
PwC₂
PwC₂
GELU
$\mathcal{P}_\mathcal{R}$

Figure 2: *Pipeline of our Adaptive Pattern Matching (APM) method, which consists of depth-wise convolutions and point-wise convolutions to enable the light-weight design. Given an input feature map $\mathcal{D} \in \mathbb{R}^{H\times W\times C}$, two depth-wise convolutions are applied to obtain $\mathcal{S}$ and $\mathcal{R}$. For a pixel d_i in $\mathcal{D}$, the pattern's information $\mathcal{P}_\mathcal{S}$, $\mathcal{P}_\mathcal{R}$, and $\mathcal{P}_\mathcal{D} \in \mathbb{R}^{C\times K}$ are sampled based on offsets $\{(\Delta s_{x,k}, \Delta s_{y,k})\} \in \mathbb{R}^{K\times 2}$. Afterwards, spatial and channel modulations are conducted in parallel to obtain adaptive weights. Finally, adaptive weights are used to conduct the aggregation for the enhanced feature vector z_i.*

in ViTs with Multi-Layer Perceptrons (MLPs) [THK*21] or its variants [HJY*22]. These replacements achieve competitive performance, which, however, have raised the question on the key to impacting the performance of ViTs. In addition, [LMW*22] devises ConvNeXt to gradually transform the standard ResNet into the design of ViTs, which discovers several critical components enabling the purely convolution-based model to outperform ViTs. [HFD*21] observes the similarities between depth-wise convolution and local attention regarding sparse connectivity and weight sharing; considerable performance is obtained by replacing the local attention in Swim Transformer with the dynamic depth-wise convolution. [HLCF22] presents Conv2Former that exploits the convolutional modulation operation and models the attention via calculating adaptive weights with large-kernel depth-wise convolutions, exhibiting performance comparable to ViTs. These methods inspire us to decompose the attention mechanism and implement its key elements using convolutions. Therefore, we devise the APM that can outperform the attention mechanism and thus increase the upper bound of convolutions' capability to extract features.

3. Proposed method

The attention mechanism can dynamically select and aggregate information based the relationships among sequences. In CNN-based methods, the attention mechanism is employed to capture long-range dependence and model context information. In ViTs, the attention mechanism serves as the key component. We formulate the general form of the attention mechanism in this paper as follows. Given queries $\mathcal{Q} \in \mathbb{R}^{N_q\times C}$, keys $\mathcal{K} \in \mathbb{R}^{N_{kv}\times C}$, and values $\mathcal{V} \in \mathbb{R}^{N_{kv}\times C}$, the attention mechanism calculates the similarities between each query q_i and all keys using dot product. It aggregates the corresponding values weighted by these similarities as follows:

$$Att(\mathcal{Q},\mathcal{K},\mathcal{V}) = \sum_{j=1}^{N_{kv}} \frac{\exp(q_i \cdot k_j)}{\sum_{j=1}^{N_{kv}} \exp(q_i \cdot k_j)} \cdot v_j, \forall i \in \{1,2,\dots,N_q\}. \quad (1)$$

In practice, the multi-head self-attention (MHSA) module has been more widely adopted given its improved performance and capability of representation. Specifically, *self-attention* refers to that queries, keys, and values are mapped from the same input flattened feature map $\mathcal{D} \in \mathbb{R}^{N\times\hat{C}}$, where N denotes the number of pixels in $\mathcal{D}$; *multi-head* refers to that $\mathcal{D}$ is mapped into h heads using h separate feature mapping sets and the final result is the concatenation of all heads', which can be defined as follows:

$$head_i = \mathrm{Att}(W_i^q\mathcal{D}, W_i^k\mathcal{D}, W_i^v\mathcal{D}), \quad (2)$$

$$MHSA(\mathcal{D}) = Concat(head_0, head_1, ..., head_{h-1})W^o, \quad (3)$$

where $head_i \in \mathbb{R}^{N\times\frac{C}{h}}$ represents the output of the i-th head. The feature mappings $W_i^q, W_i^k, W_i^v \in \mathbb{R}^{\frac{C}{h}\times\hat{C}}$ embed the input into different feature spaces, while $W^o \in \mathbb{R}^{C\times C}$ combines the results from the multi-heads.

In retrospect, the MHSA module is characterized by (1) adopting multiple feature mapping sets so that heads can capture different patterns of relationships in the input; and (2) allowing the input sequences to interact with each other and assigning weights based

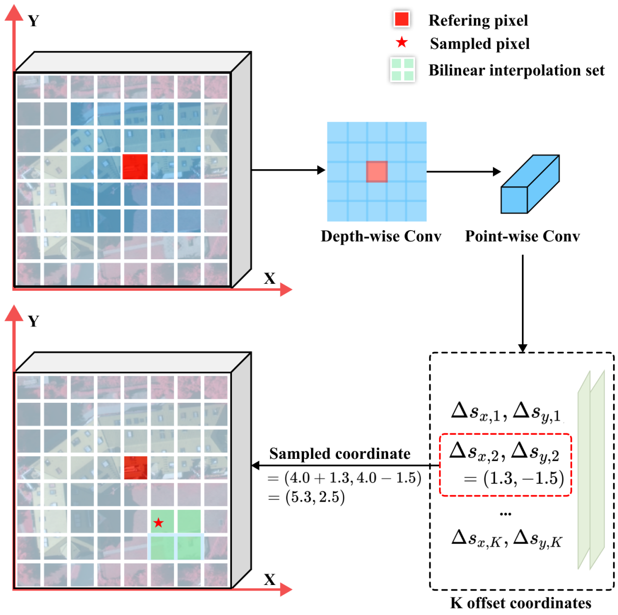


Figure 3: *Example of Adaptive Pattern Sample. Firstly K offset coordinates are obtained by successive depth-wise and point-wise convolutions. Taking the offset coordinates (1.3,-1.5) as an example, the coordinates of the sampled pixel are obtained as (4.0+1.3, 4.0-1.5) = (5.3, 2.5), and finally the value here is obtained by bilinear interpolation for the four pixels in the green box.*

on their correlations to efficiently process the various spatial distributions. However, obvious drawbacks still exist with the MHSA module, among which the most influential one is the establishment of dependence among all sequences. For RSIs with complex backgrounds, such dependence is very likely to introduce excessive noises and cause performance degradation. In addition, performing n queries for n key-value pairs leads to the $O(n^2)$ time complexity, especially when employing the multi-head strategy, which requires more significant computation. Several studies adapt sparse attention via local-block computing or location-specific pixel selection to reduce computation, but are yet to address the problem of indiscriminate dependence establishment.

3.1. Adaptive Pattern Matching

For a pixel d_i, the adaptive weight aggregation method represented by the attention mechanism can be formulated as follows:

$$\mathbf{z}_i = \sum_{d_j \in region(d_i)} \Phi\left(f_q\left(d_i\right), f_k\left(d_j\right)\right) f_v\left(d_j\right), \tag{4}$$

where $region(d_i)$ denotes the entire feature map in the global attention or specific pixels at certain locations in the sparse attention, Φ represents the dot product operation, and f_q, f_k, and f_v refer to feature mappings.

Our Adaptive Pattern Matching (APM) method, which is composed of Adaptive Pattern Sampling (APS) and Adaptive Feature Modulation (AFM), improves the above formulation of adaptive weight aggregation as follows:

$$\mathbf{z}_i = \sum_{\mathcal{P}_j \in pattern(d_i)} \Phi\left(f_s\left(\mathcal{P}_j\right), f_c\left(\mathcal{P}_j\right)\right) \mathcal{P}_j. \tag{5}$$

Specifically, the APS differs from the indiscriminate sampling adopted by the self-attention mechanism, and searches for a set of pixels $\mathcal{P} = pattern(d_i)$ belonging to the same spatial distribution pattern for each pixel according to its local features, which can identify geometric variations of ground objects and ignore background noises; the AFM includes two simple but effective methods to calculate adaptive weights $\Phi\left(f_s\left(\mathcal{P}_j\right), f_c\left(\mathcal{P}_j\right)\right)$ regarding spatial and channel dimensions, so as to identify the compositional relationships among the pixels in the same spatial distribution pattern.

Adaptive Pattern Sampling. As shown in Figure 3, for a feature map $\mathcal{D} \in \mathbb{R}^{H \times W \times C}$, where H, W, and C denote $\mathcal{D}$'s height, weight, and number of channels, two consecutive convolutions are adopted to compute 2-dimensional offsets $(\Delta S_X, \Delta S_Y)$ for all pixels as follows:

$$(\Delta S_X, \Delta S_Y) = \psi(W_0^o(W_1^o \mathcal{D})), \tag{6}$$

where $S_X, S_Y \in \mathbb{R}^{H \times W \times K}$ respectively represent the horizontal and vertical offsets, $W_0^o \in \mathbb{R}^{2K \times C}$ denotes a point-wise convolution with kernel size of 1×1 to change the number of channels, $W_1^o \in \mathbb{R}^{C \times C}$ denotes a depth-wise convolution with kernel size of 5×5 to collect local information, ψ refers to a splitting operation that divides a feature map into two parts along the channel dimension, and K refers to the number of sampled pixels.

For a reference pixel r at position (p_x, p_y), the k-th sampled pixel $r\prime_k$ corresponds to the offset $(\Delta s_{x,k}, \Delta s_{y,k})$ as follows:

$$r\prime_k = f(p_{x,k}, p_{y,k}) = f(p_x + \Delta s_{x,k}, p_y + \Delta s_{y,k}), \tag{7}$$

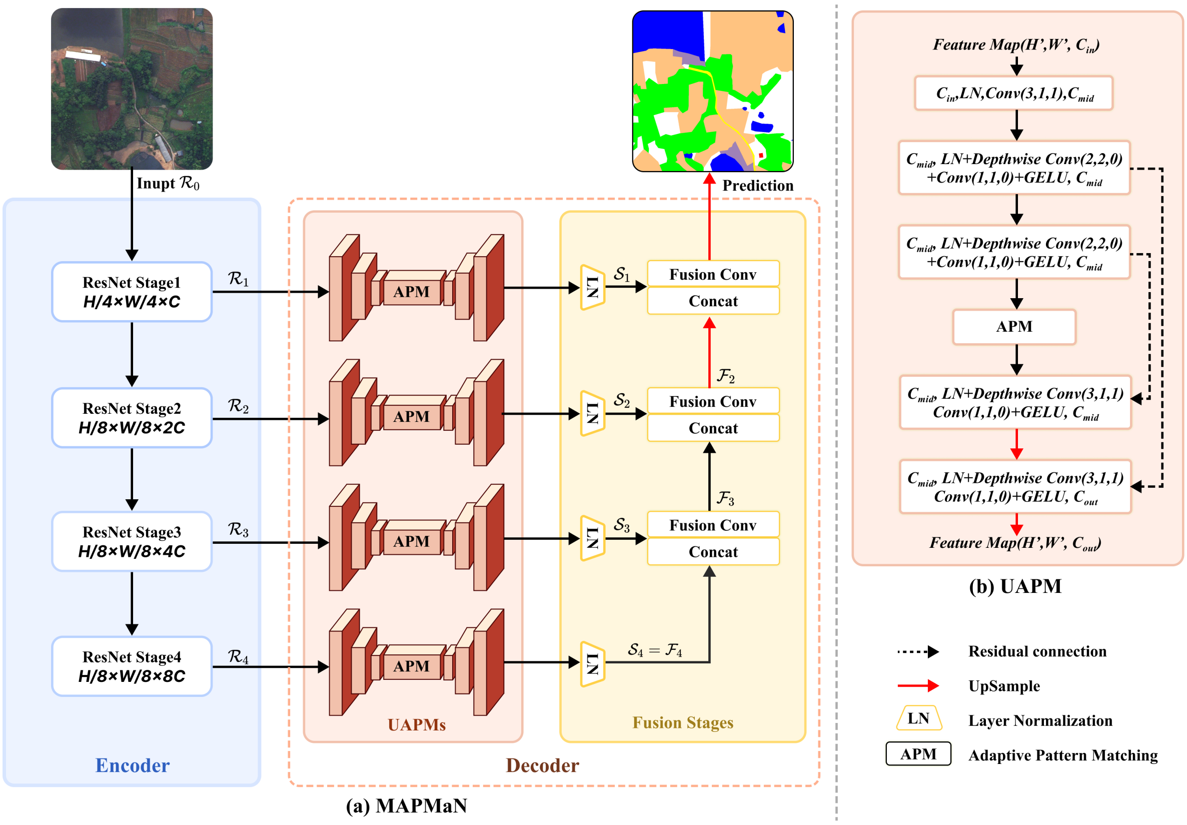
where $f(\cdot)$ denotes a function that obtains the corresponding value from the feature map based on a pair of coordinates.

Since $(\Delta s_{x,k}, \Delta s_{y,k})$ are fractions, we use a linear kernel to aggregate the pixels near $r\prime_k$ as:

$$r\prime_k = \sum_{(p_{x,t}, p_{y,t}) \in \tau} g(p_{x,t}, p_{y,t}) f(p_{x,t}, p_{y,t}), \tag{8}$$

where τ denotes a set of the four pixels closest to (p_{xk}, p_{yk}), and $g(\cdot, \cdot)$ represents a kernel that is implemented as bilinear interpolation. All sampled pixels are concatenated along the spatial dimension to obtain the pattern $\mathcal{P} \in \mathbb{R}^{K \times C}$.

Adaptive Feature Modulation. Inspired by [HLCF22] and [HFD*21], we adapt two parallel depth-wise convolutions to the input feature map $\mathcal{D} \in \mathbb{R}^{H \times W \times C}$ to obtain the preliminary modulation weights $\mathcal{S} \in \mathbb{R}^{H \times W \times C}$ and $\mathcal{R} \in \mathbb{R}^{H \times W \times C}$, as shown in Figure 2. This allows each channel to be approximated as a *head* in the MHSA module, which can capture a specific compositional relationship to enhance the APM's capability of representation. Compared to [HLCF22] using a convolutional kernel of size 11×11, our method requires a smaller kernel of size 5×5 that greatly reduces the number of parameters.



(a) MAPMaN

(b) UAPM

Figure 4: *Architecture of the proposed MAPMaN and UAPM. (a) Our MAPMaN follows the encoder-decoder architecture, whose encoder is a ResNet50 with dilated convolutions and decoder consists of UAPMs and fusion stages. (b) Our UAPM only adopts depth-wise and point-wise convolutions to enable a light-weight design that significantly reduces memory consumption and computational complexity while adding limited parameters.*

For a reference pixel r at position (p_x, p_y), we perform APS on $\mathcal{S}$, $\mathcal{R}$, and $\mathcal{D}$ to respectively obtain the pattern's information $\mathcal{P}_{\mathcal{S}}$, $\mathcal{P}_{\mathcal{R}}$, and $\mathcal{P}_{\mathcal{D}} \in \mathbb{R}^{C\times K}$ as follows:

$$\mathcal{P}_{\mathcal{S}} = \xi(\mathcal{S}, \{(\Delta s_{x,k}, \Delta s_{y,k})\}), \tag{9}$$

$$\mathcal{P}_{\mathcal{R}} = \xi(\mathcal{R}, \{(\Delta s_{x,k}, \Delta s_{y,k})\}), \tag{10}$$

$$\mathcal{P}_{\mathcal{D}} = \xi(\mathcal{D}, \{(\Delta s_{x,k}, \Delta s_{y,k})\}), \tag{11}$$

where K represents the number of sampled pixels in the pattern, and $\xi(X, \{(\Delta s_{x,k}, \Delta s_{y,k})\})$ denotes a function that selects the pixels from the feature map X based on the 2-dimensional offsets $\{(\Delta s_{x,k}, \Delta s_{y,k})\} \in \mathbb{R}^{K\times 2}$.

For $\mathcal{P}_{\mathcal{S}}$, adaptive weights are calculated along the spatial dimension according to the compositional relationship of the pattern pixels as follows:

$$f_s(\mathcal{P}_{\mathcal{S}}) = W_0^s(W_1^s \mathcal{P}_{\mathcal{S}}), \tag{12}$$

where $W_0^s \in \mathbb{R}^{K\times(K/\varepsilon)}, W_1^s \in \mathbb{R}^{(K/\varepsilon)\times K}$ are two simple point-wise convolutions along the spatial dimension, ε denotes the reduction ratio. It is noteworthy that the adaptive weights for each channel are calculated separately according to its own content following the style of depth-wise convolution. This design allows each channel to focus on an unique compositional relationship.

For $\mathcal{P}_{\mathcal{R}}$, adaptive weights are calculated along the channel dimension to determine the importance of the compositional relationship represented by each channel as follows:

$$\begin{aligned} f_c\left(\mathcal{P}_{\mathcal{R}}\right) = \sigma((&W_0^c(W_1^c AvgPool(\mathcal{P}_{\mathcal{R}}))) \\ &+ (W_0^c(W_1^c MaxPool(\mathcal{P}_{\mathcal{R}})))), \end{aligned} \tag{13}$$

where σ represents the Sigmoid function, $W_0^c \in \mathbb{R}^{C\times(C/\varepsilon)}, W_1^c \in \mathbb{R}^{(C/\varepsilon)\times C}$ are two point-wise convolutions along the channel dimension. The pairing relationships between pixels in different patterns may be different. For example, buildings and roads constitute neighborhood, and shrubs and roads form gardens. Therefore, we introduce two parallel pooling operations to capture pattern-related global context information.

Different from the dot product used in Equation 4, we employ the Hadamard product as follows:

$$\Phi\left(f_s(\mathcal{P}_{\mathcal{S},j}), f_c(\mathcal{P}_{\mathcal{R},j})\right) = f_s(\mathcal{P}_{\mathcal{S},j}) \cdot f_c(\mathcal{P}_{\mathcal{R},j}), \tag{14}$$

which outputs the adaptive weights for all pixels within the pattern.

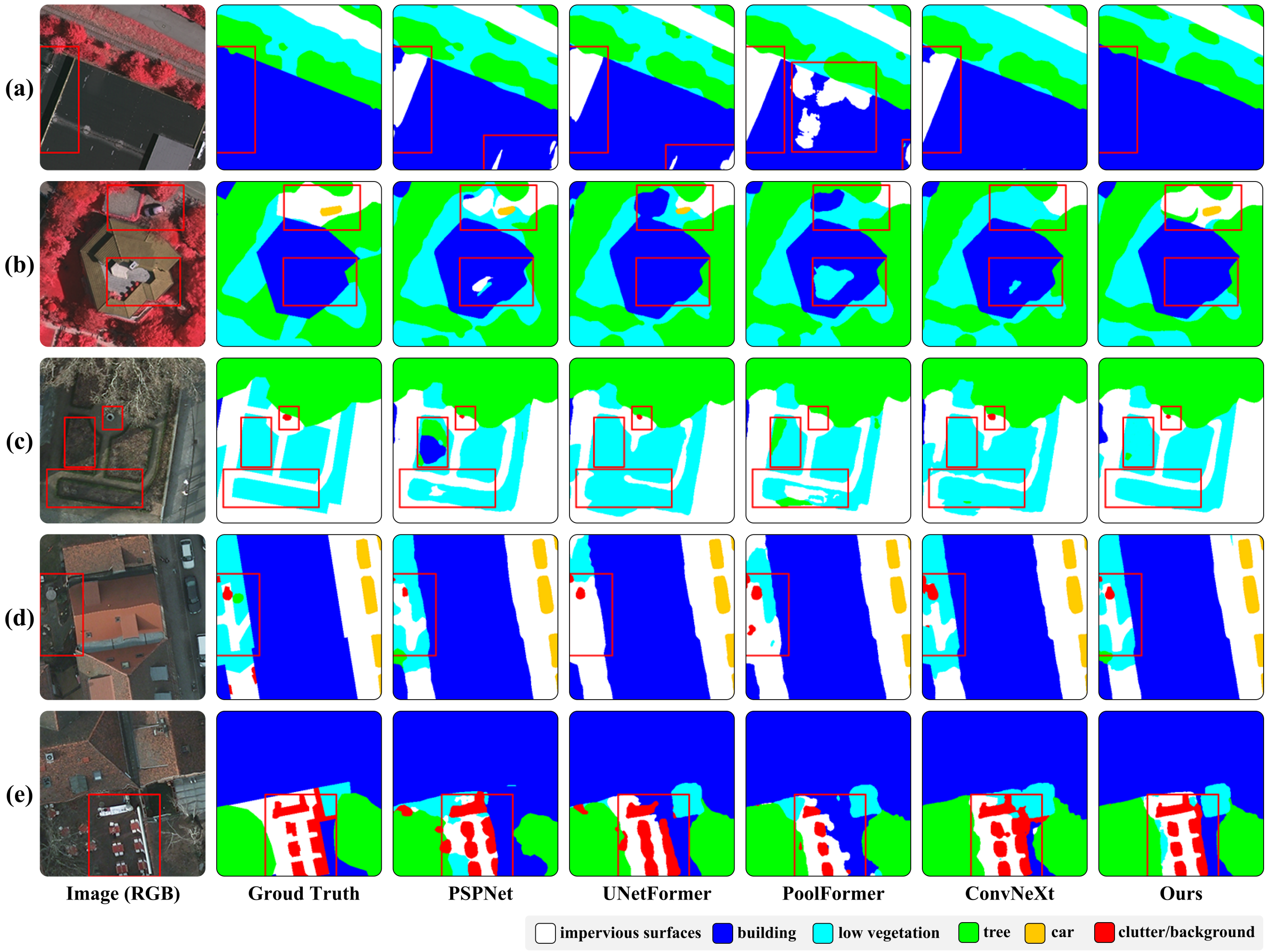


Figure 5: *Visualization of example segmentation maps output from our MAPMaN and other state-of-the-art methods for comparison. (a) and (b) are from ISPRS Vaihingen test set; (c), (d), and (e) are from ISPRS Potsdam test set. The regions in red boxes refer to the areas that are prone to confusion or ambiguity.*

Finally, the adaptive weights are element-wise multiplied with $\mathcal{P}_{\mathcal{D}}$ and summed along the spatial dimension to output the feature vector that has been enhanced via adaptive pattern matching.

3.2. Tiny U-shaped network with APM

To extract features in a multi-scale manner, our UAPM, a tiny U-shaped network, incorporates the above-discussed APM as a module at the bottom level. As shown in Figure 4(b), our UAMP adopts depth-wise convolutions with kernel size of 2 and stride of 2, which facilitates the learning of local information prior to downsampling. Besides, a residual connection is employed between the feature maps of the same size to reduce the complexity, instead of concatenation and convolution for fusion. It is noteworthy that our UAPM only uses depth-wise and point-wise convolutions to ensure the light-weight design.

Rather than being used solely for multi-scale modeling, our UAPM can be embedded into an outer network to provide the following benefits: (1) It significantly reduces the memory consumption and the computational complexity required by the bottom-level feature enhancement (e.g., APM); (2) it increases the receptive field and thus enables the bottom-level feature enhancement module to capture long-range dependence more effectively; and (3) it effectively reuses the spatial information of the feature maps using residual connections, so as to enhance the modeling of fine details.

3.3. Multi-Stage U-Shaped APM Network

To validate the UAMP's capability to be embedded in an outer network, we devise MAPMaN, a Multi-Stage U-Shaped Adaptive Pattern Matching Network that that follows the encoder-decoder architecture, as depicted in Figure 4(a). Our MAPMaN takes as input an

Table 1: *Comparison with the state-of-the-art methods on LoveDA, ISPRS Vaihingen and ISPRS Potsdam test sets. Highest scores are in bold. All scores are reported in percentage.*

Method	LoveDA								Vaihingen			Potsdam		
	back	buil	road	water	barren	forest	agri	mIoU	AF	mIoU	OA	AF	mIoU	OA
MANet [LZZ*22]	38.7	51.7	42.6	72.0	15.3	42.1	57.7	45.7	90.41	82.71	90.96	92.90	86.95	91.32
Segmenter [SGLS21]	38.0	50.7	48.7	77.4	13.3	43.5	58.2	47.1	88.23	79.44	89.93	92.27	86.48	91.04
LANet [DTB21]	40.0	50.6	51.1	78.0	13.0	43.2	56.9	47.6	88.09	79.28	89.83	91.95	85.15	90.84
DeepLabv3+ [CZP*18]	43.0	50.9	52.0	74.4	10.4	44.2	58.5	47.6	86.77	77.13	89.12	90.86	84.24	89.18
FarSeg [ZZWM20]	43.1	51.5	53.9	76.6	9.8	43.3	58.9	48.2	87.88	79.14	89.57	91.21	84.36	89.87
Semantic FPN [KGHD19]	42.9	51.5	53.4	74.7	11.2	44.6	58.7	48.2	87.58	77.94	89.86	91.53	84.57	90.16
PSPNet [ZSQ*17]	44.4	52.1	53.5	76.5	9.7	44.1	57.9	48.3	86.47	76.78	89.36	89.98	81.99	90.14
FLANet [SLL*22]	44.6	51.8	53.0	74.1	15.8	45.8	57.6	49.0	87.44	78.08	89.60	93.12	87.50	91.87
OCRNet [YCW20]	44.2	55.1	53.5	74.3	18.5	43.0	60.5	49.9	89.22	81.71	90.47	92.25	86.14	90.03
SwimUperNet [LLC*21]	43.3	54.3	54.3	78.7	14.9	45.3	59.6	50.0	89.90	81.80	91.00	92.24	86.37	90.98
DANet [FLT*19]	44.8	55.5	53.0	75.5	17.6	45.1	60.1	50.2	86.88	77.32	89.47	89.60	81.40	89.73
ConvNeXt [LMW*22]	46.9	53.5	56.8	76.1	15.9	47.5	61.8	51.2	90.50	82.87	91.36	93.03	87.17	91.66
ISNet [JLCY21]	44.4	57.4	**58.0**	77.5	**21.8**	43.9	60.6	51.9	90.19	82.36	90.52	92.67	86.58	91.27
UNetFormer [WLZ*22]	44.7	58.8	54.9	79.6	20.1	46.0	62.5	52.4	90.40	82.70	91.00	92.80	86.80	91.30
BiFormer [ZWK*23]	43.6	55.3	55.9	79.5	16.9	45.4	61.5	51.2	89.65	81.50	90.63	91.47	84.51	90.17
PoolFormer [YLZ*22]	45.8	57.1	53.3	80.2	19.8	46.1	64.5	52.4	89.59	81.35	90.30	92.62	86.45	91.12
MAPMaN (Ours)	**47.3**	**59.5**	56.7	**80.5**	20.3	**48.6**	**65.0**	**54.0**	**91.54**	**84.64**	**91.79**	**93.43**	**87.88**	**91.98**

RGB remote sensing image $\mathcal{R}_0$ and aims to output the corresponding semantic segmentation map. The encoder is based on ResNet50 with dilated convolutions [CZP*18], and the decoder comprises the UAPM, layer normalization, and a sequence of fusion convolutions following each stage of the encoder.

Specifically, the ResNet block at the i-th stage extracts a feature map $\mathcal{R}_i$ from $\mathcal{R}_{i-1}$. The corresponding UAPM then conducts multi-scale modeling on $\mathcal{R}_i$, followed by layer normalization, to obtain an intermediate feature map $\mathcal{S}_i$. Except for $\mathcal{S}_4$ that is equivalent to $\mathcal{F}_4$, each $\mathcal{S}_i$ is concatenated with $\mathcal{F}_{i+1}$ along the channel dimension, followed by fusion convolution, to generate a fused feature map $\mathcal{F}_i$. In particular, the bottleneck-like fusion convolution in our MAPMaN is light-weight by using depth-wise convolution. Finally, the fuse feature map of the 1-st stage is upsampled to the size of the input image to reach the output segmentation map.

Table 2: *Comparison with selected context aggregation modules on* Params *(M) and* FLOPs *(G). The size of feature map used for calculating* Params *and* FLOPs *is* 2048 × 128 × 128.

Method	Params (M)	FLOPs (G)
DCNv3 Block [WDC*23]	42.8	702.2
PPM [ZSQ*17]	23.1	309.5
ASPP [CZP*18]	15.1	503.0
DAB [FLT*19]	23.9	392.2
ConvMod Block [HLCF22]	21.3	348.3
OCR [YCW20]	10.5	354.0
PAM+AEM [DTB21]	10.4	157.6
ILCM+SLCM [JLCY21]	11.0	180.6
UAPM (Ours)	**3.3**	**26.4**

4. Experiments

4.1. Datasets and metrics

We conduct the experiments on three RSI datasets and adopt common metrics to evaluate the performances of our MAPMaN and other methods for comparison.

ISPRS Vaihingen. The ISPRS Vaihingen dataset [RSJ*21] consists of 33 TOP image tiles and digital surface models (DSMs) with a ground sampling distance (GSD) of 0.09 m, ranging from 1996 × 1995 to 3816 × 2550 in pixels regarding the size of image. In our experiments, we only use the TOP image tiles that have three multi-spectral bands: near-infrared, red, and green. The dataset involves 6 land cover categories (i.e., impervious surfaces, buildings, low vegetation, trees, cars, and clutter/background). We select the images with IDs 1, 5, 7, 11, 13, 15, 17, 21, 23, 26, 28, 32, 34, and 37 for training, and the image with id 30 for validation, and the rest images for testing. All images are cropped into patches of size 1024 × 1024 in pixels.

ISPRS Potsdam. The ISPRS Potsdam dataset [RSJ*21] is a 2-dimensional semantic labeling dataset that contains 38 very fine spatial resolution (GSD 0.05 m) TOP image tiles, each of which is of size 6000 × 6000 in pixels. The dataset involves the same categories as the ISPRS Vaihingen dataset, but has three different types of images: IRRG, RGB, and RGBIR. In our experiments, we only use the RGB images. We select images with IDs 2_13, 2_14, 3_13, 3_14, 4_13, 4_14, 4_15, 5_13, 5_14, 5_15, 6_13, 6_14, 6_15, and 7_13 for testing, the image with ID 2_10 for validation, and the rest images for training. All images are cropped into patches of size 1024 × 1024 in pixels.

LoveDA. The LoveDA dataset [WZM*21] contains 5987 high spatial resolution images (GSD 0.30 m) of size 1024 × 1024 in

Table 3: *Ablation study on the influence of each part in our MAPMaN on LoveDA and ISPRS Potsdam test sets. Highest scores are in bold. All scores are reported in percentage.*

Model	LoveDA	Potsdam		
	mIoU	AF	mIoU	OA
Model 1	50.8	91.87	85.19	90.30
Model 2	52.4	92.60	86.43	90.94
Model 3	53.1	92.87	86.91	91.23
Model 4	53.6	93.15	87.43	91.63
MAPMaN (Ours)	**54.0**	**93.43**	**87.88**	**91.98**

pixels, and involves 7 land cover categories (i.e., building, road, water, barren, forest, agriculture and background). Compared to ISPRS Vaihingen and Potsdam datasets, it is significantly larger and covers two domains (i.e., urban and rural areas), posing considerable challenges due to the presence of multi-scale objects, complex backgrounds, and inconsistent class distributions. We use 2522 images for training, 1669 images for validation and the other 1796 images for testing.

Metrics. We adopt *overall accuracy* (OA), *average F1 score per class* (AF), and mean intersection over union (mIoU) as evaluation metrics. Since the LoveDA dataset is tested online, we only employ the IoU of each class and mIoU for evaluation. The mIoU is calculated as follows:

$$\mathrm{IoU}_i = \frac{p_{ii}}{\sum_{j=1}^{N} p_{ij} + \sum_{j=1}^{N} p_{ji} - p_{ii}}, \tag{15}$$

$$\mathrm{mIoU} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{IoU}_i, \tag{16}$$

where p_{ij} represents the number of pixels that have true value i and predicted value j, p_{ii} refers to the true positive count, p_{ij} and p_{ji} respectively denote false positives and false negatives, and N is the number of classes.

4.2. Implement details

We implement our MAPMaN using Python and PyTorch on a workstation with two NVIDIA GTX A6000 graphics cards (96 GB GPU memory in total). For training, the AdamW optimizer with weight decay of 1e-4 is adopted, and the initial learning rate is set to 1e-4 with the poly decay strategy. The training epochs for Vaihingen, Potsdam, and LoveDA datasets are 250, 100, and 50, respectively, with a batch size of 4 for all. Following previous studies [DTB21, LZZ*22], we employ data augmentation methods, such as random scaling (0.5, 0.75, 1.0, 1.25, 1.5), random vertical flipping, random horizontal flipping, and random rotation for training. The augmented images are randomly cropped into patches of size 512 × 512 in pixels. During the inference process, data augmentation techniques such as random flipping and multi-scale prediction are used.

To avoid being impacted by other advanced architectures (e.g., ViTs), the most basic ResNet50 with dilated convolutions [CZP*18] is adopted as the backbone. After each stage of ResNet50, we embed a UAPM for feature augmentation. The optimal configuration has downsampling rates of (4, 4, 4, 4) for UAPM and the numbers of sampled pixels of (16, 9, 5, 5) for APM at each stage (see Section 4.4). In addition, we employ GELU [HG16] activation function and layer normalization to expedite the convergence, and use dropout with a ratio of 0.10 to prevent overfitting.

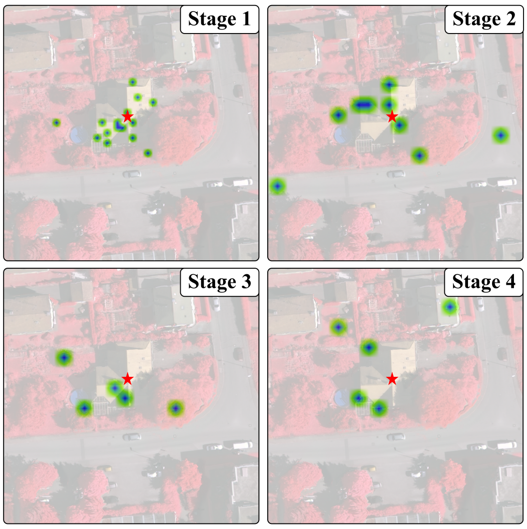


Figure 6: *Visualization of the pixels output from APM at each stage. Red star marks the query pixel.*

4.3. Comparison results

We compare our MAPMaN with a series of representative semantic segmentation methods. As shown in Table 5, our MAPMaN significantly outperforms all other methods on three datasets. Specifically, our MAPMaN obtains an increase by 1.6% in mIoU compared to PoolFormer and UNetFormer on LoveDA test set. As to ConvNeXt, the proposed method achieves an increase by 1.8% on ISPRS Vaihingen test set and by 0.7% on ISPRS Potsdam test set in mIoU. The outstanding performances on three datasets suggest that our MAPMaN has the strong capability of generalization.

Figure 5 shows the visualization of example results output from PSPNet, UNetFormer, PoolFormer, ConvNeXt and our MAPMaN. It is observable that our MAPMaN performs satisfactorily in recognizing the overall framework and edges of ground objects with complex backgrounds and identifying the shapes of complex specific compositions, such as gardens consisting of vegetation and trails. For incorporating UAPMs, the proposed method is highly capable to recognize the details in multi-scale objects, especially those small-scale ones.

In addition, we compare several context aggregation modules with our UPAM regarding the number of parameters (Params) measured in million (M) and the number of floating-point operations per second (FLOPs) measured in giga (G). Because of only using

Table 4: *Comparison of selected long-range weight aggregation methods. The size of feature map used for calculating* Params *and* FLOPs *is* $512 \times 32 \times 32$ *as same as that of the input to the APM module at the first stage.*

Method	Params (M)	FLOPs (G)	LoveDA	Potsdam		
			mIoU	AF	mIoU	OA
Global MHSA [DBK*20]	2.6	7.0	52.6	92.90	86.95	91.41
DCNv3 Layer [WDC*23]	5.2	5.3	52.9	93.06	87.24	91.63
CovMod Layer [HLCF22]	2.9	3.0	53.2	93.08	87.21	91.63
Deformable MHSA [ZSL*20]	2.6	2.7	53.4	92.97	87.17	91.59
APM (ours)	**1.1**	**2.0**	**54.0**	**93.43**	**87.88**	**91.98**

Table 5: *Ablation study on the influence of the downsampling factor and the number of sampled pixels of UAPM at each stage on LoveDA test set.*

Downsampling Factors	Pixel Combination	Numbers of Pixels	mIoU
(2, 2, 2, 2)	medium	(16, 9, 9, 9)	51.9
(4, 4, 4, 4)	small	(9, 5, 3, 3)	52.9
		(9, 5, 5, 5)	53.5
		(9, 9, 9, 9)	53.1
	medium	(16, 9, 5, 5)	**54.0**
		(16, 9, 9, 9)	53.2
		(16, 16, 16, 16)	53.1
	large	(25, 16, 9, 9)	53.0
		(25, 16, 16, 16)	52.8
		(25, 25, 25, 25)	52.5
(8, 8, 8, 8)	medium	(16, 9, 9, 9)	52.9
(16, 16, 16, 16)	medium	(16, 9, 9, 9)	50.8

depth-wise and point-wise convolutions, our UAPM requires significantly less Params and FLOPs compared to the regular methods as shown in Table 2. It is noteworthy that our UAPM requires only 31% of Params and 8% of FLOPs compared to the light-weight OCR module, which greatly enhances the efficiency for semantic segmentation.

4.4. Ablation study

Influence of each part. To investigate the contributions of each part in our MAPMaN, we conduct the ablation study on LoveDA and Potsdam datasets, and the results are shown in Table 3. Specifically, the configuration of each variant model is described as follows:

- Model 1: Removing the UAPMs as the baseline and keeping the remaining components the same as our MAPMaN;
- Model 2: Based on Model 1, adding a UAPM after each stage of the backbone; compared to Model 1, its performance verifies the effectiveness of the nested multi-scale modeling;
- Model 3: Based on Model 1, adding an APM module after each stage of the backbone; compared to Model 1, its performance verifies the APM's effectiveness;
- Model 4: Based on Model 2, embedding the APM module calculating weights only along the spatial dimension in the deepest layer of each UAPM; its performance suggests that semantic enhancement methods work well on feature maps with richer semantic information;
- MAPMaN: Based on Model 4, adding the calculation of adaptive weights along the channel dimension; the performance suggests its capability to capture context information of patterns.

Additionally, we investigate the focus of the APM module at each stage in our MAPMaN. As shown in Figure 6, the APMs at shallower stages concentrate more on spatial details and patterns formed by individual ground object, while the ones at deeper stages pay attention to broader scenes. Therefore, the stage-specific adaptability to different ground objects suggests the effectiveness of our APM and multi-scale modeling.

Influence of hyperparameters. Our MAPMaN has two hyperparameters: the downsampling factor and the number of pixels per pattern of the UAPM module at each stage. Considering the vast search space, we adopt the following strategy to approximate the optimal values for the hyperparameters: the same downsampling factor is adopted for the UAPM at each stage, which is the base for the selection of the number of pixels then. We select 25, 16, 9, 5, and 3 as the basic numbers of pixels. Under the hypothesis that shallow features contain less semantic information and require more pixels for pattern matching, we divide all pixel combinations into three categories (i.e., small, medium and large) based on the numbers of pixels in the first stage. Within each category, we progressively increase the numbers of pixels for deeper stages.

Table 5 presents the results regarding hyperparameters on LoveDA test set. It is observable that a downsampling factor of 4 enable the optimal mIoU, and other values leed to performance degradation. Regarding the numbers of pixels, we find that excessive pixels in deeper stages can result in decreased performance, while shallower stages require more pixels. Therefore, we select the downsampling factors to (4, 4, 4, 4) and the numbers of pixels to (16, 9, 5, 5) as the optimal values for the hyperparameters.

Comparison with long-range weight aggregation methods. As shown in Table 4, we compare several weight aggregation methods with long-range modeling on LoveDA and ISPRS Potsdam datasets. Specifically, we replace the APM modules in our MAPMaN with these methods, while the other parts remained unchanged. Global and deformable MHSA modules are all 8-head. The DCNv3 and CovMod blocks are the basic ones in [WDC*23] and [HLCF22]. Experimental results show that using APM for weight aggregation in our MAPMaN achieves the best performance with the least computational cost. We interpret this finding as follows: First, indiscriminate feature aggregation with either Global MHSA or CovMod introduces the background noises of RSIs; second, the DCNv3 block neglects the compositional relationships of ground objects; third, the attention mechanism compresses the dependence along the channel dimension when calculating the similarity map and lacks overall context understanding, which may lead to inconsistent segmentation within large ground objects and performance degradation of the deformable MHSA.

5. Conclusion

In this paper, we proposed *MAPMaN*, a Multi-Stage U-Shaped Adaptive Pattern Matching Network for the semantic segmentation of RSIs. Experimental results on LoveDA, ISPRS Vaihingen, and ISPRS Potsdam datasets demonstrated the outperformance of the proposed method over selected state-of-the-art methods for semantic segmentation of RSIs in several common metrics. In the future, we aim to further improve the proposed method by adopting discrete positional encoding in APM against the unclear order relationship among sampled pixels, as well as processing sampled pixels into tokens with increased semantic information. Meanwhile, we plan to employ recent methods to build a backbone network or a ViT adapter with UAPM as the core, which may lead to a unified model for various tasks on RSIs.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62202421; in part by Zhejiang Provincial Key Research and Development Program under Grant No. 2021C01031; in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LTGS23F020001; in part by Ningbo Yongjiang Talent Introduction Programme under Grant No. 2021A-157-G; and in part by the Public Welfare Science and Technology Plan of Ningbo City under Grant No. 2022S125.

References

[CGRS19] Child R., Gray S., Radford A., Sutskever I.: Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509* (2019). 2

[CNG*23] Chai B., Nie X., Gao H., Jia J., Qiao Q.: Remote sensing images background noise processing method for ship objects in instance segmentation. *Journal of the Indian Society of Remote Sensing* (2023), 1–13. 1

[CZP*18] Chen L.-C., Zhu Y., Papandreou G., Schroff F., Adam H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 801–818. 2, 7, 8

[DBK*20] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020). 1, 2, 9

[DDS22] Digra M., Dhir R., Sharma N.: Land use land cover classification of remote sensing images based on the deep learning approaches: a statistical analysis and review. *Arabian Journal of Geosciences 15*, 10 (2022), 1003. 1

[DQX*17] Dai J., Qi H., Xiong Y., Li Y., Zhang G., Hu H., Wei Y.: Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 764–773. 1, 2

[DTB21] Ding L., Tang H., Bruzzone L.: Lanet: Local attention embedding to improve the semantic segmentation of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing 59*, 1 (2021), 426–435. 1, 2, 7, 8

[FLT*19] Fu J., Liu J., Tian H., Li Y., Bao Y., Fang Z., Lu H.: Dual attention network for scene segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2019), pp. 3146–3154. 7

[HFD*21] Han Q., Fan Z., Dai Q., Sun L., Cheng M.-M., Liu J., Wang J.: On the connection between local attention and dynamic depth-wise convolution. *arXiv preprint arXiv:2106.04263* (2021). 2, 3, 4

[HG16] Hendrycks D., Gimpel K.: Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016). 8

[HJY*22] Hou Q., Jiang Z., Yuan L., Cheng M.-M., Yan S., Feng J.: Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 1 (2022), 1328–1334. 3

[HLCF22] Hou Q., Lu C.-Z., Cheng M.-M., Feng J.: Conv2former: A simple transformer-style convnet for visual recognition. *arXiv preprint arXiv:2211.11943* (2022). 2, 3, 4, 7, 9, 10

[HWH*19] Huang Z., Wang X., Huang L., Huang C., Wei Y., Liu W.: Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 603–612. 2

[JLCY21] Jin Z., Liu B., Chu Q., Yu N.: Isnet: Integrate image-level and semantic-level context for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 7189–7198. 7

[KGHD19] Kirillov A., Girshick R., He K., Dollár P.: Panoptic feature pyramid networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2019), pp. 6399–6408. 2, 7

[LLC*21] Liu Z., Lin Y., Cao Y., Hu H., Wei Y., Zhang Z., Lin S., Guo B.: Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 10012–10022. 2, 7

[LMSR17] Lin G., Milan A., Shen C., Reid I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 1925–1934. 2

[LMW*22] Liu Z., Mao H., Wu C.-Y., Feichtenhofer C., Darrell T., Xie S.: A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 11976–11986. 2, 3, 7

[LSD15] Long J., Shelhamer E., Darrell T.: Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3431–3440. 2

[LYY*23] Luo Z., Yang W., Yuan Y., Gou R., Li X.: Semantic segmentation of agricultural images: A survey. *Information Processing in Agriculture* (2023). 1

[LZZ*22] Li R., Zheng S., Zhang C., Duan C., Su J., Wang L., Atkinson P. M.: Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing 60* (2022), 1–13. doi:10.1109/TGRS.2021.3093977. 7, 8

[PZY*17] Peng C., Zhang X., Yu G., Luo G., Sun J.: Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 4353–4361. 1

[RFB15] Ronneberger O., Fischer P., Brox T.: U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (2015), Springer, pp. 234–241. 2

[RSJ*21] Rottensteiner F., Sohn G., Jung J., Gerke M., Baillard C., Bnitez S., Breitkopf U.: International society for photogrammetry and remote sensing, 2d semantic labeling contest, 2021. 7

[SGLS21] Strudel R., Garcia R., Laptev I., Schmid C.: Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 7262–7272. 2, 7

[SLL*22] Song Q., Li J., Li C., Guo H., Huang R.: Fully attentional network for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2022), vol. 36, pp. 2280–2288. 7

[THK*21] Tolstikhin I. O., Houlsby N., Kolesnikov A., Beyer L., Zhai X., Unterthiner T., Yung J., Steiner A., Keysers D., Uszkoreit J., et al.: Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems 34* (2021), 24261–24272. 3

[VSP*17] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I.: Attention is all you need. *Advances in neural information processing systems 30* (2017). 2

[WDC*23] Wang W., Dai J., Chen Z., Huang Z., Li Z., Zhu X., Hu X., Lu T., Lu L., Li H., et al.: Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 14408–14419. 2, 7, 9

[WGGH18] Wang X., Girshick R., Gupta A., He K.: Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 7794–7803. 2

[WLZ*22] Wang L., Li R., Zhang C., Fang S., Duan C., Meng X., Atkinson P. M.: Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS Journal of Photogrammetry and Remote Sensing 190* (2022), 196–214. 1, 7

[WPLK18] Woo S., Park J., Lee J.-Y., Kweon I. S.: Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 3–19. 2

[WXL*21] Wang W., Xie E., Li X., Fan D.-P., Song K., Liang D., Lu T., Luo P., Shao L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 568–578. 2

[WZC*20] Wang M., Zhao S., Cao X., Yue T., Hu X.: Remote sensing image mixed noise denoising with noise parameter estimation. In *5th International Symposium of Space Optical Instruments and Applications: Beijing, China, September 5–7, 2018* (2020), Springer, pp. 325–333. 1

[WZM*21] Wang J., Zheng Z., Ma A., Lu X., Zhong Y.: Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation. *arXiv preprint arXiv:2110.08733* (2021). 7

[YCW20] Yuan Y., Chen X., Wang J.: Object-contextual representations for semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (2020), Springer, pp. 173–190. 7

[YLZ*22] Yu W., Luo M., Zhou P., Si C., Zhou Y., Wang X., Feng J., Yan S.: Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10819–10829. 7

[YSL*20] Yuan Q., Shen H., Li T., Li Z., Li S., Jiang Y., Xu H., Tan W., Yang Q., Wang J., et al.: Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment 241* (2020), 111716. 1

[YWP*18] Yu C., Wang J., Peng C., Gao C., Yu G., Sang N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 325–341. 2

[ZHLD19] Zhu X., Hu H., Lin S., Dai J.: Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 9308–9316. 1, 2

[ZLDB20] Zhang J., Lin S., Ding L., Bruzzone L.: Multi-scale context aggregation for semantic segmentation of remote sensing images. *Remote Sensing 12*, 4 (2020), 701. 1

[ZSL*20] Zhu X., Su W., Lu L., Li B., Wang X., Dai J.: Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* (2020). 9

[ZSQ*17] Zhao H., Shi J., Qi X., Wang X., Jia J.: Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 2881–2890. 1, 2, 7

[ZTT*22] Zhang B., Tian Z., Tang Q., Chu X., Wei X., Shen C., et al.: Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems 35* (2022), 4971–4982. 2

[ZWK*23] Zhu L., Wang X., Ke Z., Zhang W., Lau R.: Biformer: Vision transformer with bi-level routing attention, 2023. `arXiv:2303.08810`. 7

[ZZT*20] Zhang D., Zhang H., Tang J., Wang M., Hua X., Sun Q.: Feature pyramid transformer. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16* (2020), Springer, pp. 323–339. 2

[ZZWM20] Zheng Z., Zhong Y., Wang J., Ma A.: Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2020), pp. 4096–4105. 7