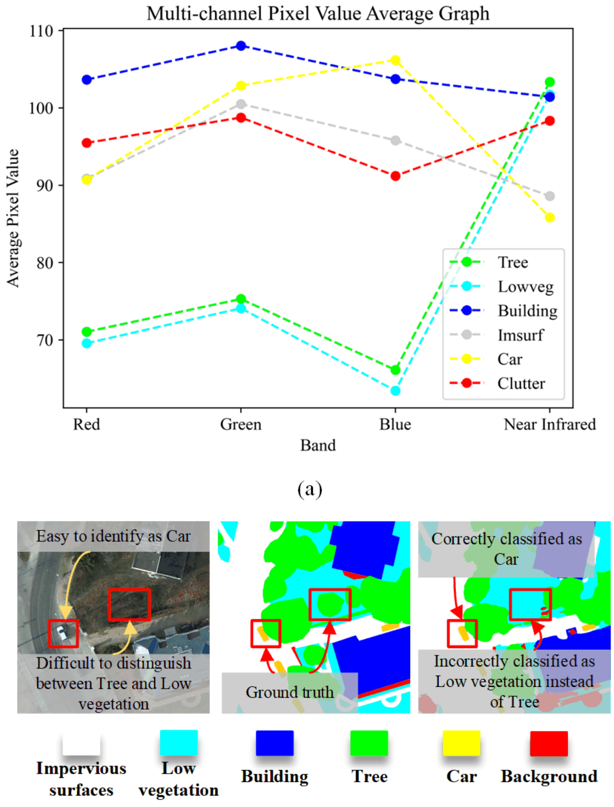
Correlated Mapping Attention Cooperative Network for Urban Remote Sensing Image Segmentation

Yunsong Yang, Genji Yuan, and Jinjiang Li

***Abstract*—In current remote sensing segmentation tasks, the difficulty of segmenting spectrally similar objects is a significant issue. Solving this problem is crucial for improving segmentation accuracy. Traditional image-domain segmentation methods rely on color and texture features, but spectrally similar objects have negligible color differences, leading to suboptimal segmentation results. To address this, we propose a network framework called Correlated Mapping Attention Cooperative Network (CMACNet) by extending the problem from the image domain to the feature domain. Image-domain methods depend on color and texture features, whereas feature-domain methods process higher-level abstract features, avoiding issues caused by color similarity. Specifically, CMACNet first employs an autoencoder structure. The autoencoder compresses the input data and attempts to reconstruct the original data, ensuring that the latent space representations capture essential and representative features of the input data, thereby extracting highly generalized and versatile features. Next, we introduce the correlated mapping attention mechanism, which adaptively adjusts the attention to different features based on their correlations, effectively addressing the challenge of segmenting spectrally similar objects. Furthermore, to efficiently establish global relationships among features, we design a cross global interaction layer for global feature remapping. Comprehensive experiments on the Vaihingen and Potsdam datasets demonstrate that CMACNet outperforms existing state-of-the-art methods, achieving mean intersection over union scores of 84.77% and 87.69%, respectively.**

***Index Terms*—Attention mechanism, global modeling, remote sensing (RS), semantic segmentation, transformer.**



(a)

(b)

Fig. 1. Issue of segmenting land cover with similar spectral characteristics in synthetic images due to color similarity. (a) shows the average pixel values in the red, green, blue, and near-infrared channels for all synthetic images in the Potsdam dataset. It can be seen that the curves for tree and lowveg are very close, indicating strong color similarity. (b) shows examples where categories with similar colors are difficult to distinguish while those with different colors are easy to distinguish. The first column is the original image, the second column is the GT, and the third column is the segmentation result by MPCNet (2023). MPCNet incorrectly segments categories with similar colors but correctly classifies categories such as car, which have significantly different colors.

I. Introduction

With the continuous development of sensors and aerospace technology, people can easily access high-resolution satellite and aerospace remote sensing (RS) images. These images provide high-resolution observations of diverse landscapes on the Earth's surface, covering various scenes from cities to farmlands, and from forests to lakes. RS image segmentation is a crucial technique aimed at partitioning RS images on Earth into different landforms or landform categories. This is essential for geographical information systems (GIS), resource management, environmental monitoring, and crisis management. Here are some key applications of RS image segmentation, such as land cover mapping [1], [2], [3], change detection [4], [5], environmental protection [6], [7], road and building extraction [8], [9], and many other practical applications [10], [11].

In recent years, significant progress has been made in the field of RS image processing with deep learning. Compared to

Manuscript received 21 April 2024; revised 31 May 2024; accepted 10 June 2024. Date of publication 20 June 2024; date of current version 24 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62002200, Grant 61972235, Grant 62272281, and Grant 62202268, in part by the Shandong Natural Science Foundation of China under Grant ZR2023MF026 and Grant ZR2021MF107, in part by the Yantai Science and Technology Innovation Development Plan under Grant 2022JCYJ031 and Grant 2023JCYJ040, and in part by the Special Funds for Taishan Scholars Project under Grant tsqn202306274. *(Corresponding author: Genji Yuan.)*

The authors are with the School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China (e-mail: yuangenji@outlook.com).

The code is available online at https://github.com/yysdck/CMACNet.

Digital Object Identifier 10.1109/JSTARS.2024.3417211

traditional machine learning algorithms, such as in [12], [13], and [14], deep learning automatically extracts useful features from raw data, reducing the burden of feature engineering and aiding in understanding the complex structure and semantics of the data. Various deep-learning-based methods have been proposed in the RS domain [15], covering multiple data types [16], [17]. These methods primarily rely on convolutional neural networks (CNNs) and are capable of effectively classifying and semantically segmenting images [18].

In the field of semantic segmentation, fully convolutional networks (FCNs) [19] were initially proposed, but their design was relatively crude. Subsequently, more refined encoder–decoder structures [20] were introduced, such as U-Net [21], which combines the encoder and decoder through skip connections, thereby improving segmentation accuracy. In the domain of remote sensing images, models such as U-Net has performed exceptionally well, prompting researchers to continually refine them to adapt to different segmentation tasks, such as Unet++[22] and AFF-UNet [23].

Although CNN networks are encouraging in overall segmentation accuracy, they have certain limitations in the field of RS segmentation due to the characteristics of landforms (large scale, high similarity, and mutual occlusion) caused by the local patterns designed by convolution [24]. Some scholars hope to continue using convolutional attention mechanisms for global modeling. For example, MSCSA-Net [25] designs local channel spatial attention and multiscale attention to effectively conduct global context modeling. These methods are still designed to aggregate global information from locally obtained CNN features rather than directly encoding a global context. To address this issue, researchers have proposed Transformer-based networks such as Vit [26] and Swin transformer [27] for global modeling. Currently, mainstream networks tend to directly incorporate Swin-transformer networks into CNN for global modeling, such as ST-Unet [28] and CSTUnet [29]. Although these methods effectively enable global modeling of features within CNN networks, directly introducing Swin-transformer networks into RS segmentation would significantly increase the number of parameters and computational complexity. This would severely impact the practical application of the network in RS segmentation. Therefore, to achieve more efficient global modeling in the field of RS segmentation, we propose a simple cross-pooling method to replace the complex shift-window operation in the Swin transformer. This approach allows for simple and effective global context modeling of features in the RS segmentation domain.

The aforementioned networks are effective to a certain extent in image segmentation. However, in RS image segmentation, there still exists a prominent issue caused by the color similarity of two types of spectrally similar land cover classes in the synthetic images, making them difficult to distinguish (see Fig. 1). These factors also affect the final segmentation accuracy. In the segmentation network, different feature mappings focus on different landforms. Therefore, combining channels according to their importance can determine different landforms. Channels contain texture, edges, and other information, making similar landforms (such as low vegetation and trees, both defined as vegetation in spectroscopy but belonging to different semantic categories in segmentation tasks) show high color similarity in the image [30], which leads to higher channel correlation between the same channel combinations and blurs the difference between different landforms. Although SE-Net [31] and ECA-Net [32] can effectively assign weights to various feature channels, they cannot effectively combine correlation information between channels due to their design constraints.

Therefore, to address the challenge of spectral similarity causing difficulties in segmenting similar land features in RS images, we propose a novel method called the Correlated Mapping Attention Cooperative Network (CMACNet). The key component, correlated mapping attention (CMA), extends the problem from the image domain to the feature domain. It measures the correlation between feature mappings by generating a feature mapping correlation matrix and utilizes this correlation to weight the feature mappings, enhancing the independence of unrelated feature mappings. We conducted experiments to validate the feasibility and effectiveness of this method, and the results demonstrate its effectiveness in addressing the segmentation difficulties caused by spectral similarity among similar land features. In addition, considering the dependence of segmentation networks on downsampled features, we adopt a novel autoencoder approach to enhance the sampling of original features, thereby improving feature generalization. Compared to mainstream RS segmentation methods, we enhance the feature generalization capability of the network using the autoencoder structure and design CMA to address the prominent issue of spectral similarity among similar land features. Furthermore, we introduce a cross global interaction layer (CGIL), which utilizes simple cross-pooling operations to achieve simple and efficient global modeling in RS segmentation, replacing the complex shift-window operations in traditional Swin transformer models.

In summary, the contributions of this article are as follows.

1) *Proposed Correlated Mapping Attention (CMA):* We integrate the correlation information between feature channels by generating a matrix of interchannel correlations, thus embedding the interchannel relationships into the features. This enables the network to better understand the relationships between different feature mappings. Through iterative learning, we gradually reduce the correlation between unrelated channels, thereby enhancing the independence of each channel. This process effectively improves the discriminability of various land cover types, particularly for objects with similar spectral characteristics.
2) *Proposed CGIL:* To integrate the correlated information from feature mappings and incorporate other essential details, we propose a global feature remapping module called CGIL. CGIL employs cross-pooling to interact information across different windows, facilitating efficient global context modeling in RS segmentation. Experimental results demonstrate the effectiveness of this method for global context modeling.
3) *Proposed CMACNet:* Based on CMA and CGIL, we propose CMACNet. In addition, we innovatively introduce an autoencoder structure into the segmentation network. The autoencoder structure emphasizes preserving details and

structure of the original data, offering more possibilities for model generalization and universal feature learning. This enables our network to perform exceptionally well across multiple tasks and datasets.

II. Related Work

RS segmentation is a crucial issue in the field of RS image processing, aiming to divide RS images into different regions with similar objects or object categories. Over the past few decades, with the continuous development of RS technology and the large-scale acquisition of RS data, RS segmentation has attracted widespread attention as an important research task. Related work covers various methods and technologies aimed at improving the accuracy and efficiency of RS image segmentation. When addressing the challenges in this field, researchers need to fully consider the characteristics of the data to ensure the accuracy and robustness of the segmentation results. The success of RS semantic segmentation is of great significance for urban planning, agricultural monitoring, natural disaster management, and other fields. This section will introduce some important works related to RS segmentation and their contributions to this field.

A. CNN-Based RS Image Semantic Segmentation

In RS image semantic segmentation, traditional methods focus on designing robust features compatible with spectral information and local image textures [33], [34], such as the method proposed by Huang et al. [35], which considers environmental and spectral information. With the advancement of RS technology, current research tends to use high-resolution datasets, which, although providing clear geometric information and fine textures [36], also introduce more noise, increasing the difficulty of segmentation.

In recent years, deep learning has been widely applied in the field of RS segmentation. FCN [19] was the first CNN to successfully address the semantic segmentation problem, and since 2015, methods based on CNNs have been dominant in RS semantic segmentation research [37], [38]. To improve segmentation accuracy, researchers introduced UNet [21], an encoder–decoder network focusing on finer semantic segmentation tasks. UNet extracts features through an encoder and gradually restores resolution through a decoder, and the two are interrelated through skip connections, strengthening the network's performance. This encoder–decoder network structure has become the standard model for RS image segmentation, laying the foundation for subsequent research [39]. Subsequent research based on this structure has been conducted extensively, for example, ResUnet [40] improves segmentation accuracy by designing more refined encoder–decoder networks, combining residual stitching, subattribute convolution, pyramid-style scene understanding, and multitask inference to establish conditional relationships between tasks. Wang et al. [41] followed the idea of designing deeper networks and developed a multitask deformable MDE-UNet based on a variable UNet as the basic network. Qiu et al. [42] focused on UNet skip connections and designed a Refine-UNet with a refined skip connection scheme.

CNN segmentation networks guided by object boundary prediction are also an important aspect not to be overlooked. For example, Jung et al. [43] proposed a method combining a holistic nested edge detector and boundary enhancement module. Zheng et al. [44] proposed a Markov-random-field-based multigranularity edge-preserving optimization algorithm for RS segmentation. Sui et al. [45] proposed a segmentation network structure with blockwise edge detection. These novel networks have been carefully designed for RS segmentation tasks, fully exploiting the important role of boundaries in RS segmentation networks.

The previous CNN-based methods have brought impetus to the progress of RS image segmentation. In this article, based on the CNN network, through targeted analysis and design of RS image similarity objects, we propose a CMACNet with a local multiscale enhanced autoencoder, enhancing the generality of features to improve the accuracy of segmentation networks.

B. Global Context Modeling and Attention Mechanism

To overcome the limitations of local patterns in CNNs, researchers have made various attempts, among which one of the most popular methods is to introduce attention mechanisms. This mechanism enables the network to focus on specific regions of the image, thus capturing global context information more effectively. For example, DANet utilizes a dual attention mechanism [46] for global context modeling. Chen et al. [47] introduced local aggregation with graph convolution and attention blocks to more comprehensively capture context information.

Attention mechanisms allow models to focus on specific aspects. For instance, channel attention [31] allows the model to pay more attention to channels with higher weights while spatial attention [48] mechanisms enable models to concentrate on specific spatial locations when processing images or sequence data. This mechanism helps the model capture local features and key information, thereby enhancing the model's perception of local details. These attention mechanisms also play a significant role in RS image segmentation. In addition, some attention mechanisms are used for capturing a global context. For example, Li et al. [40] proposed a linear attention mechanism that reduces computational complexity while maintaining performance for global modeling. Ding et al. [49] introduced local attention blocks to capture richer context information. MSCSA-Net [25] designed local channel spatial attention and multiscale attention to effectively perform global context modeling. These attention mechanisms contribute to performance improvements in segmentation networks, but due to their reliance on convolutional operations, these methods still focus on aggregating global information from locally extracted features by CNNs rather than directly encoding a global context.

Different from CNNs, Transformers transform the task of 2-D image processing into a 1-D sequence task, leveraging powerful sequence-to-sequence modeling capabilities to effectively perform global modeling. Researchers have made significant progress in applying transformers for global context modeling in RS segmentation. The Swin transformer, as an innovative architecture, provides new ideas and paradigms for RS image

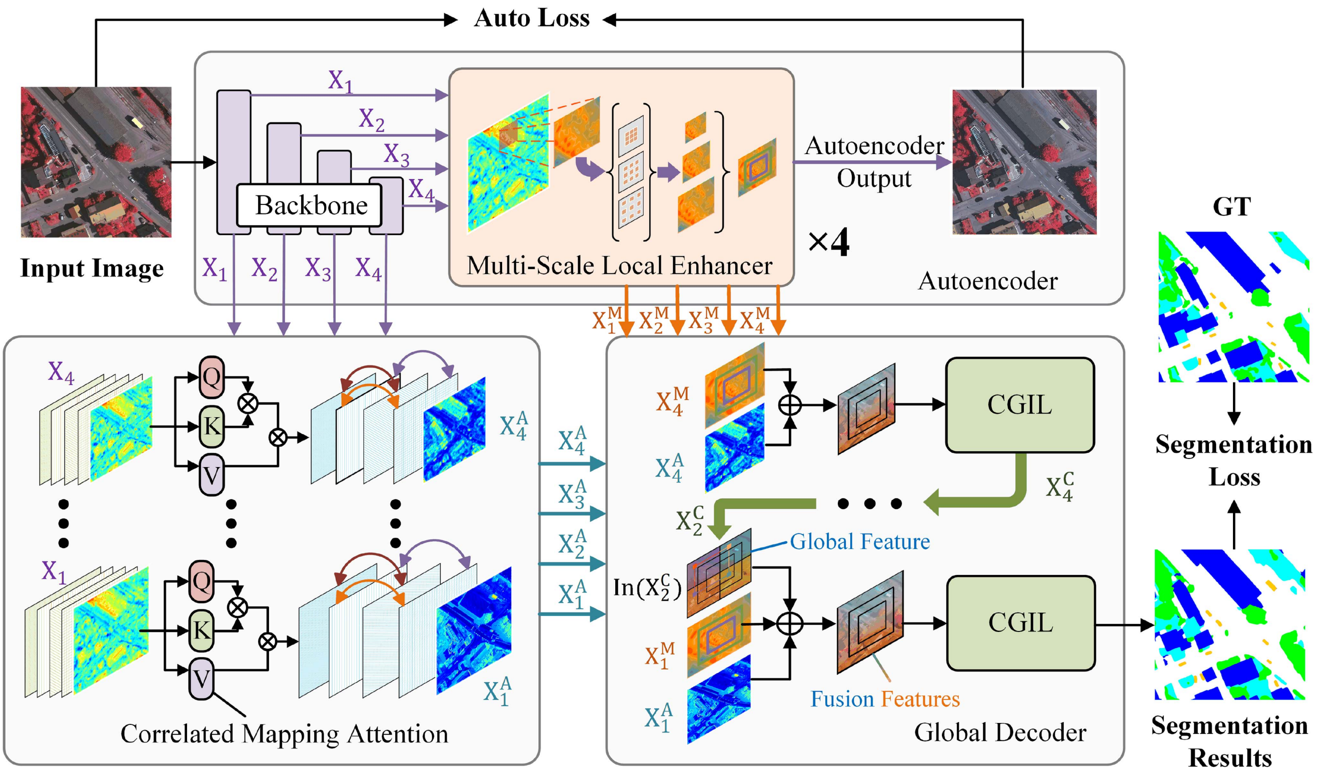


Fig. 2. CMACNet structure diagram. It can be generally divided into two parts. The autoencoder at the top serves as one part, mainly aimed at enhancing the generality of features while the bottom consists of the CMA and global decoder, which form the segmentation part.

segmentation. The application of the Swin transformer in remote sensing image segmentation typically involves two approaches: 1) using models based on a pure transformer structure, such as Segmenter [50] and SwinUNet [51], and 2) integrating transformer and CNN models, such as GLOTS [52] and EMRT [53], which have achieved significant success in this field. This article adopts a method that combines transformerlike and CNN structures. However, due to the computational complexity of the SW-MSA operation in the Swin transformer, networks based on the Swin transformer pose a certain challenge in terms of computation. Therefore, exploring a simpler and more effective global interaction method is crucial.

In this article, inspired by self-attention, we design a modal attention mechanis mechanism capable of perceiving the interrelationship between feature maps to effectively address the difficulty in segmenting spectrally similar land covers. We also propose a novel global transformer block that replaces the complex SW-MSA operation with cross-pooling, which, combined with multihead self-attention, achieves efficient global context modeling and ultimately improves segmentation accuracy.

III. Method

In this section, we will first introduce the overall structure of CMACNet. Following that, we will discuss two important components of CMACNet, namely CMA and CGIL. Finally, we will present the loss function we utilized.

A. CMACNet Structure

The structure of CMACNet is illustrated in Fig. 2. The network can be divided into two main parts: 1) the autoencoder part and 2) the segmentation network part. For the autoencoder part, we adopt an encoder–decoder structure, where the encoder serves as the backbone for feature extraction. In this article, we use a CNN-based downsampling method, with Convnext [54] serving as the backbone due to its proven effectiveness in feature extraction. The decoder part reconstructs the features obtained from the encoder through four upsampling stages to generate the autoencoder's output image. During this process, each downsampled feature is combined with the corresponding downsampled feature size through skip connections while each upsampling stage is followed by a multiscale local enhancement to map local features to multiscale local features. The structure of the multiscale local enhancement decoder is shown in Fig. 3. We extract multiscale information using pooling pyramids of different sizes ($5 \times 5, 9 \times 9, 13 \times 13$) and integrate them with the original features through residual connections, followed by fully connected layers to help the model learn richer and more abstract feature representations. The initial purpose of the autoencoder structure is to learn compact and characteristic representations of the input data and then reconstruct these representations into the original input through the decoder. Since the autoencoder learns effective feature representations by minimizing the error between the input and reconstructed output, it enables the model to capture key information from the input data, enhancing its ability to accurately reconstruct the input data beyond feature extraction and reconstruction solely for segmentation purposes.

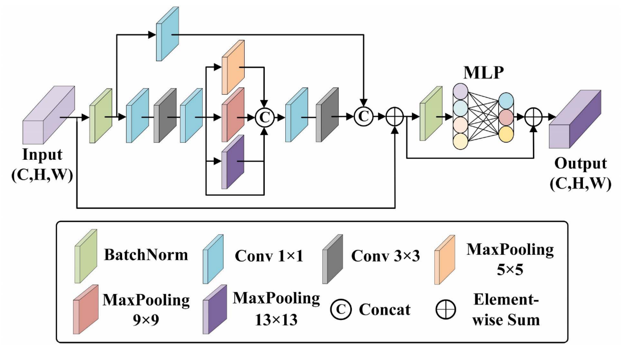


Fig. 3. Structural diagram of the multiscale local enhancer. Obtaining multiscale features through pyramid pooling.

Thus, autoencoders effectively improve feature generalization and model generalization.

For the segmentation network part, the four features from the backbone undergo feature mapping correlation weighting using the CMA mechanism, which dynamically adjusts the relevance between feature mappings. The output features of this process enhance the model's attention to the correlations between different feature mappings, enhancing its perceptual and discriminative abilities. The subsequent operation is global decoding, where the small-scale features are first upsampled, then fused with features from multiscale local enhancement and passed through the CGIL layer for global feature mapping. This process ensures that the model can perceive global features while preserving local multiscale information. Finally, the segmentation head produces the final segmentation image.

Specifically, for an input image $x \in R^{3\times H\times W}$, after initial convolutional operations, it becomes $x' \in R^{C\times H\times W}$, where C represents the number of output channels after convolution, and in this case, $C = 96$. Subsequently, through the four stages of the backbone, feature maps $X_i \in R^{2^i C\times(\frac{H}{2^i})\times(\frac{W}{2^i})}$ are obtained, where $i \in \{1, 2, 3, 4\}$.

For the autoencoder part, starting from X_4, the input feature map X_4 is first mapped to X_4^M through a multiscale local enhancer. Then, X_4^M undergoes convolution and interpolation to adjust its size to match the output X_3 of the backbone's third stage. After being added to X_3 and enhanced by a multiscale local enhancer, X_3^M is obtained, followed by similar operations to obtain X_2^M and X_1^M.

Prior to describing the entire process, some necessary definitions are provided

$$F_{\text{Spp}}(x) = \text{Cat}(\text{MP}_{5\times5}(x), \text{MP}_{9\times9}(x), MP_{13\times13}(x), x) \quad (1)$$

$$F_{\text{Conv1}}(x) = \text{Conv}_{1\times1}(\text{Conv}_{3\times3}(\text{Conv}_{1\times1}(x))) \quad (2)$$

$$F_{\text{Conv2}}(x) = \text{Conv}_{1\times1}(\text{Conv}_{3\times3}(x)) \quad (3)$$

where $F_{\text{Spp}}(\cdot)$ represents spatial pyramid pooling, $F_{\text{Conv1}}(\cdot)$ and $F_{\text{Conv2}}(\cdot)$ represent two consecutive convolution operations, $\text{MP}_{x\times x}(\cdot)$ denotes max-pooling of size $x \times x$, $\text{Cat}(\cdot)$ denotes concatenation, $\text{Conv}_{x\times x}(\cdot)$ represents 2-D convolution with a kernel size of $x \times x$.

Based on the previous formulas, $F_{\text{MSLE}}(\cdot))$ is defined as follows:

$$\begin{aligned} F_m(x) &= \text{Cat}(F_{\text{Conv2}}(F_{\text{Spp}}(F_{\text{Conv1}}(BN(x)))), \\ &\text{Conv}_{1\times1}(BN(x)) + x) \quad (4) \\ F_{\text{MSLE}}(x) &= \text{MLP}(\text{BN}(F_m(x))) + F_m(x) \quad (5) \end{aligned}$$

where $\text{BN}(\cdot)$ denotes BatchNorm operation, $F_m(\cdot)$ is an intermediate function, $F_{\text{MSLE}}(\cdot)$ represents multiscale local enhancement operation, and $\text{MLP}(\cdot)$ denotes fully connected layers. Then, the recursive relationship of X_i^M can be described as follows:

$$X_4^M = F_{\text{MSLE}}(x_4) \quad (6)$$

$$X_i^M = F_{\text{MSLE}}(\text{In}(\text{Conv}_{1\times1}(X_{i+1}^M)) + X_i), i \in \{1, 2, 3\} \quad (7)$$

where $F_{\text{MSLE}}(\cdot)$ represents multiscale local enhancement operation, $\text{In}(\cdot)$ denotes interpolation. X_i represents the output feature map from the ith stage of the backbone, $i \in \{1, 2, 3, 4\}$, X_i^M represents the multiscale locally enhanced features of the ith stage, and the size of X_i^M is represented as $X_i^M \in R^{2^i C\times(\frac{H}{2^i})\times(\frac{W}{2^i})}$, $i \in \{1, 2, 3, 4\}$. X_4^M, X_3^M, X_2^M, and X_1^M are derived using (6) and (7). During model training, X_1^M undergoes simple processing as the output image of the autoencoder used for auto loss with the original image X.

For the segmentation network, the features X_i downsampled by the backbone are first weighted by feature mapping correlation, described as

$$X_i^A = F_{\text{CMA}}(X_i), \quad i \in \{1, 2, 3, 4\} \quad (8)$$

where $F_{\text{CMA}}(\cdot)$ represents the features after undergoing the CMA operation, and X_i^A represents the features after CMA operation for the ith original feature, with the size $X_i^A \in R^{2^i C\times(\frac{H}{2^i})\times(\frac{W}{2^i})}$, $i \in \{1, 2, 3, 4\}$.

In the global decoder part of the segmentation network, a weighted sum operation is performed on a set of features obtained through multiscale weighting and a set of features weighted by feature mapping correlation. Subsequently, several stages of global feature remapping are conducted. Specifically, in the initial stage, X_4^A and X_4^M are weighted and summed to obtain X_4^C through CGIL. Then, X_4^C is interpolated to match the size of X_3^A, and then weighted and summed with X_3^A and X_3^M to obtain X_3^C through CGIL. Similar operations are then performed to obtain X_2^C and X_1^C. For computational convenience, the entire process of global decoding maintains the channel number as $C_1 = 96$ through 1×1 convolutions. This recursive process can be formally described as follows:

$$X_4^C = F_{\text{CGIL}}(a \times \text{Conv}_{1\times1}(X_4^A) + (1-a) \times \text{Conv}_{1\times1}(X_4^M)) \quad (9)$$

$$\begin{aligned} X_i^C = F_{\text{CGIL}}(&b \times \text{ln}(\text{Conv}_{1\times1}(X_{i+1}^C)) + c \times \text{Conv}_{1\times1}(X_i^A) \\ &+ (1-b-c) \times \text{Conv}_{1\times1}(X_i^M)), \quad i \in \{1, 2, 3\} \quad (10) \end{aligned}$$

where $a, b,$ and c represent the weight coefficients, $a, b, c > 0$, $a \leq 1, b + c \leq 1$, $F_{\text{CGIL}}(\cdot)$ represents global feature remapping, and X_i^C represents the features after CGIL operation for the

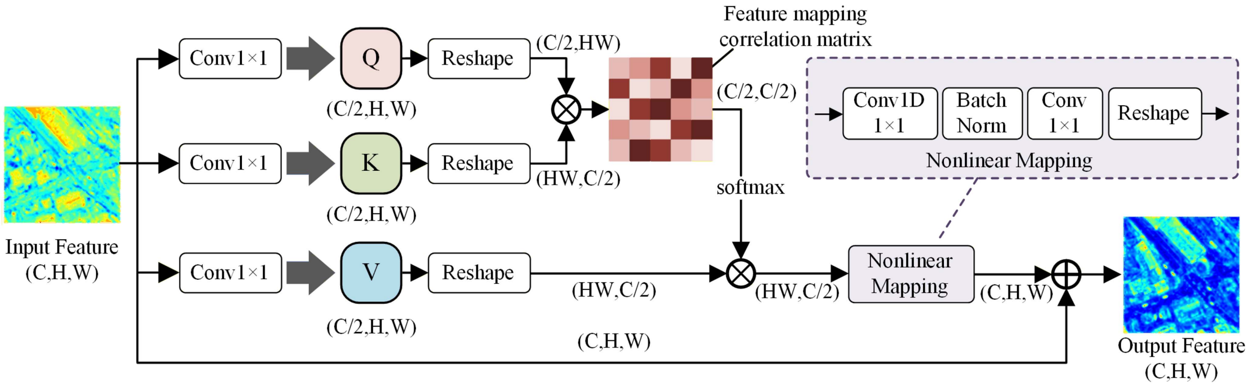


Fig. 4. Structure diagram of CMA.The workflow of CMA involves generating a feature mapping correlation matrix through the QK matrix, then combining it with the V matrix to generate features with feature mapping correlation information. Subsequently, these features are dimensionally increased to match the size of the original features. Finally, they are combined with the original features through residual connections to obtain the output.

ith stage, with the size $X_i^C \in R^{C\times(\frac{H}{2^i})\times(\frac{W}{2^i})}, i \in \{1,2,3,4\}$. X_4^C, X_3^C, X_2^C, and X_1^C are derived using (9) and (10). Finally, the segmentation result is obtained from X_1^C through the final segmentation head.

In summary, CMACNet achieves higher performance levels in both feature learning and segmentation task execution by combining the advantages of feature generalization enhancement and model generalization through the autoencoder, as well as the advantages of dynamically adjusting feature mapping correlation and efficiently utilizing global information through the segmentation network. The structural design of CMACNet enables the model to effectively utilize information from input data, achieving superior performance in RS image segmentation tasks. In the next section, we will introduce the details of two important components in CMACNet, namely 1) CMA and 2) CGIL.

B. Correlated Mapping Attention

In RS images, objects with highly similar spectral characteristics typically exhibit similar colors or brightness levels. This means that their color intensity and brightness in composite images are very close, making them difficult to distinguish through conventional visual analysis. This is one of the main reasons why image-domain segmentation methods struggle to segment spectrally similar objects in RS images.

Self-attention mechanisms address this by computing attention values between pairs of pixels to form a self-attention matrix, enabling the model to focus on different parts of the spatial domain. This allows the model to automatically learn and determine which areas are important for specific tasks based on the content of the input image. Inspired by this concept, we designed the CMA mechanism. Unlike self-attention, CMA calculates correlation values between pairs of feature maps in the channel direction and uses these values to construct a feature map correlation weight matrix. In addition, we apply nonlinear mapping to project the feature space into a higher-dimensional space, making the differences between objects more pronounced and easier to distinguish. CMA dynamically adjusts the feature map correlation weight matrix, increasing the independence of unrelated channels and decreasing the correlation of related channels to address the aforementioned problem. CMA demonstrates significant advantages in segmenting objects with high spectral similarity and similar color characteristics. The computational process is illustrated in Fig. 4.

Specifically, for an initial input $F_{\text{in}} \in R^{C\times H\times W}$, we first perform nonlinear weighted operations using three 1×1 convolutions to obtain three matrices, described as follows:

$$\begin{cases} Q = \text{Conv}_{1\times1}(F_{\text{in}}) \\ K = \text{Conv}_{1\times1}(F_{\text{in}}) \\ V = \text{Conv}_{1\times1}(F_{\text{in}}) \end{cases} . \tag{11}$$

Here, $Q \in R^{(C/2)\times H\times W}$, $K \in R^{(C/2)\times H\times W}$, and $V \in R^{(C/2)\times H\times W}$ represent the three matrices obtained after convolution, used for subsequent attention calculation. In this step, by reducing the number of channels by half, we decrease the dimensionality of the features, filtering out minor noise and redundancy while retaining essential information. This aids in enhancing the model's abstraction capability to better capture key features in the data. We then reshape Q, K, and V matrices into $Q_{\text{re}} \in R^{(C/2)\times(HW)}$, $K_{\text{re}} \in R^{(C/2)\times(HW)}$, and $V_{\text{re}} \in R^{(HW)\times(C/2)}$, respectively. Next, by multiplying Q_{re} and K_{re}^T, we obtain a feature map association matrix with channel-to-channel correlation scores of size $(C/2, C/2)$, where an increase in matrix values signifies a higher correlation between feature maps, and a decrease indicates a lower correlation. Subsequently, by applying Softmax to the product of Q_{re} and K_{re}^T, we normalize the result to obtain the feature map association weight matrix. This weight matrix is then used to weight V_{re}, yielding attention values, expressed by the following equations:

$$\begin{cases} Q_{\text{re}} = \text{re}(Q) \\ K_{\text{re}} = \text{re}(K) \\ V_{\text{re}} = \text{re}(V) \end{cases} \tag{12}$$

$$F_w = \text{Softmax}(Q_{\text{re}} \cdot K_{\text{re}}^T) \tag{13}$$

$$F_{\text{AV}} = V_{\text{re}} \cdot F_w. \tag{14}$$

Here, re$(\cdot)$ denotes the reshape operation, F_w represents the association weight matrix with dimensions $(C/2, C/2)$, and F_{AV} denotes the attention values matrix with dimensions $(H \times W, C/2)$. To ensure consistency in subsequent operations and introduce more nonlinear features to enhance the model's expressive power, we perform nonlinear mapping of features using a combination of 1-D and 2-D convolution kernels. Finally, we perform a residual connection by combining the expanded attention values with the original input. By applying nonlinear mapping, we elevate the feature space to a higher dimension, making the differences between objects more pronounced and distinguishable. In RS images, objects with similar color or brightness might be difficult to differentiate in the original feature space. However, through nonlinear mapping, we can generate more complex feature representations, thereby better capturing the subtle differences between objects.The operation of nonlinear mapping is defined as follows, for input features $x \in R^{(C/2)\times(HW)}$:

$$\text{NM}(x) = \text{re}(\text{Conv}_{1\times 1}(\text{BN}(\text{Conv1D}_{1\times 1}(x)))). \tag{15}$$

Here, $\text{NM}(\cdot)$ represents the operation of nonlinear mapping on x, and $\text{Conv1D}_{1\times 1}(\cdot)$ denotes a 1×1 convolution, which outputs features of size $R^{C\times H\times W}$. Combining the earlier equations, the CMA operation can be expressed as

$$F_{\text{out}} = F_{\text{CMA}}(F_{\text{in}}) = \text{re}(\text{NM}(F_{\text{AV}})) + F_{\text{in}}. \tag{16}$$

Here, $F_{\text{out}} \in R^{C\times H\times W}$ represents the output features. Throughout the entire process, the model progressively learns to adaptively increase the correlation of channels with high similarity while reducing the correlation of channels with low similarity by dynamically adjusting the feature map association matrix. This adjustment aims to enhance the independence of noncorrelated channels while reducing the independence of correlated channels

Objects with high spectral similarity typically exhibit similar band combinations, which manifest as similar RGB values in the image domain of RS imagery. In the feature domain of deep learning, objects with high spectral similarity usually show high similarity in the importance of corresponding channels in feature maps. Therefore, to effectively differentiate between different objects, it is necessary to consider the contribution of each channel in differentiating objects by assessing the correlation between channels. This means that when processing features of similar objects in segmentation networks, the network is more likely to use combinations of highly similar channels, where the correlation between these channel contribution combinations is high. Conversely, for distinguishing between different objects, the network may prefer to avoid strong channel correlations. This is because different object categories usually have different channel contribution combinations, and using combinations with lower correlations enables the network to better differentiate between objects. Therefore, when facing similar objects with similar spectral characteristics or texture and color information, accurate segmentation becomes challenging due to the similarity in their channel contribution combinations. CMA utilizes the association weight matrix to enhance the independence of noncorrelated channels and reduce the independence of correlated channels to address this issue. This operation helps the network to handle similar objects more flexibly and improves the performance of segmentation networks in scenarios with high spectral similarity. Using the correlation of feature maps as a way to allocate channel contributions is necessary for distinguishing between similar objects with similar spectral characteristics.

C. Cross Global Interaction Layer

Existing effective strategies for global feature reconstruction mainly rely on the Swin transformer. However, through an in-depth analysis of existing methods, we found that the complex window shift mechanism may lead to inefficiency issues. In practical applications, this mechanism may introduce a large number of parameters and computational costs, limiting the practical availability of the model in resource-limited environments. Through our research, we have proposed a more efficient method for global feature mapping, called CGIL. The structure of CGIL is illustrated in Fig. 5. Its overall design is inspired by the Swin transformer, the structure of which can be found in [27]. The Swin transformer comprises two sequential standard modules in its base layer. The W-transformer block ensures global information interaction within the window while the SW-transformer block, with a window shift mechanism, is used to achieve interwindow information interaction. Our proposed CGIL still utilizes the W-transformer for intrawindow global information interaction. However, unlike the Swin transformer, we directly employ a cross-pooling method after the multihead self-attention mechanism in the W-transformer to achieve efficient interwindow information interaction. This enables a single CGIL to perform both intrawindow and interwindow information interaction simultaneously.

Specifically, in the global context interaction module, we first divide the input features into multiple windows and then embed relative position encodings for each relative position, similar to the Swin transformer. Next, we establish global context relationships within each window using the multihead self-attention mechanism of the W-transformer block. For features that have already established intrawindow global context relationships, we employ an efficient cross-pooling method to iteratively build information exchanges between the current window and its adjacent windows to the right and below, thus achieving interwindow information interaction.

In detail, for a single pixel $P(x, y)$, we define ws as the window size and use a horizontal average pooling kernel of size $1 \times (ws)$ and a vertical average pooling kernel of size $(ws) \times 1$ to establish relationships between $P(x, y)$ and pixels $P(x + ws, y)$ and $P(x, y + ws)$, respectively. These pixels belong to different windows. Finally, the features obtained after two cross-pooling operations are added together. Through this interaction mechanism, we successfully establish correlations between different windows. The mathematical expressions are as follows:

$$C_{\text{horizontal}}(x, y) = \frac{1}{ws}\sum_{i=1}^{ws} P(x + i, y) \tag{17}$$

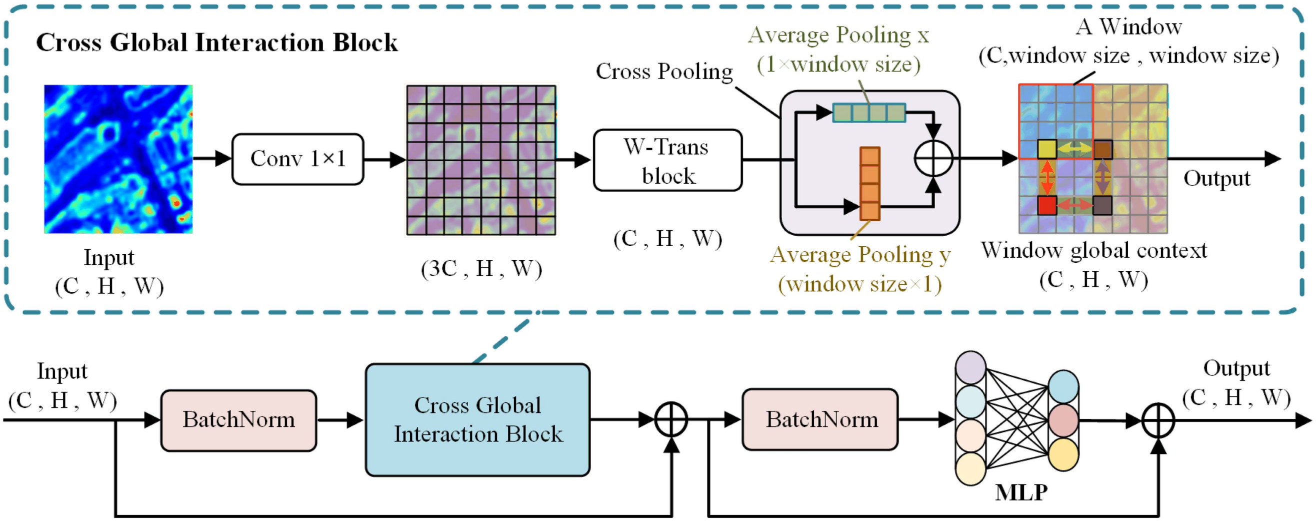


Fig. 5. Structure diagram of CGIL. Replacing SW-MSA with cross-pooling.

$$C_{\text{vertical}}(x, y) = \frac{1}{ws} \sum_{j=1}^{ws} P(x, y + j) \tag{18}$$

where $C_{\text{horizontal}}(x, y)$ represents the contextual information of $P(x, y)$ in the horizontal direction, and $C_{\text{ vertical}}(x, y)$ represents the contextual information of $P(x, y)$ in the vertical direction. The contextual information in both directions is then fused, defined as

$$C_{\text{global}}(x, y) = C_{\text{horizontal}}(x, y) + C_{\text{vertical}}(x, y). \tag{19}$$

Here, $C_{\text{global}}(x, y)$ represents establishing global context for a single feature pixel $P(x, y)$ through simple horizontal and vertical pooling operations, successfully establishing global context relationships between windows.

To establish the global context for each pixel of the entire feature map, we first concatenate the previously divided windows into the input size, reshape it to match the original image size, and then establish global context for each feature pixel using a single-pixel global context approach. Specifically, for a feature map x concatenated according to windows, $x \in R^{C\times H\times W}$, we have

$$F_{\text{GC}} = C_{\text{global}}(x) \tag{20}$$

where F_{GC} represents the features with established global context between windows, with size $F_{\text{GC}} \in R^{C\times H\times W}$. $C_{\text{global}}(\cdot)$ denotes the operation of establishing global context for the entire feature. Then, the CGIB process can be defined as follows: for input features $x \in R^{C\times H\times W}$:

$$F_{\text{CGIB}}(x) = F_{\text{wt}}(F_{\text{GC}}) \tag{21}$$

where $F_{\text{wt}}(.)$ represents the operation of the W-transformer block in the Swin transformer, the derivation of which can be referred to in [27]. $F_{\text{CGIB}}(x)$ represents the operation through CGIB. Finally, for input features $F_{\text{in}} \in R^{C\times H\times W}$, the features through CGIL can be defined as

$$\begin{aligned} F_{\text{out}} &= F_{\text{CGIL}}(F_{\text{in}}) \\ &= \text{MLP}(\text{BN}(F_{\text{CGIB}}(\text{BN}(F_{\text{in}}))) + F_{\text{in}}) \\ &\quad + (F_{\text{CGIB}}(\text{BN}(F_{\text{in}})) + F_{\text{in}}). \end{aligned} \tag{22}$$

Here, $F_{\text{out}} \in R^{C\times H\times W}$ represents the output features after the CGIL operation.

This kind of global feature integration provided by CGIL helps to improve the discrimination of complex objects and enables the model to better adapt to different regions and scenes, thereby enhancing the performance of the model on new data. Meanwhile, CGIL significantly reduces computational costs through cross-pooling operations. In summary, the global feature reconstruction strategy not only enhances the adaptability and generalization capabilities of the model but also effectively balances the attention to local and global information, improving the accuracy and robustness of RS image segmentation models.

D. Loss Function

In this section, we will introduce the loss function used in our proposed method in this article.

For the two parts of the entire network, we adopt a joint loss L_{joint} for both parts as our overall loss function. For the autoencoder part, since the task of the autoencoder is to learn a compact representation of the input data and minimize the error between the input data and its reconstruction, we use mean-squared error (MSE) loss as the standard to measure the difference between input and reconstruction. The formula for MSE loss is as follows:

$$L_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \widehat{y_i})^2 \tag{23}$$

where N represents all samples, $(y_i - \widehat{y_i})^2$ represents the squared error for each sample, L_{MSE} represents the MSE loss used by the autoencoder. This loss function measures the difference between the predicted values and the true values, and by optimizing the MSE loss, the model aims to make the predicted values as close to the true values as possible.

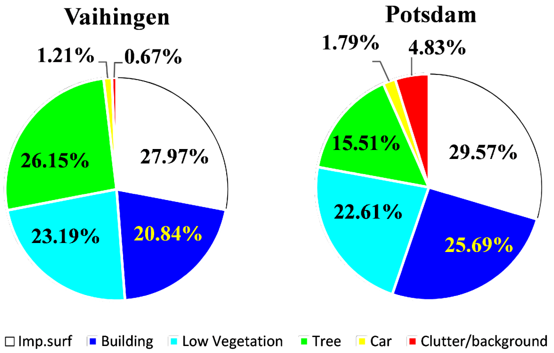


Fig. 6. This figure illustrates the proportion of each semantic label in the two datasets.

For the segmentation loss, the loss L_s used in this article is a combination of Dice loss L_{dice} and cross-entropy loss L_{ce}. This combined loss considers two different contributions in the image segmentation task. The dice loss mainly focuses on matching object boundaries and is suitable for accurate segmentation of similar objects of the same class. It evaluates the model performance by measuring the overlap between the predicted results and the ground truth (GT) labels. Cross-entropy loss is commonly used in image segmentation tasks to encourage the model to produce segmentation results closer to the GT labels. However, in cases of similar objects, Cross-entropy loss may cause the model to struggle in distinguishing object boundaries, affecting the segmentation results. The expressions for L_{ce} and L_{dice} are as follows:

$$L_{\text{ce}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} y_k^{(n)} \log \hat{y}_k^{(n)} \tag{24}$$

$$L_{\text{dice}} = 1 - \frac{2}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} \frac{\hat{y}_k^{(n)} y_k^{(n)}}{\hat{y}_k^{(n)} + y_k^{(n)}} \tag{25}$$

$$L_s = L_{\text{ce}} + L_{\text{dice}}. \tag{26}$$

In which N is the number of samples, and K is the number of categories. $y^{(n)}$ and $\widehat{y}^{(n)}$ represent the one-hot encoding of the true semantic labels and their corresponding Softmax outputs from the network, where $n \in [1, \ldots, N]$). $\hat{y}_k^{(n)}$ denotes the confidence score of class k for sample n.The final joint loss L_{joint} is given by

$$L_{\text{joint}} = k_1 L_{\text{MSE}} + k_2 L_s \tag{27}$$

where $k1$ and $k2$ represent the weighting parameters for the autoencoder loss and segmentation loss, respectively. We set them to 0.3 and 0.7, respectively.

IV. Experiment

In this section, we will first introduce the dataset, experimental setup, and relevant metrics. Then, we will present our ablation experiments, and finally, we will discuss comparative experiments with other methods.

A. Experimental Settings

1) Datasets: In the experimental data support section of this study, we adopted the Vaihingen and Potsdam datasets as evaluation standards (see Fig. 6). These datasets are recognized as benchmark data in the field of semantic segmentation of RS images and are widely used to assess the performance of RS algorithms. This ensures that our research conclusions have broad applicability and comparability. These datasets are characterized by their diversity of land cover classes and complex environmental modalities, ranging from buildings and road systems to forest cover. They also incorporate scenes with seasonal changes, varying weather conditions, and different lighting conditions, providing a highly realistic testbed for examining the robustness and generalization capabilities of our model. Importantly, since these datasets have been extensively studied and widely adopted by the academic community, we can easily compare our work with previous research. In the following, we will provide a detailed introduction to these two datasets.

ISPRS Vaihingen: The dataset originates from the RS images of the Vaihingen area in Germany, comprising 33 high spatial resolution true orthophoto (TOP) image blocks. Each image block has an average size of 2494×2064 pixels. Each block consists of TOP, digital surface model (DSM), and normalized DSM (NDSM), using only three multispectral bands (near-infrared, red, and green). The dataset includes five foreground land cover classes (impervious surfaces, buildings, low vegetation, trees, and cars) and one background land cover class (clutter). In our experiments, we strictly followed the specific training IDs provided by the ISPRS benchmark (IDs 1, 3, 5, 7, 11, 13, 15, 17, 21, 23, 26, 28, 30, 32, 34, 37) to select training data. The remaining 17 images were reserved for testing. This selection ensures consistency with other researchers' data, facilitating better comparative analysis. Image blocks were cropped into 1024×1024 pixel patches for processing.

ISPRS Potsdam: The dataset utilizes the aerial images of Potsdam, Germany, consisting of 38 high spatial resolution TOP image blocks. Each image block has a ground sampling distance of 5 cm and a size of 6000×6000 pixels. Similar to Vaihingen, each block comprises TOP and DSM, providing four multispectral bands (red, green, blue, and near-infrared). During the training process, we also utilized specific training IDs provided by the ISPRS benchmark for model training. These training IDs include images labeled as 2_10, 2_11, 2_12, 3_10, 3_11, 3_12, 4_10, 4_11, 4_12, 5_10, 5_11, 5_12, 6_7, 6_8, 6_9, 6_10, 6_11, 6_12, 7_7, 7_8, 7_9, 7_11, and 7_12. The remaining 15 images were kept as the test set. Similarly, they were cropped into 1024×1024 pixel patches for analysis.

In quantitative evaluation, the "clutter/background" category was disregarded.

2) Implementation Details: In this study, we selected the Ubuntu 18.04 operating system and deployed all models on a single NVIDIA GeForce RTX 2080 Ti 11 GB GPU, implemented using the PyTorch 1.11 framework. To achieve faster model convergence, we employed the AdamW optimizer with a base learning rate set to 6e-4 and utilized a cosine learning rate schedule for adjusting the learning rate. For the Vaihingen and Potsdam datasets, we conducted the following data preprocessing steps. First, we randomly cropped images into patches of size 512×512. During the training phase, we introduced multiple data augmentation techniques, including random scaling ([0.5, 0.75, 1.0, 1.25, 1.5]), random vertical flipping, random horizontal flipping, and random rotation. The training process consisted of 105 epochs. During the testing phase, we employed multiscale evaluation and random flipping augmentation techniques to ensure the robustness and performance of the models.

3) Evaluation Metrics: In this experiment, commonly used evaluation metrics in RS segmentation, including overall accuracy (OA), F1 Score, and mean Intersection over Union (mIoU), are employed as evaluation metrics. In addition, the number of parameters is utilized as a metric for assessing the model's complexity. Before delving into these metrics, it is essential to introduce some related terms, such as precision and recall. Moreover, an understanding of certain symbols is required: tp (true positive), fp (false positive), fn (false negative), and tn (true negative).

Precision: Precision measures the proportion of true positive samples among all samples predicted as positive by the model. In other words, precision indicates the likelihood of a sample being genuinely positive when the model predicts it as positive

$$\mathrm{Precision} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp}}. \tag{28}$$

Recall: Recall refers to the proportion of true positive samples correctly predicted as positive among all true positive samples. Recall evaluates the model's ability to identify all positives

$$\mathrm{Recall} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fn}}. \tag{29}$$

Overall Accuracy (OA): OA is a commonly used performance evaluation metric in image classification tasks. It represents the proportion of correctly classified samples to the total number of samples. However, OA may not handle class imbalances well, as the model might lean toward predicting classes with more samples when some classes have significantly more samples than others

$$\mathrm{OA} = \frac{\mathrm{tp} + \mathrm{tn}}{\mathrm{tp} + \mathrm{fp} + \mathrm{fn} + \mathrm{tn}}. \tag{30}$$

F1 Score: The F1 score is the harmonic mean of precision and recall. It synthesizes the model's accuracy and ability to capture positives. For multiclass problems, F1 score is typically calculated for each class and then averaged

$$F1 = \frac{2 \times (\mathrm{Precision} \times \mathrm{Recall})}{\mathrm{Precision} + \mathrm{Recall}}. \tag{31}$$

Overall F1 score = Average of F1 scores for all classes.

Mean Intersection over Union (mIoU): mIoU is a commonly used evaluation metric in semantic segmentation tasks, measuring the model's accuracy in pixel-level segmentation. Intersection over Union (IoU) evaluates the model's segmentation results for each class while mIoU computes the average IoU for all classes. For each class

$$\mathrm{IoU} = \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp} + \mathrm{fn}} \tag{32}$$

TABLE I
RESULTS OF REMOVING INDIVIDUAL COMPONENTS FROM CMACNET

Method	**Vaihingen**		**Potsdam**	
	mIoU(%)	**F1(%)**	**mIoU(%)**	**F1(%)**
CMACNet	84.77	91.66	87.69	93.33
CMACNet W/O Autoencoder	83.76	91.07	86.70	92.77
CMACNet W/O CMA	83.49	90.88	86.38	92.60
CMACNet W/O CGIL	83.52	90.90	86.46	92.61

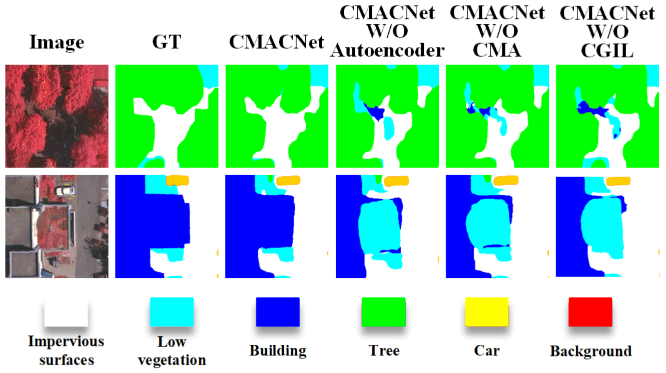


Fig. 7. Comparison of experimental results between CMACNet and structures removal. From the figure, it can be observed that removing any structure from CMACNet leads to deteriorated segmentation results.

mIoU is the sum of IoU values for all categories divided by the number of categories.

B. Ablation Experiment

1) Components of CMACNet: To evaluate the performance of various components of CMACNet, a series of ablation experiments were conducted, and validation was carried out using the Vaihingen and Potsdam datasets. In the discussion, our focus primarily lies on two performance metrics: 1) mIoU and 2) meanF1. All experimental results presented are averages obtained from multiple trials.

Table I presents the results of CMACNet with individual modules removed. Here, "Autoencoder" represents the autoencoder, "CMA" denotes correlated mapping attention, and "CGIL" stands for cross global interaction layer. For CMACNet, when removing the autoencoder operation, since the multiscale local enhancer is part of the autoencoder, we set the removal of autoencoder in the experiment to be the removal of multiscale local enhancer simultaneously, without generating the original image at the end or computing loss with the original image. The components removed in the following ablation experiments are marked with (w/o), and the added components are marked with (+).

Fig. 7 presents the segmentation results after removing individual modules from CMACNet. It can be observed that the segmentation performance of CMACNet decreases when any module is removed. To further validate the effectiveness of each component, experiments were conducted by incrementally

TABLE II
RESULTS OF ADDING INDIVIDUAL MODULES ON BASELINE MODEL ON THE VAIHINGEN DATASET

Method	mIoU(%)	F1(%)
Baseline	82.54	90.28
Baseline+Autoencoder	83.31	90.79
Baseline+CMA	83.64	90.98
Baseline+CGIL	83.54	90.91

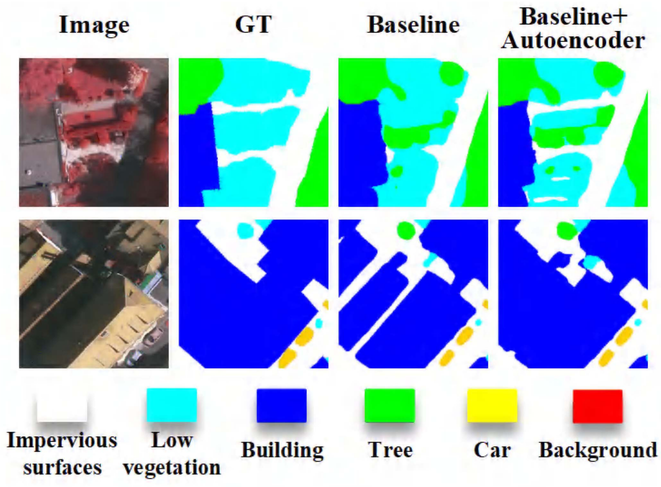


Fig. 8. Experimental results of adding autoencoder on top of the baseline. From the figure, it can be observed that the segmentation results with autoencoder are more complete, demonstrating the benefit of improved feature generalization.

adding individual modules to the baseline model on the Vaihingen dataset. The baseline model was set to use Unet as the main framework and ConvNext as the encoder. Table II shows the experimental results of adding individual modules to the baseline model.

For experiments adding the autoencoder to the baseline model, we incorporated multiscale local enhancement into the baseline and generated original images and segmentation maps after the baseline model. The combined loss of both was used as the final loss. Figs. 8, 10, and 11 illustrate the effects of adding any single component to the baseline model. The comprehensive results indicate that each component proposed in our model has a positive impact on model performance.

2) Effect of Autoencoder: From Table I, it can be observed that removing the autoencoder from CMACNet results in a decrease of mIoU and F1 scores by 1.01% and 0.59%, respectively, on the Vaihingen dataset, and by 0.99% and 0.56%, respectively, on the Potsdam dataset. From Table II, a more intuitive view reveals that using autoencoder on the baseline model brings about improvements of 0.77% and 0.51% in mIoU and F1 scores, respectively, on the Vaihingen dataset. It can be said that employing autoencoder can lead to at least a 0.77% increase in mIoU and a 0.51% increase in F1 for segmentation tasks.

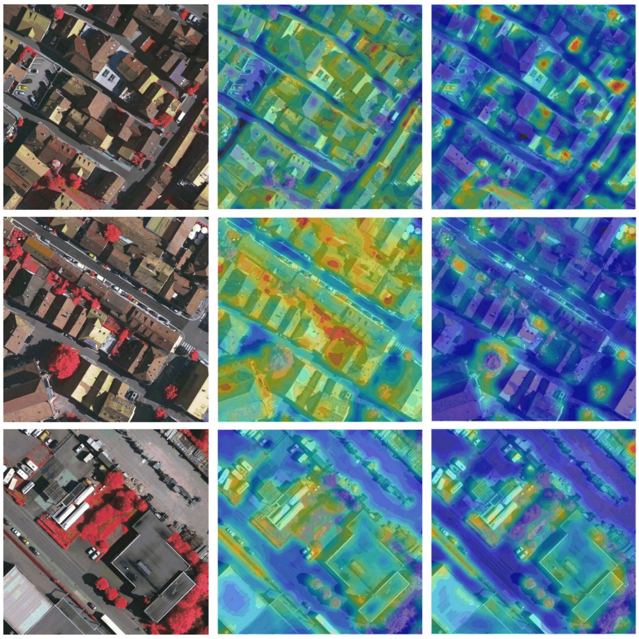


Fig. 9. Attention maps before and after using CMA. The first column represents the original image, the second column represents the features before using CMA, and the third column represents the features after using CMA. The colors indicate the weights, with red indicating higher weights and blue indicating lower weights. It can be observed that the attention is dispersed before using CMA, while after using CMA, the weights are concentrated on objects with similar spectral characteristics such as trees and low vegetation.

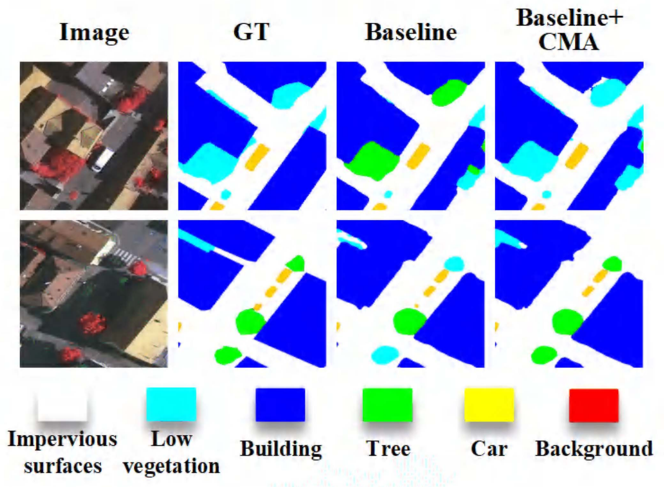


Fig. 10. Segmentation results with the addition of CMA on top of the baseline. It can be observed that the model achieves more accurate segmentation of tree and low vegetation classes after incorporating CMA.

From the visualized results in the fourth column of Fig. 7, it is evident that removing the autoencoder results in model errors due to the lack of multiscale and local characteristics, leading to inaccuracies in identifying objects with subtle features. Fig. 8 confirms this observation as well, showing that due to the absence of multiscale characteristics, the model fails to accurately capture features of objects at different scales, resulting in incomplete or inaccurate segmentation results. The first row in Fig. 8

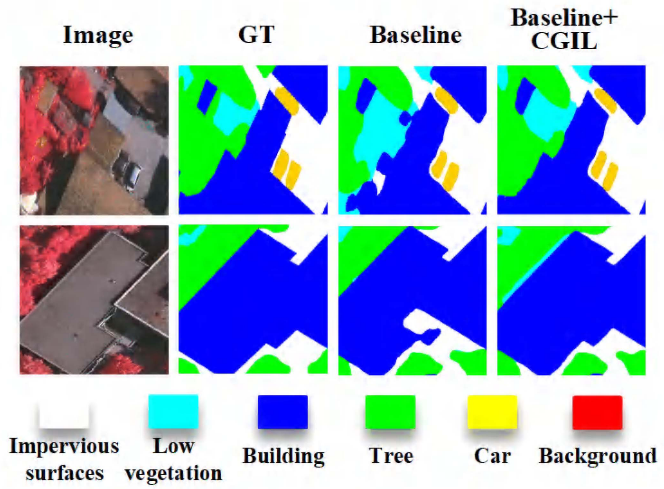


Fig. 11. Segmentation results with CGIL added on top of the baseline. The segmentation accuracy for large buildings is improved by the CGIL model, highlighting the advantage of global features.

demonstrates that the baseline model has some perception of the impervious surface class but it is incomplete while the second row shows that the baseline model perceives buildings but, due to factors such as lighting, misclassifies buildings as impervious surfaces. In contrast, features enhanced by the autoencoder generalize better to multiscale local characteristics, resulting in more complete and accurate segmentation results while also preserving detailed features.

3) Effect of CMA: Table I shows that for CMACNet, removing CMA results in a decrease of 1.28% and 0.78% in mIoU and F1 scores, respectively, on the Vaihingen dataset, and a decrease of 1.31% and 0.73%, respectively, on the Potsdam dataset. Table II indicates that adding CMA to the baseline model increases mIoU and F1 scores by 1.1% and 0.7%, respectively. From the data, it is evident that using CMA in the model can lead to at least a 1.1% increase in mIoU and at least a 0.7% increase in F1.

For the lowveg and tree classes in Vaihingen and Potsdam, their spectral similarities often lead to misclassifications in the segmentation network. The segmentation results of these two classes represent the segmentation effects of the model on classes with similar spectral characteristics. To further demonstrate the classification advantages of CMA for classes with better spectral similarity, additional experiments were conducted. Segmentation performance experiments were performed on the lowveg and tree classes in the baseline model with the addition of CMA on the Vaihingen dataset. The experimental results in Table III show that after applying the CMA method, the IoU and F1 of the lowveg class are improved by 3.24% and 2.16%, respectively, while the IoU and F1 of the tree class are improved by 2.37% and 1.42%, respectively. The table clearly demonstrates the superiority of the CMA method in achieving higher classification accuracy for classes with similar spectral characteristics.

TABLE III
SEGMENTATION RESULTS OF LOW VEGETATION AND TREES ON THE VAIHINGEN DATASET AFTER ADDING CMA TO THE BASELINE

Method	**Lowveg**		**Tree**	
	mIoU(%)	**F1(%)**	**mIoU(%)**	**F1(%)**
Baseline	71.42	83.33	81.26	89.66
Baseline+CMA	74.66	85.49	83.63	91.08

Fig. 9 presents the visualized attention maps of features before and after CMA. It can be seen that the features before CMA do not exhibit obvious focus points, while after CMA, the features tend to focus on objects with similar spectral characteristics. Fig. 10 illustrates the segmentation results after adding the CMA method to the baseline model, showing noticeable improvements in segmentation accuracy for the tree and lowveg classes. This demonstrates the effective improvement of segmentation accuracy for objects with similar spectral characteristics by the CMA method.

4) Effect of CGIL: From Table I, it can be observed that when CGIL is removed, CMACNet experiences a decrease of 1.25% and 0.76% in mIoU and F1 scores, respectively, on the Vaihingen dataset. Similarly, on the Potsdam dataset, there is a reduction of 1.23% and 0.72% in mIoU and F1 scores, respectively. Furthermore, Table II demonstrates that after the addition of CGIL, the mIoU improves by 1% in the baseline model.

To demonstrate the efficiency and effectiveness of CGIL in global modeling, experiments were conducted by replacing CGIL with other global information mapping modules in the global decoding stage across four stages. The modules compared are derived from commonly used global information mapping models, including Vit [26], Swin-T [27], Mobile-Vit [55], and Fast-Vit [56]. Experimental results are detailed in Table IV. Since the channel number remains constant throughout all stages of the feature reconstruction phase in our designed model, the parameter count remains unchanged. Therefore, only one instance of parameter count is recorded in the table. It can be seen from Table IV that CGIL outperforms other networks in terms of both parameter count and FLOPs. The only model comparable to CGIL in terms of metrics is a Swin transformer block, but the parameter count and FLOPs of the Swin transformer block are much higher than those of CGIL. While Fast-Vit-block has parameter count and FLOPs closest to CGIL, its accuracy is inferior to that of using CGIL.

Fig. 11 illustrates the comparison of segmentation results before and after using CGIL in the baseline model. It is evident that with the incorporation of CGIL, the baseline exhibits improved segmentation performance for large-scale objects, panoramic scenes, and large buildings. These experiments effectively demonstrate the efficiency and effectiveness of CGIL in the global decoding stage.

5) Effect of Backbone: In order to eliminate the influence of the backbone on the results, we conducted a series of replacement experiments on the Vaihingen dataset, using several common and widely used backbone models, including ConvNext-Tiny, ResNet50, ResNext50, and ResNest50. The

TABLE IV
PARAMETERS AND FLOPS OF DIFFERENT GLOBAL FEATURE EXTRACTION MODULES AT DIFFERENT STAGES, ALONG WITH MIOU, AND F1 SCORES ON THE VAIHINGEN DATASET

Method	Params(M)↓	Stage1	Stage2	Stage3	Stage4	mIoU(%)↑	F1(%) ↑
		FLOPs(G)↓	FLOPs(G)↓	FLOPs(G)↓	FLOPs(G)↓		
Vit-Block [26]	3.68	0.24	0.95	3.82	15.86	84.41	91.33
Swin-T-Block [27]	0.40	0.41	1.64	6.58	26.3	84.75	91.59
Mobile-Vit-Block [55]	0.51	0.53	2.11	8.42	33.7	84.43	91.36
Fast-Vit-Block [56]	0.22	0.22	0.89	3.55	14.21	84.63	91.44
CGIL (Ours)	**0.21**	**0.21**	**0.85**	**3.38**	**13.54**	**84.77**	**91.66**

The best results are indicated in bold black. In the table header, an upward arrow indicates that a higher evaluation value is better, and a downward arrow indicates that a higher evaluation value is better.

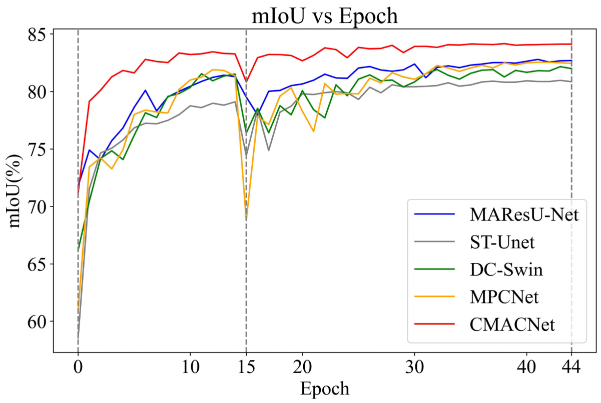


Fig. 12. Curve graph of mIoU variation with 44 epochs on the Vaihingen dataset for CMACNet and four other mainstream networks. Compared to the other networks, CMACNet reaches stability faster. The curve of CMACNet fluctuates less, indicating that the model can learn the features of the task and dataset more stably. The 15th epoch in the graph marks the beginning of the second cycle of the cosine annealing learning strategy, during which the model exhibits significant fluctuations but quickly returns to stability thereafter.

TABLE V
NUMBER OF PARAMETERS FOR DIFFERENT BACKBONES ON THE VAIHINGEN DATASET'S MIOU

Method	Backbone	Parameters(M)↓	mIoU(%) ↑
CMACNet	ResNet50	25.56	84.40
	ResNext50	25.03	84.44
	ResNest50	27.48	84.51
	ConvNext-Tiny	28.59	84.77

experimental results are shown in Table V. Observing the data in the table, it can be seen that although the ConvNext-Tiny model has a larger number of parameters, the CMACNet model performs best when using ConvNext-Tiny as the backbone. Therefore, we recommend ConvNext-Tiny as the preferred backbone model.

C. Comparative Experiments

In the experimental section of the article, the selected models for comparison are as follows: the "Bilateral Network"

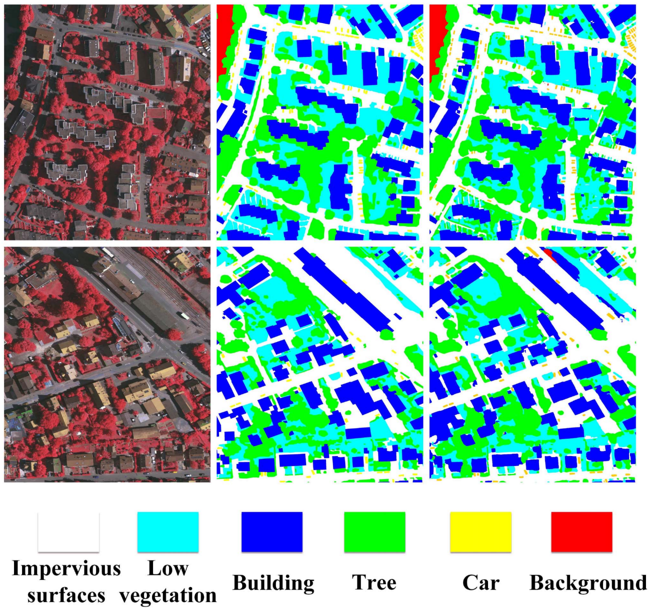


Fig. 13. Segmentation results of high-resolution images with IDs 2 and 10 on the Vaihingen dataset. The first column displays the original images, the second column displays the GT, and the third column displays the segmentation results of CMACNet.

ABCNet [57] with spatial and contextual pathways, MACU-Net [58] based on multiscale skip connections and asymmetric convolutions, multistage attention residual UNet (MAResU-Net) [40] with linear attention mechanism, attention-aggregated feature pyramid network A2-FPN [59], DC-Swin [60] Swin-transformer network with dense connection feature aggregation module, segmentation network ST-UNet [28] with global-local feature fusion scheme, and MPCNet [61], a network with multiscale prototype transformer decoder. Our model ultimately achieves higher accuracy on the ISPRS Vaihingen and ISPRS Potsdam datasets widely used for RS segmentation tasks compared to the aforementioned models.

Results on the Vaihingen Dataset: Table VI presents a numerical comparison of various semantic segmentation methods on the Vaihingen dataset. The research findings demonstrate that our proposed CMACNet achieved performance of 91.66% in average F1, 84.77% in mIoU, and 91.83% in OA. CMACNet

TABLE VI
COMPARISON OF SEGMENTATION RESULTS ON THE VAIHINGEN DATASET

Method	**F1(%)**					**Evaluation index**		
	Imp.Surf	**Building**	**Lowveg**	**Tree**	**Car**	**MeanF1(%)**	**mIoU(%)**	**OA(%)**
ABCNet [57]	88.13	90.23	76.71	87.21	68.72	82.20	70.58	85.62
MACU-Net [58]	90.70	92.40	81.90	89.37	82.53	87.38	77.85	88.61
MAResU-Net [40]	92.91	95.26	84.95	89.94	88.33	90.28	83.30	90.86
A2-FPN [59]	92.99	95.53	84.67	90.34	87.62	90.23	82.42	91.04
ST-Unet [28]	92.00	94.85	83.87	90.88	86.40	89.60	81.39	90.53
DC-Swin [60]	93.46	96.00	85.32	90.03	84.88	89.94	82.01	91.29
MPCNet [61]	92.76	95.50	84.70	90.40	90.44	90.76	83.27	90.93
CMACNet (Ours)	**93.55**	**96.08**	**86.03**	**91.20**	**91.41**	**91.66**	**84.77**	**91.83**

The best results are indicated in bold black.

TABLE VII
COMPARISON OF SEGMENTATION RESULTS ON THE POTSDAM DATASET

Method	**F1(%)**					**Evaluation index**		
	Imp.Surf	**Building**	**Lowveg**	**Tree**	**Car**	**MeanF1(%)**	**mIoU(%)**	**OA(%)**
ABCNet [57]	87.05	88.26	78.94	84.63	93.11	86.4	76.35	84.21
MACU-Net [58]	91.39	93.74	85.87	86.8	94.53	90.46	82.78	88.81
MAResU-Net [40]	92.38	95.83	86.65	88.44	96.13	91.89	85.22	90.28
A2-FPN [59]	92.85	95.84	87.17	88.76	96.13	92.15	85.66	90.68
ST-Unet [28]	92.77	96.90	86.45	88.11	95.80	92.01	85.46	90.53
DC-Swin [60]	93.26	96.86	87.74	88.68	95.50	92.41	86.10	91.16
MPCNet [61]	92.69	96.38	87.30	88.74	96.34	92.29	85.91	90.56
CMACNet (Ours)	**93.88**	**97.39**	**88.80**	**90.04**	**96.55**	**93.33**	**87.69**	**92.00**

The best results are indicated in bold black.

exhibited significantly superior performance in F1, OA, and mIoU compared to other networks. CMACNet not only outperformed the excellent convolutional lightweight network ABCNet but also surpassed the network DC-Swin based on the Swin transformer, which has strong global information representation capabilities. In addition, CMACNet also outperformed ST-Unet, which combines global and local features.

In addition to comparing the accuracy of various models, we also compared the convergence speed of mainstream models with our proposed CMACNet. The results are shown in Fig. 12, illustrating the trend of mIoU changes with Epoch during training on the Vaihingen dataset. We selected the results of the first 44 rounds of training as a reference because the learning rate adopted a cosine annealing strategy, leading to a significant drop in the metric at the 15th epoch in Fig. 12. From Fig. 12, it can be observed that the peak value of our network surpasses all other mainstream networks and stabilizes around the 10th epoch, indicating a faster convergence speed compared to other networks. This suggests that CMACNet has better fitting capability compared to other networks, indirectly indicating that our model is more capable of learning the key features of the task. In addition, compared to other models, our model exhibits smaller fluctuations, indicating that our model can more stably learn the features of the task and dataset during training.

Fig. 13 shows the segmentation results of CMACNet on images with IDs 2 and 10 from the Vaihingen dataset. Fig. 14 presents the segmentation results of CMACNet and other state-of-the-art networks on the Vaihingen dataset. From the second and fourth rows of Fig. 14, it can be seen that models with global modeling, such as ST-UNet and CMACNet, are more inclined to segment spatially correlated objects over large contiguous areas (e.g., buildings) compared to CNN-based networks such as ABCNet and MACU-Net and achieve better results. Although newer networks such as DC-Swin and MPCNet surpass previous networks in OA, they still do not perform satisfactorily in segmenting tree and low vegetation classes. Almost all network models can segment the car class, which we attribute to the distinct texture and spectral characteristics of cars in RS images

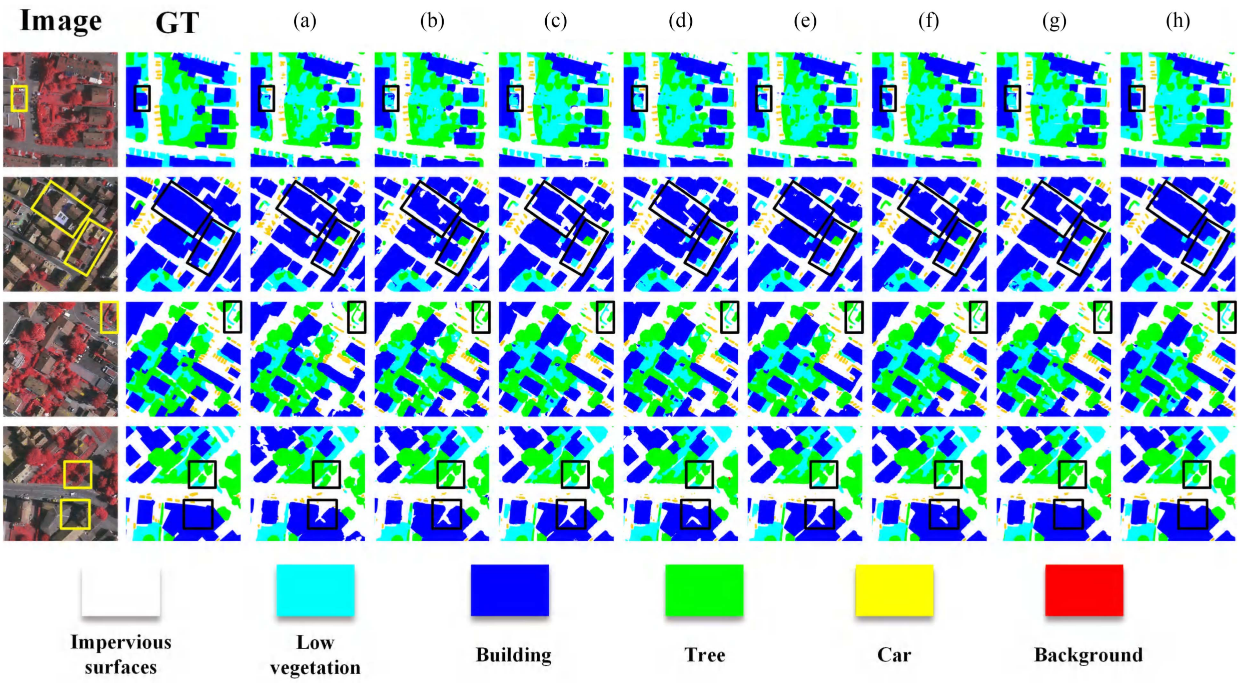


Fig. 14. Examples of segmentation results for different models on the Vaihingen dataset. (a) ABCNet. (b) MACU-Net. (c) MAResU-net. (d) A2-FPN. (e) ST-Unet. (f) DC-Swin. (g) MPCNet. (h) CMACNet. In areas with significant differences, regions are outlined with yellow boxes in the original image, and with black boxes in the GT and segmentation network's prediction results.

compared to other classes. In contrast, our network performs exceptionally well in distinguishing classes with high color similarity, such as trees and low vegetation, as highlighted in the second and third rows of Fig. 14. This further validates the effectiveness of our network in segmenting spectral-similar classes.

Results on the Potsdam Dataset: To comprehensively evaluate the network performance, we conducted further experiments on the Potsdam dataset. The experimental results are presented in Table VII, where CMACNet achieved significantly outstanding performance on the Potsdam test set: The average F1 score reached 93.33%, mIoU reached 87.69%, and OA index reached 92.00%, all surpassing other methods. Due to differences in dataset size and type, the segmentation accuracy on the Potsdam dataset is generally higher than that on the Vaihingen dataset. It is worth noting that on the Potsdam dataset, CMACNet achieved F1 scores for the lowvegetation and tree classes that were, respectively, 1.06% and 1.3% higher than the next best performing networks. This once again validates the superiority of our MCA in distinguishing categories with similar spectral characteristics.

As shown in Fig. 15, we also provide the overall segmentation images for IDs 3_13 and 4_15. Fig. 16 presents the segmentation results of the network models described in Table VII on the Potsdam dataset. From the second and fourth rows of Fig. 16, it can be observed that networks with global modeling capabilities, such as CMACNet, DC-Swin, and ST-UNet, outperform previous networks such as ABCNet and MACU-Net when handling contiguous area segmentation, such as building segmentation. From the second and third rows of Fig. 16, it can be seen that except for MPCNet and CMACNet, all other networks exhibit poor segmentation performance in environments with artificial influences, such as vegetation possibly present on buildings in Image 2 or farmland in Image 3. This also indicates that these networks lack strong interference resistance. Moreover, similar

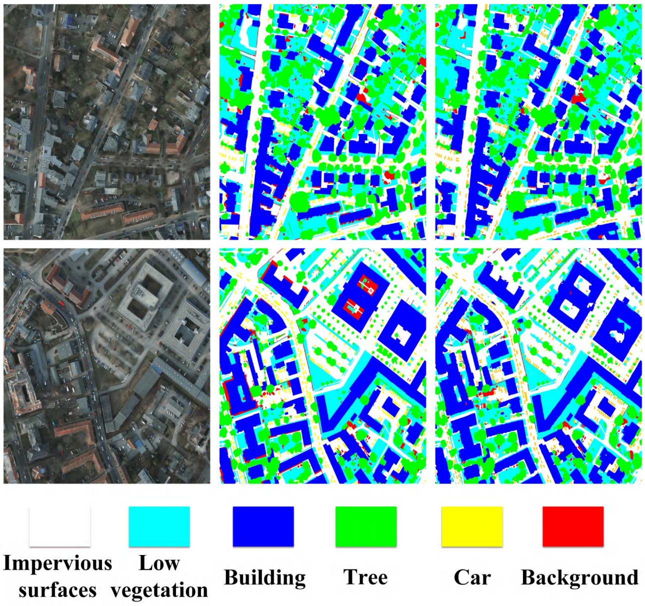


Fig. 15. Segmentation results of high-resolution images with IDs 3_13 and 4_15 on the Potsdam dataset. The first column displays the original images, the second column displays the GT, and the third column displays the segmentation results of CMACNet.

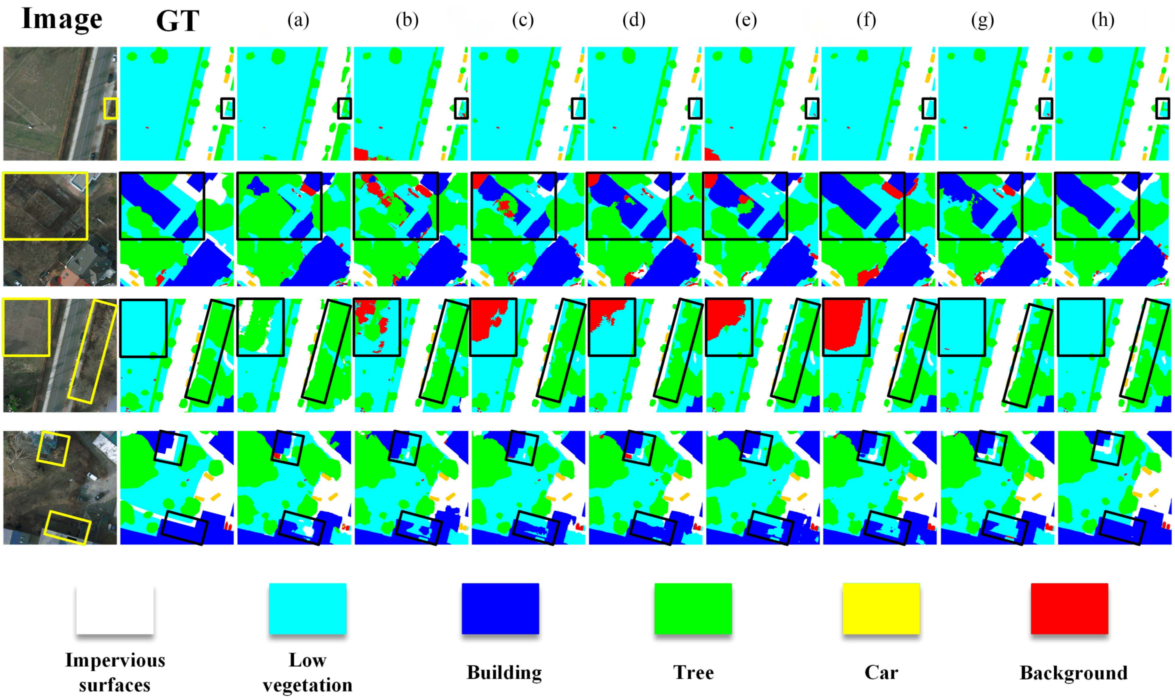


Fig. 16. Examples of segmentation results for different models on the Potsdam dataset. (a) ABCNet. (b) MACU-Net. (c) MAResU-Net. (d) A2-FPN. (e) ST-Unet. (f) DC-Swin. (g) MPCNet. (h) CMACNet. In areas with significant differences, regions are outlined with yellow boxes in the original image, and with black boxes in the GT and segmentation network's prediction results.

to their performance on the Vaihingen dataset, DC-Swin and MPCNet show good performance but still lack sensitivity to classes with similar spectral characteristics. The third row of Fig. 16 demonstrates that CMACNet has a significant advantage in distinguishing classes with similar spectral characteristics, such as trees and low vegetation. In addition, the fourth row of Fig. 16 shows that CMACNet outperforms other networks when dealing with nonhomogeneous color-similar objects.

By conducting experiments on the Vaihingen and Potsdam datasets, we have demonstrated that CMACNet performs excellently on both datasets, showing significant segmentation performance. CMACNet outperforms other methods in terms of average F1 score, mIoU, and OA index, as well as model generalization capability, particularly excelling in distinguishing classes with similar spectral characteristics. Furthermore, the experimental results indicate that CMACNet also performs excellently in segmenting both small and large objects. Therefore, we strongly recommend CMACNet as an effective segmentation model, especially suitable for handling categories with similar spectral characteristics.

V. Limitations and Future Prospects

Despite the advantages demonstrated by our CMACNet on experimental data, and its partial resolution of the challenging segmentation of spectrally similar classes, it still exhibits certain limitations. For instance, our model currently applies only to semantic segmentation of urban RS images, without extending to other RS visual tasks such as road segmentation, parcel segmentation, or agricultural and forestry segmentation. Perhaps our network would perform better in scenarios with more spectrally similar classes, such as agricultural and forestry scenes. However, due to space constraints, we did not further explore the model's performance in segmenting agricultural and forestry scene images. In future work, we plan to investigate superior segmentation methods by combining RS image characteristics and deep learning theories to optimize our network for a broader range of RS visual tasks. In addition, we will focus on model compression to design lighter and more efficient models.

VI. Conclusion

This article aims to address the challenge of segmenting land cover with similar spectral characteristics by designing a feature mapping correlation attention mechanism. We propose CMACNet, which enhances the versatility of local multiscale information through autoencoding and integrates it with feature mapping correlation attention before combining it with global information. Specifically, we first employ an autoencoder to enhance feature generalization by reconstructing the original image and introduce multiscale information to obtain a more comprehensive feature representation. Then, to tackle the challenge of segmenting land cover with similar spectral characteristics, we extend this problem to the feature domain and design a correlated mapping attention mechanism. This mechanism not only dynamically adjusts the correlations between feature mappings but also assigns weights to features based on their correlations, enabling the model to focus more on regions with similar spectral characteristics during this stage. Finally, for efficient global decoding, we design an efficient

CGIL that remaps comprehensive features from earlier stages globally to establish long-range dependencies. Experimental results demonstrate the superiority of our network architecture and the effectiveness of each component. We hope to inspire more researchers to propose practical and effective solutions to address the segmentation challenge of spectrally similar land cover and encourage further exploration of the potential and applications of feature mapping correlation in the field of RS.

References

[1] R. Li, S. Zheng, C. Duan, L. Wang, and C. Zhang, "Land cover classification from remote sensing images based on multi-scale fully convolutional network," *Geo-Spatial Inf. Sci.*, vol. 25, no. 2, pp. 278–294, 2022.

[2] D. Marcos, M. Volpi, B. Kellenberger, and D. Tuia, "Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 96–107, 2018.

[3] J. Yao, B. Zhang, C. Li, D. Hong, and J. Chanussot, "Extended vision transformer (ExViT) for land use and land cover classification: A multimodal deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jun. 9, 2023, Art. no. 5514415.

[4] J. Xing, R. Sieber, and T. Caelli, "A scale-invariant change detection method for land use/cover change research," *ISPRS J. Photogrammetry Remote Sens.*, vol. 141, pp. 252–264, 2018.

[5] I. de Gélis, S. Lefèvre, and T. Corpetti, "Siamese KPconv: 3D multiple change detection from raw point clouds using deep learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 197, pp. 274–291, 2023.

[6] A. Samie et al., "Examining the impacts of future land use/land cover changes on climate in Punjab Province, Pakistan: Implications for environmental sustainability and economic growth," *Environ. Sci. Pollut. Res.*, vol. 27, pp. 25415–25433, 2020.

[7] F. Chen, H. Balzter, F. Zhou, P. Ren, and H. Zhou, "DGNet: Distribution guided efficient learning for oil spill image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jan. 30, 2023, Art. no. 4201317.

[8] D. Griffiths and J. Boehm, "Improving public data for building segmentation from convolutional neural networks (CNNs) for fused airborne Lidar and image data using active contours," *ISPRS J. Photogrammetry Remote Sens.*, vol. 154, pp. 70–83, 2019.

[9] P. Shamsolmoali, M. Zareapoor, H. Zhou, R. Wang, and J. Yang, "Road segmentation for remote sensing images using adversarial spatial pyramid networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4673–4688, Jun. 2021.

[10] M. C. A. Picoli et al., "Big earth observation time series analysis for monitoring Brazilian agriculture," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 328–339, 2018.

[11] Y. Shen, J. Chen, L. Xiao, and D. Pan, "Optimizing multiscale segmentation with local spectral heterogeneity measure for high resolution remote sensing images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 157, pp. 13–25, 2019.

[12] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, 2005.

[13] Y. Zhang and T. Chen, "Efficient inference for fully-connected CRFs with stationarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 582–589.

[14] U. Maulik and D. Chakraborty, "Remote sensing image classification: A survey of support-vector-machine-based advanced techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 1, pp. 33–52, Mar. 2017.

[15] D. Hong et al., "SpectralGPT: Spectral remote sensing foundation model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5227–5244, 2024.

[16] M. Zhang, W. Li, R. Tao, H. Li, and Q. Du, "Information fusion for classification of hyperspectral and Lidar data using IP-CNN," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jul. 13, 2021, Art. no. 5506812.

[17] X. Liu et al., "Deep multiview union learning network for multisource image classification," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 4534–4546, Jun. 2022.

[18] X.-Y. Tong et al., "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, 2020, Art. no. 111322.

[19] 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.

[20] 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, Dec. 2017.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervention*, 2015, pp. 234–241.

[22] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support: 4th Int. Workshop, 8th Int. Workshop, Held Conjunction MICCAI*, 2018, vol. 4, pp. 3–11.

[23] X. Wang, Z. Hu, S. Shi, M. Hou, L. Xu, and X. Zhang, "A deep learning method for optimizing semantic segmentation accuracy of remote sensing images based on improved UNet," *Sci. Rep.*, vol. 13, no. 1, 2023, Art. no. 7600.

[24] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[25] K.-H. Liu and B.-Y. Lin, "MSCSA-Net: Multi-scale channel spatial attention network for semantic segmentation of remote sensing images," *Appl. Sci.*, vol. 13, no. 17, 2023, Art. no. 9491.

[26] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[27] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[28] X. He, Y. Zhou, J. Zhao, D. Zhang, R. Yao, and Y. Xue, "Swin transformer embedding UNet for remote sensing image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 19, 2022, Art. no. 4408715.

[29] L. Fan, Y. Zhou, H. Liu, Y. Li, and D. Cao, "Combining Swin transformer with UNet for remote sensing image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Nov. 1, 2023, Art. no. 5530111.

[30] J. C. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[32] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11534–11542.

[33] K. Wang and D. Ming, "Road extraction from high-resolution remote sensing images based on spectral and shape features," *Proc. SPIE*, vol. 7495, pp. 968–973, 2009.

[34] D. Li, G. Zhang, Z. Wu, and L. Yi, "An edge embedded marker-based watershed algorithm for high spatial resolution remote sensing image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2781–2787, Oct. 2010.

[35] X. Huang, W. Yuan, J. Li, and L. Zhang, "A new building extraction postprocessing framework for high-spatial-resolution remote-sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 654–668, Feb. 2017.

[36] J. Wang, Z. Zheng, A. Ma, X. Lu, and Y. Zhong, "LoveDA: A remote sensing land-cover dataset for domain adaptive semantic segmentation," in *Proc. NeurIPS*, 2021, pp. 1–12.

[37] R. Kemker, C. Salvaggio, and C. Kanan, "Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 60–77, 2018.

[38] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 152, pp. 166–177, 2019.

[39] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[40] R. Li, S. Zheng, C. Duan, J. Su, and C. Zhang, "Multistage attention ResU-Net for semantic segmentation of fine-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Mar. 15, 2021, Art. no. 8009205.

[41] Y. Wang, L. Gu, T. Jiang, and F. Gao, "MDE-UNet: A multitask deformable UNet combined enhancement network for farmland boundary segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, Mar. 6, 2023, Art. no. 3001305.

[42] W. Qiu, L. Gu, F. Gao, and T. Jiang, "Building extraction from very high-resolution remote sensing images using refine-UNet," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, Feb. 9, 2023, Art. no. 6002905.

[43] H. Jung, H.-S. Choi, and M. Kang, "Boundary enhancement semantic segmentation for building extraction from remote sensed image," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Sep. 2, 2021, Art. no. 5215512.

[44] C. Zheng, Y. Chen, J. Shao, and L. Wang, "An MRF-based multigranularity edge-preservation optimization for semantic segmentation of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Feb. 24, 2021, Art. no. 8008205.

[45] B. Sui, Y. Cao, X. Bai, S. Zhang, and R. Wu, "BIBED-Seg: Block-in-block edge detection network for guiding semantic segmentation task of high-resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1531–1549, Jan. 17, 2023.

[46] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.

[47] Z. Chen, G. Wu, H. Gao, Y. Ding, D. Hong, and B. Zhang, "Local aggregation and global attention network for hyperspectral image classification with spectral-induced aligned superpixel segmentation," *Expert Syst. Appl.*, vol. 232, 2023, Art. no. 120828.

[48] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 9423–9433.

[49] L. Ding, H. Tang, and L. Bruzzone, "LANet: Local attention embedding to improve the semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 426–435, Jan. 2021.

[50] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7262–7272.

[51] H. Cao et al., "Swin-UNet: UNet-like pure transformer for medical image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 205–218.

[52] Y. Liu, Y. Zhang, Y. Wang, and S. Mei, "Rethinking transformers for semantic segmentation of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Aug. 4, 2023, Art. no. 5617515.

[53] T. Xiao, Y. Liu, Y. Huang, M. Li, and G. Yang, "Enhancing multiscale representations with transformer for remote sensing image semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Mar. 10, 2023, Art. no. 5605116.

[54] 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. 11976–11986.

[55] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.

[56] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "FastViT: A fast hybrid vision transformer using structural reparameterization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 5785–5795.

[57] 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. Photogrammetry Remote Sens.*, vol. 181, pp. 84–98, 2021.

[58] R. Li, C. Duan, and S. Zheng, "MACU-Net semantic segmentation from high-resolution remote sensing images," 2020, *arXiv:2007.13083*.

[59] R. Li, L. Wang, C. Zhang, C. Duan, and S. Zheng, "A2-FPN for semantic segmentation of fine-resolution remotely sensed images," *Int. J. Remote Sens.*, vol. 43, no. 3, pp. 1131–1155, 2022.

[60] L. Wang, R. Li, C. Duan, C. Zhang, X. Meng, and S. Fang, "A novel transformer based semantic segmentation scheme for fine-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Jan. 14, 2022, Art. no. 6506105.

[61] Q. Wang, X. Luo, J. Feng, G. Zhang, X. Jia, and J. Yin, "Multi-scale prototype contrast network for high-resolution aerial imagery semantic segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jul. 6, 2023, Art. no. 5615114.

Yunsong Yang received the bachelor's degree in computer science and technology from the School of Computer Science, University of South China, Hunan, China, in 2021. He is currently working toward the master's degree in computer science and technology with the School of Computer Science and Technology, Shandong University of Finance and Economics, Yantai, China.

His research interests include computer graphics, computer vision, and image processing.

Genji Yuan received the bachelor's degree in electronic science and technology from Shandong Technology and Business University, Yantai, China, in 2017, the master's degree in computer science and technology from Shandong Technology and Business University, Yantai, Shandong, in 2020, and the Ph.D. degree in computer science and technology from Qingdao University, Qingdao, China, in 2024.

His research interests include computer vision, pattern recognition, machine learning, data analysis, and intelligent transportation system.

Jinjiang Li received the B.S. and M.S. degrees in computer science from the Taiyuan University of Technology, Taiyuan, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science from Shandong University, Jinan, China, in 2010.

From 2004 to 2006, he was an Assistant Research Fellow with the Institute of Computer Science and Technology, Peking University, Beijing, China. From 2012 to 2014, he was a Postdoctoral Fellow with Tsinghua University, Beijing. He is currently a Professor with the School of Computer Science and Technology, Shandong Technology and Business University. His research interests include image processing, computer graphics, computer vision, and machine learning.