Masked Feature Modeling for Generative Self-Supervised Representation Learning of High-Resolution Remote Sensing Images

Shiyan Pang, Hanchun Hu, Zhiqi Zuo, Jia Chen, and Xiangyun Hu

Abstract—**Intelligent interpretation of remote sensing images using deep learning is heavily reliant on large datasets, and models trained in one domain often struggle with crossdomain application. Pretraining the backbone network via masked image modeling can effectively diminish this reliance on extensive sample data, thereby reducing crossdomain transfer obstacles. However, current masked image models typically employ a pure Transformer architecture, which may not fully capitalize on low-level features. To address these issues, this article proposes masked feature modeling (MFM), a methodology for the generative self-supervised learning of high-resolution remote sensing images that combines convolutional neural network (CNN) and Transformer architectures. This methodology has several advantages: 1) The hybrid CNN + Transformer architecture not only retains the advantages of the local feature representation of the CNN architecture but also has the full-text information modeling capabilities of the Transformer architecture; 2) the feature extraction network outputs multiscale features, and it is easier to add upsampling and a skip connection to improve the accuracy of the downstream dense prediction task; and 3) the pretrained MFM can be applied to various downstream tasks through fine-tuning with limited samples. The publicly available WHU and Massachusetts Building Datasets are used to verify the effectiveness of the proposed method. Extensive experiments involving main properties of the MFM for generative self-supervised learning, fine-tuning the MFM on the downstream semantic segmentation task, and comparisons with the other state-of-the-art generative self-supervised learning algorithms show that, through the combined advantages of the CNN and Transformer architectures, the proposed method has better feature extraction capability and higher accuracy on downstream tasks such as semantic segmentation.**

Index Terms—**Generative self-supervised learning (SSL), masked feature modeling (MFM), remote sensing, semantic segmentation, transformer.**

I. Introduction

With the continuous development of remote sensing technology, the number of remote sensing images captured by high-resolution satellites has surged. Optimal interpretation of these high-resolution remote sensing images has become a hotspot of research. In recent years, intelligent interpretation technology for remote sensing images has developed rapidly due to in-depth research on deep learning. In various deep learning-based tasks, the backbone network is used for feature extraction, and the extracted features are then sent to downstream tasks, such as image classification, target detection, and image semantic segmentation. In this process, due to the high number of parameters of the backbone network and the high training difficulty, it is usually necessary to load pretraining parameters. These pretraining parameters are obtained by training with larger datasets (such as ImageNet) and have strong feature representation capabilities, which greatly reduces the training difficulty of the network. In the field of remote-sensing image processing, when faced with new tasks without sufficient ground-truth information, it is necessary to obtain the pretraining parameters of the backbone network through self-supervised learning (SSL) to improve the feature extraction capability of the backbone network.

SSL has become a popular research topic in deep learning. Unlike supervised learning, SSL does not require ground-truth information. Its supervision information comes from itself or its representation characteristics, and the feature extraction capability of the model is improved by designing auxiliary tasks. In computer vision, early examples of SSL have been achieved through contrastive learning, in which data is first augmented to obtain positive and negative samples and then compared with positive and negative samples in the latent feature space. A contrastive loss function is set to reduce the distance from positive samples and extend the distance from negative samples, so that the model learns the feature representation of the data. In natural language processing (NLP), generative SSL methods, of

Manuscript received 15 November 2023; revised 12 February 2024; accepted 31 March 2024. Date of publication 5 April 2024; date of current version 22 April 2024. This work was supported in part by the Special Fund of Hubei Luojia Laboratory under Grant 220100028, in part by the Knowledge Innovation Program of Wuhan-Shuguang Project under Grant 2022010801020281, in part by the Ministry of Education of the People's Republic of China under Grant 22YJC880058, in part by the Hubei Provincial Natural Science Foundation under Grant 2021CFB539, in part by the Fundamental Research Funds for the Central Universities under Grant CCNU22QN011 and Grant CCNU22QN019, in part by the University-level Educational Reformation Research Project for Undergraduate Education, Central China Normal University, China under Grant 202143, in part by the Teaching Innovation Research Project of Faculty of Artificial Intelligence in Education, Central China Normal University under Grant ZNXBJY202114, and in part by the 2022 Experimental Teaching Research Project of Liberal Arts Comprehensive Experimental Teaching Center under Grant WZ2022-02. *(Corresponding author: Xiangyun Hu.)*

Shiyan Pang, Hanchun Hu, and Jia Chen are with the Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, Hubei 430079, China (e-mail: pangsy@ccnu.edu.cn; 1224897618@qq.com; jc@ccnu.edu.cn).

Zhiqi Zuo is with the College of Informatics, Huazhong Agricultural University, Wuhan 430070, China (e-mail: zuo668@mail.hzau.edu.cn).

Xiangyun Hu is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei 430079, China, and also with the Institute of Artificial Intelligence in Geomatics, Wuhan University, Wuhan, Hubei 430079, China (e-mail: huxy@whu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3385420

which masked language modeling in BERT is a representative example, have achieved great success. These methods allow the model to learn to predict the specific content of random masked words. This approach has now become the standard pretraining paradigm in NLP. Inspired by BERT, this simpler and more efficient method of generative SSL has also become the focus of current SSL research in computer vision. Currently, many masked image models that follow the mask-reconstruction paradigm have been proposed, such as masked autoencoder (MAE) and BEiT. These methods divide the image into blocks, mask some of them, and then use the encoded unmasked blocks to predict the masked ones, which enhances the model's feature extraction capability.

However, current masked image modeling methods are primarily based on the Vision Transformer (ViT) architecture. Compared with convolutional neural networks (CNNs), a pretrained ViT model often has weaker local feature extraction capabilities in downstream tasks and is prone to losing image details during the global modeling process. Currently, an increasing number of studies have demonstrated that combining CNNs with Transformer architecture not only preserves the inherent locality of convolution but also addresses the issue that convolution-based networks struggle with effectively modeling long-range relationships. This combined approach has shown promising results in dense prediction tasks such as semantic segmentation.

To pretrain the parameters of a hybrid CNN-Transformer model, in this study, we introduce masked feature modeling (MFM), a generative SSL method suitable for remote sensing images. MFM masks the high-order features extracted by the CNN and predicts the low-level features of the image. The contributions of this work are summarized as follows.

1) We propose a MFM for generative SSL based on a hybrid architecture of CNN and Transformer. The architecture fully combines the advantages of CNN's local feature representation and Transformer's global modeling, and has stronger feature extraction capabilities. In contrast with previous methods of masked image modeling, the proposed method directly masks some of the features extracted by the CNN and predicts the original image from the unmasked features. This masking method is more suitable for the hybrid CNN + Transformer architecture, and it more easily obtains high-order features with good representation.
2) The hybrid CNN + Transformer architecture outputs multiscale image features, and it is easier to add upsampling and "skip connection" modules to build a U-Net [1] architecture network that is suitable for dense prediction tasks and improves the accuracy of the downstream semantic segmentation of remote sensing images.
3) The proposed method is more accurate than other generative SSL methods on two publicly available datasets, namely, the WHU and Massachusetts Building datasets.

The rest of this article is organized as follows. Section II describes the related work of SSL. Section III gives the details of our proposed method. The experimental results and analysis are given in Section IV. Finally, Section V concludes this article.

II. Related Work

SSL can effectively alleviate the high dependence on annotated samples and has become a research hotspot in the field of remote sensing. SSL is widely used in remote-sensing scene classification [2], [3], [4], [5], [6], [7], [8], [9], image classification [10], [11], [12], [13], [14], [15], [16], [17], [18], semantic segmentation [19], [20], [21], [22], change detection [23], and target recognition [24], [25]. At present, there are two main representative SSL schemes, namely, contrastive and generative SSL methods.

A. Contrastive SSL Methods

In contrastive SSL methods, multiple augmented views of the same sample are first obtained through data augmentation. Here, different views of the same sample are taken as similar pairs, while views of different samples are taken as dissimilar pairs. A contrastive loss function is then constructed to keep similar pairs close in the latent feature space and dissimilar pairs far apart in the latent space. This enables the learning of effective image high-order feature representations from unlabeled samples. For the scene classification task, Kang et al. [2] proposed a new unsupervised deep metric learning model called the spatially augmented momentum contrast model, or SauMoCo, to characterize unlabeled remote sensing scenes. Jung et al. [3] introduced a remote-sensing contrastive SSL method with smoothed representation based on the SimCLR framework, which uses multiple input images and averages their representations. Li et al. [4] introduced the end-to-end self-supervised contrastive learning-based metric learning network, or SCL-MLNet, for few-shot remote-sensing scene classification. Wang et al. [5] proposed a few-shot remote-sensing scene classification named the class-shared sparsePCA classifier, or CSSPCA, to train a feature extractor in the case of few training samples. Xiao et al. [6] proposed a simple and effective SSL algorithm, named Lite-SRL, for the scene classification task, designing a lightweight contrastive learning structure and adopting the stop-gradient operation to reduce the calculation cost. For synthetic aperture radar (SAR) and optical images, Stojnić and Risojević [8] adopted contrastive multiview coding for self-supervised pretraining to obtain the remote-sensing scene representation and obtained better results on the downstream classification task for remote sensing images than for natural scenes. In addressing the problem of limited and small hyperspectral samples, various researchers [10], [11], [12], [13], [14], [15] have used different augmentation methods to construct pairs of augmented views from a hyperspectral sample, conducted contrastive learning in the pretraining stage, and finally used the learned features for hyperspectral image classification. Furthermore, Guan and Lam [16] studied a crossdomain contrastive learning framework to extract domain-invariant information in a crossdomain discrimination task. In terms of SAR and optical data fusion, Chen and Bruzzone [17] proposed a self-supervised framework for SAR–optical data fusion and land-cover mapping tasks. In this framework, a multiview contrastive loss at image level and super pixel level was first used to fuse the SAR and optical images, and each pixel was then assigned a land-cover class by the joint use of pretrained features and spectral information of the image itself. For PolSAR image classification, Zhang et al. [18] explored the learning of transferrable representations from unlabeled PolSAR data through CNN. In terms of high-resolution image semantic segmentation, Li et al. [19] proposed a global style and local matching contrastive learning network, named GLCNet, for the semantic segmentation of remote sensing images. Gao et al. [20] proposed an unsupervised domain adaptation framework for the semantic segmentation of remote sensing images, which is based on the consistency principle. Cha et al. [22] proposed a multimodal algorithm for SAR semantic segmentation, leveraging both electro-optical imagery and SAR imagery alongside a label mask. In terms of change detection, Saha et al. [23] focused on the combination of images acquired by optical and SAR sensors and proposed a multisensor change detection method. This method used only unlabeled bitemporal images of objects, adopting deep clustering and contrastive learning methods to train the network in a self-supervised manner. In terms of target detection, the authors in [24], [25], and [26] have effectively reduced the number of labeled samples required for target detection through contrastive SSL. In addition, Manas et al. [27] introduced seasonal contrast, a method that enhanced contrastive learning by treating images of the same location at different times as similar pairs. This approach led to superior performance over ImageNet pretraining and other self-supervised methods in multiple downstream tasks.

B. Generative SSL Methods

The generation models mainly consist of flow-based models, autoregressive (AR) models, and masked image models.

Flow-based models estimate complex high-dimensional densities from data by transforming simple base distributions through the use of invertible transformations with a deterministic mapping. This allows them to learn a sequence of transformations that gradually convert a simple distribution, such as a standard Gaussian, into a complex target distribution that matches the data. For example, NICE [28] designs affine transformations to parameterize the data distribution.

AR models can predict new data based on previous data, which can model context dependence well. PixelRNN [29] and PixelCNN [30] model images pixel by pixel using RNNs and CNNs, respectively. These models assume that there is a dependency between pixels, and that the current pixel value is related to the previous pixel value in the process of generating the image, which is expressed in the form of an AR model.

The current mainstream generative SSL is the masked image model (MIM), which this article primarily investigates. MIM realizes feature learning by reconstructing the original data from the corrupted input. BEiT [31], which adopted the ViT architecture as the backbone network, was the first MIM used in computer vision. In this model, the image was first represented as a sequence of discrete tokens, a fraction of image patches were then masked randomly, and the visual tokens of the masked image patches were predicted through masked image modeling. Subsequently, the algorithm of MAE proposed by He et al. [32] achieved some success in computer vision. This algorithm has three notable characteristics. First, the algorithm adopts the ViT architecture to directly mask and position the original image, instead of converting the image blocks into discrete tokens, which is simpler and more effective. Second, the strategy of randomly masking most image blocks reduces information redundancy, resulting in a challenging self-supervision task, such that the trained model goes beyond low-level image statistics. Third, the algorithm has an asymmetric encoder–decoder structure, which only performs encoder operations on visible image blocks (i.e., blocks without masks) and, thus, reduces the time and memory costs. Subsequently, Cong et al. [33] proposed SatMAE, a pretraining framework for temporal or multispectral satellite imagery based on MAE. SatMAE includes a temporal embedding along with independently masking image patches across time, and encoding multispectral data as groups of bands with distinct spectral positional encodings. On the basis of the MAE algorithm, Xue et al. [34] proposed a self-supervised feature learning architecture for multimodal remote sensing imagery, which extracts meaningful high-level feature representations from multiview data and combines the learned features with the corresponding spectral information for land-cover classification. The SimMIM algorithm [35] uses the same masking strategy, the differences being that the block to be masked is replaced by a learnable vector, the encoder processes all blocks, and the decoder uses simple linear layers; this algorithm also performs well. The subsequently developed MaskFeat algorithm [36] uses the histogram of oriented gradient features of the image as the prediction target and has been shown to be more accurate than the MAE and SimMIM algorithms on the ImageNet dataset. This demonstrates that some handcrafted features may be more helpful for machines to understand the image information. The context autoencoder (CAE) [37] strictly separates the representation learning (encoding) role from the pretext task completion role, such that the encoder is only responsible for learning image features in the process of SSL and the generalization ability of the model on downstream tasks is improved. Due to the abundance of small targets in remote sensing images, Sun et al. [38] introduced a masking strategy that reserves random pixels within masked regions to preserve small target information. Liu et al. [39], on the other hand, proposed a self-supervised multilevel feature fusion method that enhances low-frequency semantic information capture by using shallow, low-level features to aid pixel reconstruction.

The above MIMs learn meaningful high-level feature representations from images, reducing the use of annotated samples on downstream tasks. However, the current MIMs are basically pure Transformer frameworks, and Transformers are born with a global self-attention mechanism, but due to insufficient low-level features, resulting in limited local localization capabilities. This article, therefore, presents the design of MFM for the generative SSL of high-resolution remote sensing images based on the fusion of CNN and Transformer architecture. This method fully combines the advantages of CNN's local feature representation and Transformer's global modeling, and obtains better results.

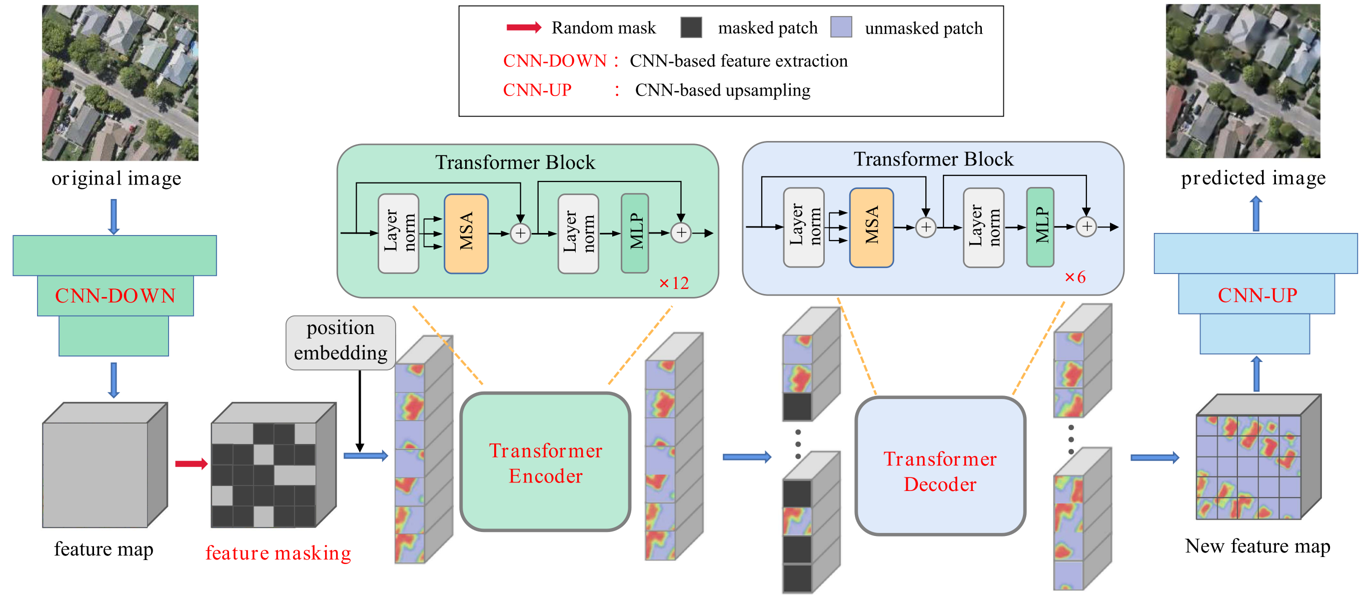


Fig. 1. Network structure of the proposed MFM for the generative SSL of high-resolution remote sensing images.

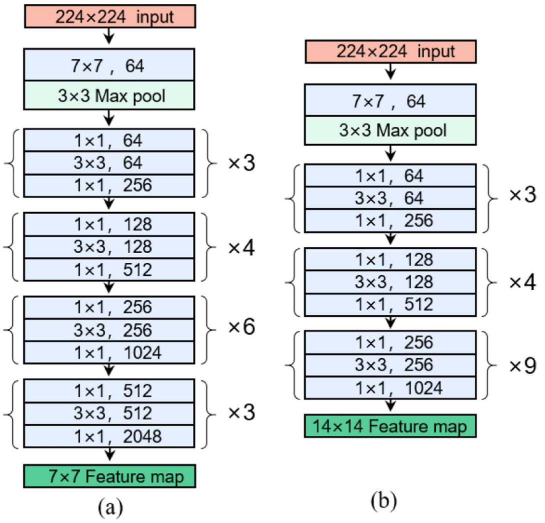
III. Methodology

This article proposes an MFM for the generative SSL of high-resolution remote sensing imagery. The proposed method comprises six parts. Among them, Parts A–D are the components of network structure of the proposed MFM, as shown in Fig. 1, Part E is the loss function, and Part F is the model evaluation, which is realized by the fine-tuning of the MFM on the downstream task. Each part is described as follows.

A. CNN-Based Feature Extraction

The aim of the convolution feature extraction module is to obtain small-size, high-dimensional convolutional features from the original image. In this specific implementation, we use the classic ResNet50 as a feature extraction module, and we use the hierarchical structure of convolution kernels to learn local spatial context information of varying complexity, such as information from simple low-level edges and textures to high-level semantic patterns. This multilevel local feature helps the model to understand the image and plays an important role in pixelwise prediction tasks such as semantic segmentation.

In our method, a slight change is made to the structure of ResNet50, such that the size of the feature map output by ResNet50 is increased from the original size of 7 × 7 to 14 × 14. Dividing the 14 × 14 feature map into patches yields the same number of patches (e.g., 196) as in the ViT architecture, and the use of finer patches avoids the loss of too much local information. Here, we let x be the input original image. Its height (e.g., 224) and width (e.g., 224) are denoted W and H, respectively, and it has three channels. The height (e.g., 14), width (e.g., 14), and number of channels of the feature map obtained after feature extraction through ResNet50 are denoted $H/16, W/16$, and C, respectively. Here, the feature map is denoted $g(x)$. The relationship between $g(x)$ and x is then expressed as follows:



(a) (b)

Fig. 2. ResNet50 with different structures. (a) Original ResNet50 structure [40]. (b) ResNet50 structure used in this study.

$$g(x) = \text{Resnet50}(x) \tag{1}$$

where $x \in \mathbb{R}^{H\times W\times 3}$, $g(x) \in \mathbb{R}^{H/16\times W/16\times C}$, and ResNet50 is the CNN-based feature extraction module. The detailed structure is shown in Fig. 2.

B. Feature Masking

Unlike other generative SSL methods that directly mask the original image, our method masks the high-order convolutional features of the image. For the 14 × 14 feature map extracted by the above CNN, the feature map is first divided into patches and flattened into a sequence of feature vectors with a length of 196, and the feature vectors in the sequence correspond one-to-one with the patches divided by the feature map. Our masking method follows the masking strategy of the MAE algorithm in that sequences of length 196 are randomly sampled according to a certain mask ratio. The sampled feature vectors are masked (e.g., deleted), and the unsampled remaining feature vectors form an unmasked sequence, which is considered the unmasked part of the feature map. In this masking method, in the Transformer-based feature encoding stage, the encoder only processes the unmasked blocks, which greatly accelerates the training and reduces the computational cost. In addition, because the Transformer block in the encoder only performs feature encoding, the task of predicting the original image is completed by the decoder. Separating the representation learning of feature extraction from the pretext task of predicting pixels improves the feature extraction capability of the encoder.

The flattened feature sequence P is calculated as follows:

$$P = \text{flatten}\left(\text{Conv}_{1\times1}\left(g(x)\right)\right) \tag{2}$$

where $\text{Conv}_{1\times1}$ represents a convolution with a convolution kernel size of 1 × 1 and a stride of 1, and flatten converts the 2-D 14 × 14 feature map to a 1-D sequence of length 196. The flattened feature sequence $P = \{p_1, p_2, p_3, \ldots, p_n, n = H/16 \times W/16\} \in \mathbb{R}^{n\times C}$. After P is acquired, the generated mask matrix is used for feature masking. The masked feature sequence P' is calculated as follows:

$$P' = P \times M^T \tag{3}$$

$$M = \left\{m_1, m_2, m_3, \ldots, m_n, n = \frac{H}{16} \times \frac{W}{16}\right\}, m \in \{0, 1\} \tag{4}$$

where M is the mask matrix, which is a 1-D matrix with elements of 0 and 1 representing masked patches and unmasked patches, respectively. The numbers of zeroes and ones are calculated using the mask ratio, and the positions of these values in the mask matrix are randomly assigned. The final unmasked sequence P'' is calculated as follows:

$$P'' = \{p_{i1}, p_{i2}, p_{i3}, \ldots, p_{ik}\}, p_i \in P' \cap p_i \neq 0 \tag{5}$$

where any element p_i in $\{p_{i1}, p_{i2}, p_{i3}, \ldots, p_{ik}\}$ corresponds to one of the patches divided by the feature map, and k in the subscript ik is the number of unmasked patches in the feature map.

C. Transformer-Based Encoder

The Transformer-based encoder is the same as the standard ViT and comprises a series of stacked Transformer blocks. This Transformer block only processes unmasked sequences. After adding position embedding, the unmasked sequences are then encoded to further extract more important high-order features with full-text information. Due to the large mask ratio in the experiment, this is a challenging task for the encoder. After sufficient training, the encoder develops a strong feature extraction ability.

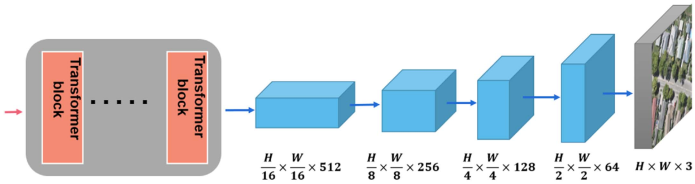

Transformer block
......
Transformer block
$\frac{H}{16} \times \frac{W}{16} \times 512$
$\frac{H}{8} \times \frac{W}{8} \times 256$
$\frac{H}{4} \times \frac{W}{4} \times 128$
$\frac{H}{2} \times \frac{W}{2} \times 64$
$H \times W \times 3$

Fig. 3. Structure of the decoder.

The Transformer block used in this study is the standard Transformer block from the ViT architecture, which comprises two LayerNorm layers, a multihead attention mechanism, and a multilayer perceptron (MLP) layer. The input and output feature sizes of each Transformer block are the same. The Transformer-based encoder comprises a series of stacked Transformer blocks. The calculation of each Transformer block is as follows:

$$Z_1 = P'' + \text{MultiHead}\left(\text{LayerNorm}\left(P''\right)\right) \tag{6}$$

$$Z_2 = Z_1 + \text{MLP}\left(\text{LayerNorm}\left(Z_1\right)\right) \tag{7}$$

where MultiHead represents the multihead attention mechanism, LayerNorm is a function that normalizes the data of the tensor, MLP is a multilayer perceptron, Z_1 is the result calculated through the Transformer attention mechanism, and Z_2 is the output of the Transformer block. P'' is the unmasked sequence obtained in Part B, P'' is only used as input to the first Transformer block, and the input of the remaining Transformer blocks is the output of the previous block. The calculation of MultiHead is as follows:

$$\text{MultiHead}(p) = \text{Concat}\left(\text{head}_1(p), \ldots, \text{head}_h(p)\right) \tag{8}$$

where Concat represents feature concatenation, head_i is a single attention head, MultiHead is the multihead attention obtained by concatenating multiple attention heads, and p is the input vector of the multihead attention. Here, head_i is calculated as follows:

$$\begin{aligned}\text{head}_i(p) &= \text{Attention}\left(Q_i, K_i, V_i\right)\\ &= \text{Attention}\left(pW_i^Q, pW_i^k, pW_i^V\right)\end{aligned} \tag{9}$$

where Attention is the attention calculation function; W_i^Q, W_i^K, and W_i^V are weight matrices; and vectors Q_i, K_i, and V_i are calculated from the input vector P and the three weight matrices W_i^Q, W_i^K, and W_i^V. The calculation of Attention is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{10}$$

where softmax is the column-by-column normalization function, Q is the query vector, K is the vector of the correlation between the queried information and other information, V is the vector of the queried information, and d_k is the dimensionality of K.

D. Decoder

The decoder used in this study is a hybrid network of the Transformer-based decoder and CNN-based upsampling (TR-CNNs) and is shown in Fig. 3. The decoder reconstructs the original image information through the unmasked feature sequence with full-text information, including a Transformer-based decoder module and a CNN-based upsampling module. The Transformer-based decoder module recovers the masked feature sequence, whereas the CNN-based upsampling module generates the predicted image.

In the previous section, Z_2 was the encoded sequence, containing only the unmasked patches. Before entering the decoder, the masked patches are filled with zeroes to form a complete sequence. Here, let Z_3 be the complete sequence after filling. The detailed calculation of the overall decoder is as follows:

$$Z_4 = \text{Conv}_{3\times 3}\left(\text{reshape}\left(\text{TR}\left(Z_3\right)\right)\right) \tag{11}$$

$$Z_5 = \text{Conv}_{3\times 3}\left(\text{Conv}_{3\times 3}\left(\text{Upsample}\left(Z_4\right)\right)\right) \tag{12}$$

$$Z_6 = \text{Conv}_{3\times 3}\left(\text{Conv}_{3\times 3}\left(\text{Upsample}\left(Z_5\right)\right)\right) \tag{13}$$

$$Z_7 = \text{Conv}_{3\times 3}\left(\text{Conv}_{3\times 3}\left(\text{Upsample}\left(Z_6\right)\right)\right) \tag{14}$$

$$Z_8 = \text{Conv}_{1\times 1}\left(\text{Conv}_{3\times 3}\left(\text{Conv}_{3\times 3}\left(\text{Upsample}\left(Z_7\right)\right)\right)\right) \tag{15}$$

where Upsample is the upsampling function, Conv represents the convolutional layer, reshape refers to readjusting the number of rows, columns, and dimensions of the matrix, TR represents six-layer stacked Transformer blocks, and the calculation process of the Transformer blocks is presented in Part C. TR reconstructs the masked feature sequence based on the unmasked feature sequence with full-text information, and Z_4 denotes the decoded features. The image features decoded by the Transformer architecture are upsampled to reconstruct the original image through four stages of convolution and upsampling blocks. The feature maps obtained in the four stages are Z_5, Z_6, Z_7, and Z_8, and Z_8 is the predicted image that is the output of the model.

E. Loss Function

In current mainstream generative SSL methods, the masked content is usually consistent with the prediction. For example, the MAE algorithm masks the original color image, and the final prediction of its model is also the pixel value of the masked patch. In our method, the masked content is the high-order image features obtained by the CNN, but the prediction is image pixels. This SSL method of masking high-level features and predicting low-level features was shown to be effective in our experiments. In the back-propagation of the loss calculation, the two modules of the CNN-based feature extraction and Transformer-based encoder are optimized simultaneously.

Our method considers only the masked patches when calculating the loss function. We need to process the mask matrix M by reshaping M from a 1-D sequence of $h/16 \times w/16$ to a 2-D matrix of $[h/16, w/16]$ and then enlarging the size of the 2-D matrix to the original size $[h, w]$ to obtain the mask matrix M_0 corresponding to the original image. The loss function calculation based on the mask matrix M_0 is as follows:

$$\text{Loss} = \frac{\sum_{i=1}^{n} \left| f\left(x_i\right) - x_i \right| \cdot \left(N - M_0\right)}{\text{Sum}\left(N - M_0\right)} \tag{16}$$

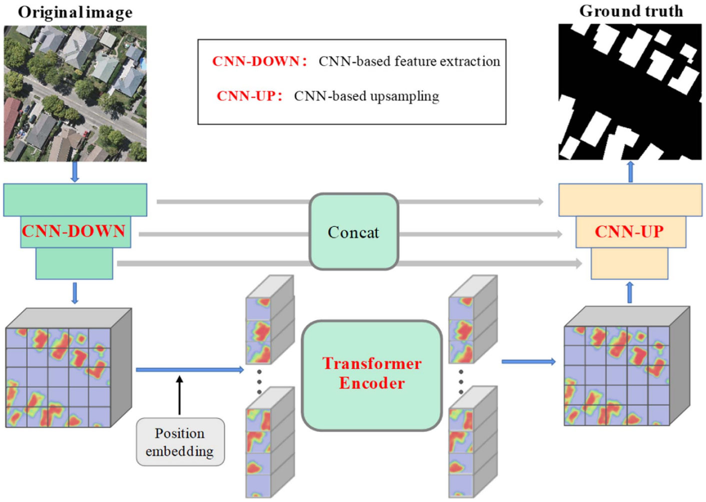


Fig. 4. Network structure of the downstream semantic segmentation task.

where N is a 2-D matrix having the same size as M_0, all values in the matrix are 1, $N - M_0$ is a 2-D matrix with masked regions of 1, $\cdot$ represents matrix dot multiplication, Sum is the sum operation of matrix elements, x_i is the original image of the input, and $f(x_i)$ is the predicted image, which is consistent with the meaning of Z_8 in (15).

F. Model Evaluation

The aim of our method is to improve the representation learning ability of the pretrained MFM model, and we thus directly evaluate the accuracy of the fine-tuned MFM model on the downstream semantic segmentation task. We use the same dataset in the MFM for SSL and the downstream task, i.e., we adopt a "self-pretraining" strategy. In the MFM for SSL, only original images are used for training, and the model obtained from self-supervised training is used as a pretrained model for the downstream task. The performance of the pretrained model is evaluated through few-shot fine-tuning on the downstream task. To verify the effectiveness of the proposed MFM for SSL in this study, we design a downstream task of semantic segmentation. Inspired by the work of He et al. [32], using the two modules of the CNN-based feature extraction and transformer-based encoder, CNN-based upsampling and skip connections are added to obtain a complete semantic segmentation network, and the structure is consistent with the TransUnet network [41]. Fig. 4 shows the network structure, which mainly comprises CNN-based feature extraction, the Transformer-based encoder, and CNN-based upsampling. Among them, the CNN-based feature extraction and Transformer-based encoder use the modules and parameters of the MFM, and these parameters are fixed on the downstream semantic segmentation task. Skip connections between the two modules of CNN-based feature extraction and CNN-based upsampling are added to fuse multiscale features and, thus, reduce the loss of spatial information due to downsampling and improve the pixelwise prediction of the semantic segmentation task.

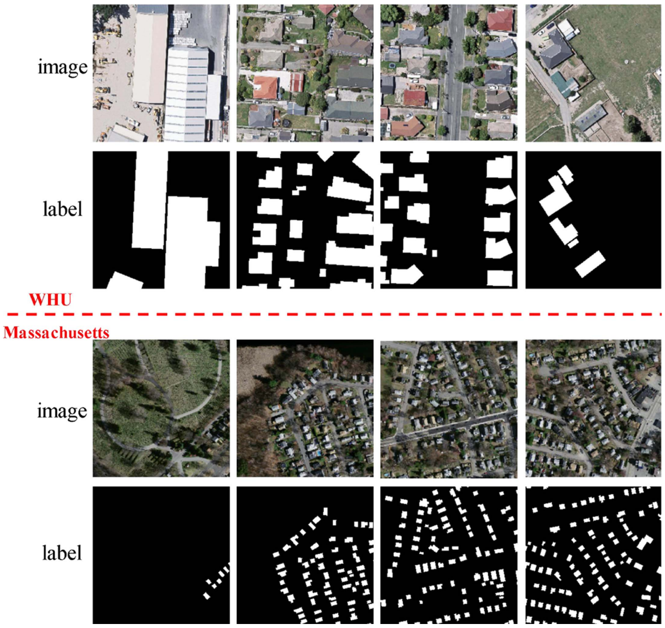


Fig. 5. Examples of the WHU and Massachusetts Building Datasets.

IV. Experiments and Analysis

A. Datasets

In this study, two publicly available datasets, namely the WHU and Massachusetts Building Datasets, are used to validate the proposed method. The datasets contain original images and corresponding building labels and are described as follows.

1) WHU Building Dataset: The original aerial images of the WHU Building Dataset were obtained from the New Zealand Land Information Service website. The spatial resolution of the original images is 0.3 m.The dataset contains 8188 images of 512 × 512 pixels with the corresponding ground truth. To approximate the size of the images in the ViT architecture, we crop each 512 × 512 image into four nonoverlapping images of 256 × 256 pixels. In the downstream task of semantic segmentation, the numbers of training / validation/test data are 18 940/4140/9660, respectively. Examples of this dataset are shown in the upper part of Fig. 5.

2) Massachusetts Building Dataset: The Massachusetts Building Dataset comprises 151 aerial images of the Boston area, each having a size of 1500 × 1500 pixels and an area of 2.25 km^2. Therefore, the entire dataset covers approximately 340 km^2. The original dataset is randomly divided into a training set of 137 images, a test set of 10 images, and a validation set of four images. We further crop the training set, validation set, and test set into images of 256 × 256 pixels. In the downstream task of semantic segmentation, the numbers of training/validation/test data are 4420/230/560, respectively. Examples of this dataset are shown in the lower part of Fig. 5.

B. Training Details

The hardware environment used in the experiment is an Intel CoreI9-10900 K CPU@3.70GHZ, 64 GB memory, and a NVIDIA Tesla V100 32 GB graphics card. The code is written using Pytorch in the Ubuntu environment. The implementation of our method can be divided into two stages, namely, MFM for SSL and the fine-tuning of the MFM on the downstream semantic segmentation task. The training details of the two networks are as follows.

1) MFM for SSL: During training, the CNN-based feature extraction and the Transformer-based encoder are initialized with the parameters of the model pretrained on the ImageNet21k dataset. For all the SSL experiments, the number of training epochs is 800 and the number of batch size is 96. The AdamW optimizer is used, which has β_1=0.9 and β_2=0.999, the initial learning rate is 1×10^{-3}, and the weight decay is $1e^{-8}$. The input image size is scaled to 224×224×3, and the output image also has dimensions of 224×224×3. During data loading, the dataset is augmented through random crop scaling, random horizontal flipping, and color dithering. After training, we save only the network parameters of the CNN-based feature extraction and the Transformer-based encoder modules of the pretrained model and load the parameters of these two modules on the downstream task.

2) Fine-Tuning of the MFM on the Downstream Semantic Segmentation Task: The aim of fine-tuning is to verify the effectiveness of the SSL pretrained model on the downstream task. During training, the parameters of the CNN-based feature extraction and the Transformer-based encoder are obtained from the corresponding weights of the pretrained model, and the parameters of the decoder are initialized randomly. In addition, to verify the representational ability of our SSL method, the parameters of the CNN-based feature extraction and the Transformer-based encoder are fixed during training, and only the parameters of the decoder are updated. For all fine-tuning MFM tasks on the downstream semantic segmentation task, the number of training epochs is 200 and the number of batch size is 196. The Adam optimizer is used, which has β_1=0.9 and β_2=0.999, the initial learning rate is 1×10^{-3}, and the weight decay is $1e^{-8}$. The input image size is scaled to 224×224×3, and the output image is a binarized image of dimensions of 224×224×1.

C. Metrics

Five metrics, namely, intersection over union (IoU), OA, precision, recall, and $F1$, are used to evaluate the results of the downstream semantic segmentation task as follows:

$$\text{IoU} = \text{TP}/\left(\text{TP+FP+FN}\right) \tag{17}$$

$$\text{OA} = \text{TP+TN}/\left(\text{TP++TN+FP+FN}\right) \tag{18}$$

$$\text{Precision} = \text{TP}/\left(\text{TP+FP}\right) \tag{19}$$

$$\text{RecallTP}/\left(\text{TP+FN}\right) \tag{20}$$

$$F1 = 2 \times \left(\text{Precision} \times \text{Recall}\right)/\left(\text{Precision} + \text{Recall}\right) \tag{21}$$

where TP, TN, FP, and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively.

TABLE I
COMPARISON OF SEMANTIC SEGMENTATION RESULTS OF DIFFERENT MASKING STRATEGIES ON THE WHU BUILDING DATASET (UNIT:%)

Mask ratio	IoU	Accuracy	Precision	Recall	F1	Hours
25%	83.21	**97.99**	90.76	**90.91**	90.83	135.6
50%	82.56	97.92	**92.28**	88.68	90.45	114.2
75%	82.25	97.89	90.80	89.73	90.26	**88.9**
Random ratio	**83.59**	97.96	91.57	90.56	**91.06**	116.5

The best results indicated in bold.

TABLE II
COMPARISON OF SEMANTIC SEGMENTATION RESULTS OF THE THREE DIFFERENT DECODERS ON THE WHU BUILDING DATASET (UNIT:%)

Decoder	IoU	Accuracy	Precision	Recall	$F1$
CNNs	82.76	97.93	90.73	90.41	90.57
TR	81.38	97.74	89.62	89.85	89.73
TRCNNs	**83.59**	**97.96**	**91.57**	**90.56**	**91.06**

The best results indicated in bold.

TABLE III
COMPARISON OF THE SEMANTIC SEGMENTATION RESULTS ON THE WHU BUILDING DATASET FOR DIFFERENT SAMPLE AMOUNTS (UNIT:%)

Model	5%	10%	20%	40%	60%	100%
TransUnet	58.55	72.47	78.65	80.86	82.59	**85.66**
ours	**75.39**	**79.05**	**80.20**	**82.32**	**82.85**	83.59

The best results indicated in bold.

TABLE IV
COMPARISON OF SEMANTIC SEGMENTATION RESULTS OF THE DIFFERENT METHODS ON THE WHU BUILDING DATASET AND THE MASSACHUSETTS BUILDING DATASET FOR DIFFERENT SAMPLE AMOUNTS (UNIT:%)

Dataset	Model	5%	10%	20%	40%	60%	100%	hours
WHU	BEiT (baseline) [31]	66.23	72.33	73.84	76.44	77.23	78.02	**50.3**
	SimMIM [35]	71.62	74.68	75.75	76.06	77.28	79.16	132.3
	MAE [32]	71.84	75.53	77.48	78.27	79.30	80.87	59.6
	CAE [37]	73.01	77.16	78.56	79.05	80.81	81.86	73.4
	ours	**75.39**	**79.05**	**80.20**	**82.32**	**82.85**	**83.59**	116.5
Massachusetts	BEiT (baseline) [31]	37.69	40.56	42.66	47.08	47.55	47.93	**8.5**
	SimMIM [35]	35.50	39.17	42.28	42.29	43.33	44.29	23.3
	MAE [32]	37.70	39.81	41.70	45.98	47.20	47.96	8.9
	CAE [37]	40.36	42.59	46.37	48.28	50.43	50.91	11.2
	ours	**42.49**	**45.67**	**47.79**	**50.86**	**51.89**	**53.48**	19.6

The best results indicated in bold.

D. Main Properties

The WHU Building Dataset is used to evaluate the main properties (i.e., the optimal mask ratio, the optimal decoder, and comparisons between MFM pretraining and supervised training with different sample amounts) of our method. In the self-supervised pretraining MFM, we use all the original images (i.e., the images in the training, validation, and test sets) of the WHU Building Dataset as training data. The ground truth is not used, and there is, thus, no information leakage of the downstream task. In the fine-tuning MFM on the downstream task, we load the parameters of the CNN-based feature extraction and Transformer-based encoder of the pretrained MFM to perform semantic segmentation experiments on the WHU Building Dataset. During the training of the downstream semantic segmentation network, we fix the parameters of the CNN-based feature extraction and Transformer-based encoder and only update the parameters of the decoder of the network, so as to evaluate the representation learning ability of the MFM model for different masking strategies and different decoders.

1) Different Mask Ratios: To compare the SSL effects of the different masking strategies, we conduct experiments on building extraction, a downstream semantic segmentation task. We use three fixed mask ratios of 25%, 50%, and 75%, and a random mask ratio ranging from 25% to 80% on the WHU Building Dataset in experiments. The TRCNNs decoder is used. To ensure a fair evaluation, the training parameters of the MFM for SSL and the fine-tuning MFM on the downstream semantic segmentation task for these four masking strategies are consistent with the training details in Part B. Table I compares semantic segmentation results of different mask strategies on the WHU Building Dataset. Table I shows that the pretrained models obtained by the MFM with the fixed mask ratios of 25%, 50%, and 75% have similar performances on the downstream task. However, when the MFM has a higher mask ratio, the

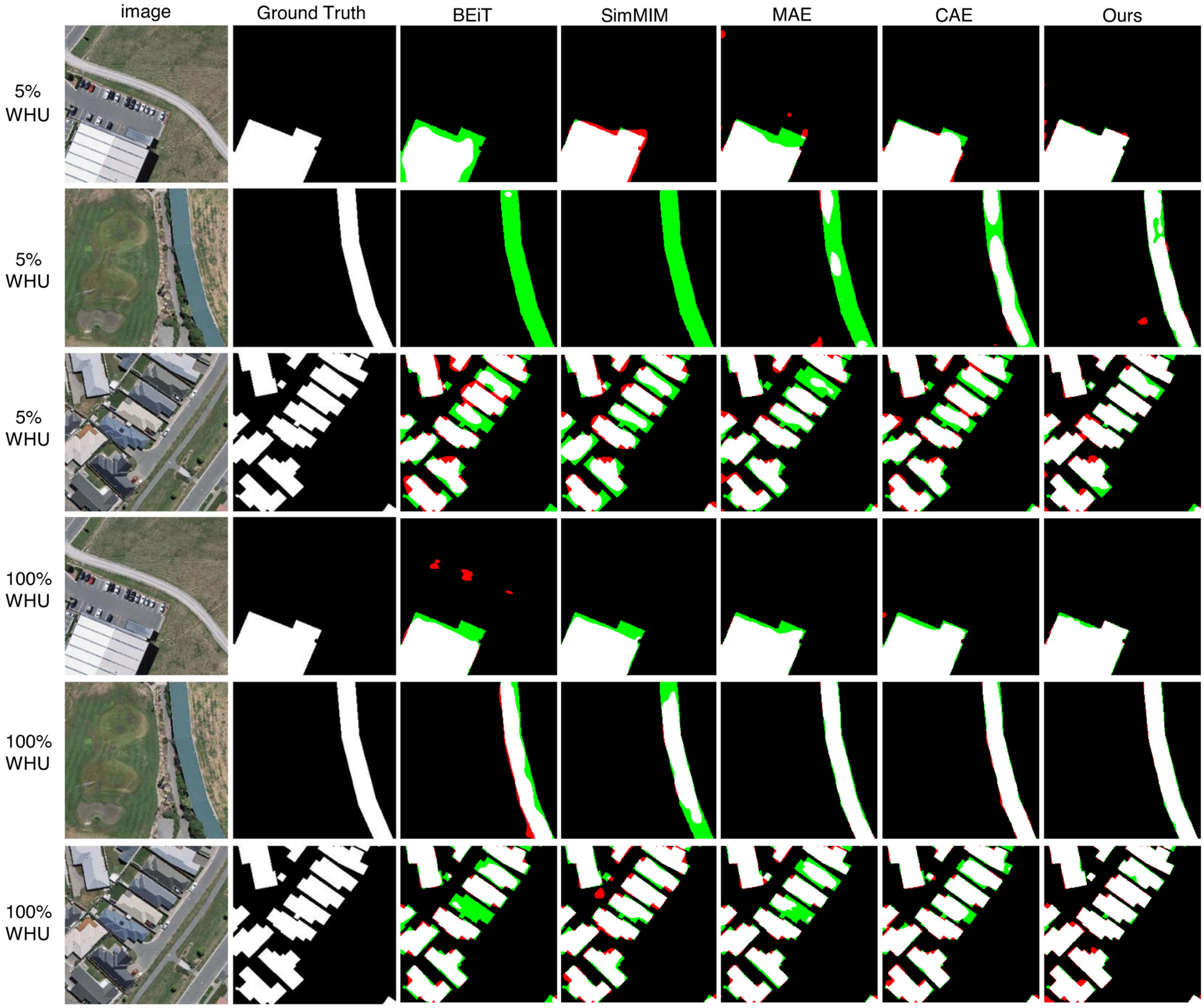


Fig. 6. Comparison of visual semantic segmentation results of different methods on the WHU Building Dataset, with white showing true positives, black true negatives, red false positives, and green false negatives.

transformer-based encoder module used in this study needs to process fewer features, and the MFM with a higher mask ratio is thus less computationally expensive and takes less time. In addition, the MFM with a random proportion mask ratio provides the best results. One possible explanation is that the MFM with a random proportion mask ratio creates more diverse reconstruction tasks for the network, which is beneficial for the network to better understand the image information.

2) Different Decoders: In addition to the TRCNNs decoder introduced in Section III, two other decoders, namely, the CNN-based upsampling and the Transformer-based decoders, are designed for ablation studies. All three decoders are lightweight, with parameters much smaller than those of the encoder. The decoders are described as follows.

1) CNN-based upsampling (CNNs decoder): This decoder aims to generate predictions in a 2-D image space, and we thus need to reshape the complete sequence after filling, from the 2-D shape of $(H \times W)/256 \times C$ to the standard 3-D feature map of $(H/16 \times W/16) \times C$. Next, we gradually reconstruct the original image using four stages of convolution and upsampling blocks, each of which comprises an upsample layer with a scale factor of 2 and two convolutional layers with a kernel size of 3 × 3. In the last convolution layer of this decoder, the number of channels is adjusted to three, i.e., the original image with three channels (e.g., RGB channels) is predicted;
2) Transformer-based decoder (TR decoder): This decoder uses only the Transformer architecture for feature decoding, with a small number of Transformer blocks (six blocks or fewer) applied to the complete sequence after filling;
3) Hybrid decoder (TRCNNs decoder): The TRCNNs decoder combines the CNN-based upsampling and Transformer-based decoders. It first uses a small number of Transformer blocks for feature decoding, followed by lightweight convolutional upsampling layers to reconstruct the original image. Details are presented in Section III.

Semantic segmentation results of the three decoders on the WHU Building Dataset are compared in Table II. Table II shows that all metrics are lowest for the TR decoder, which uses only

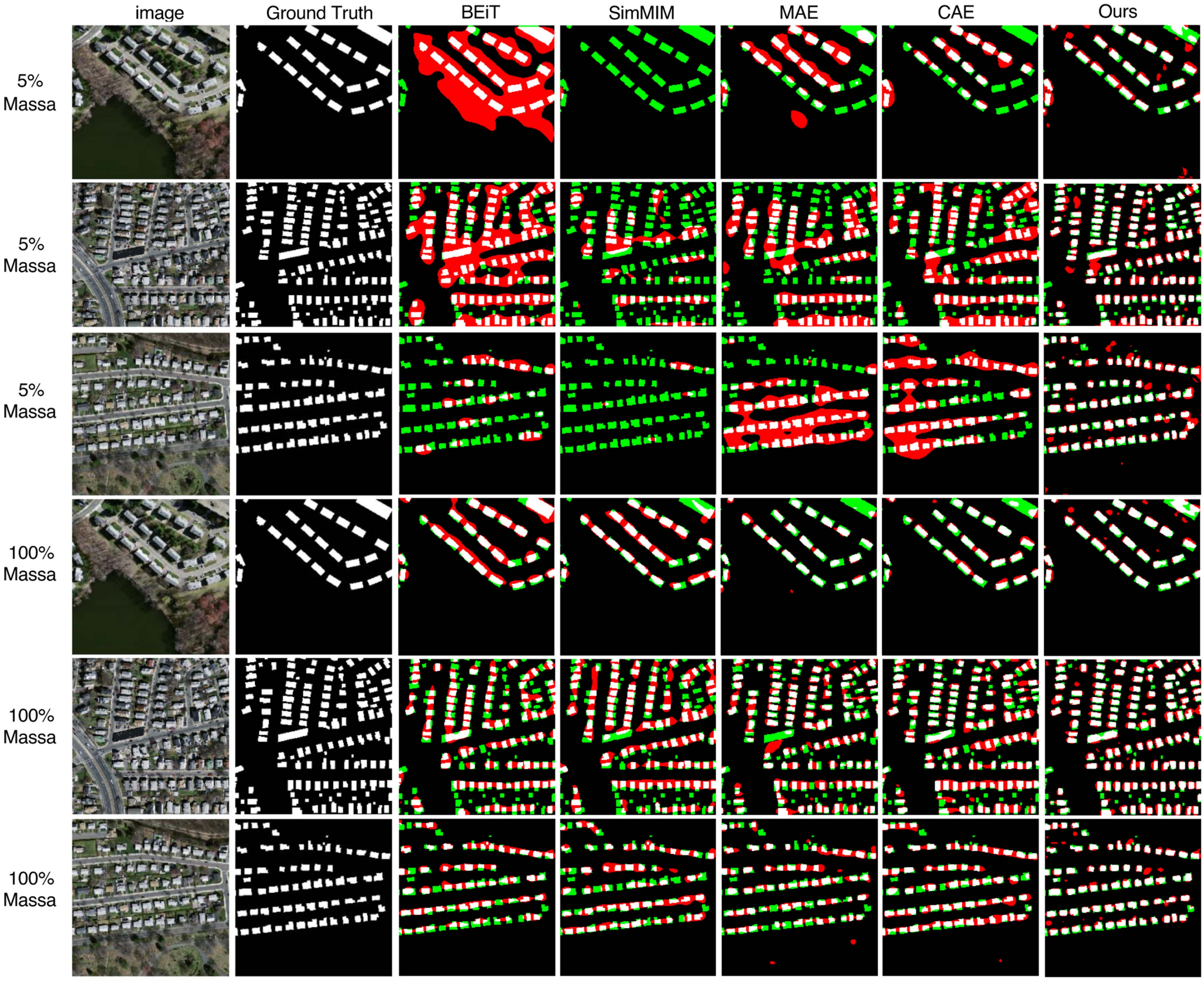


Fig. 7. Comparison of visual semantic segmentation results of the different methods on the Massachusetts Building Dataset, with white indicating true positives, black true negatives, red false positives, and green false negatives.

the Transformer architecture for decoding. The IoU metric of the CNNs decoder, which uses only convolutional upsampling, is 1.38 percentage points higher than that of the TR decoder. This indicates that in our task of predicting original images, the finer local information reconstructed by convolutional upsampling makes a greater contribution to the model's effectiveness. Among the three decoders, the TRCNNs decoder performs the best, having the highest metrics during fine-tuning. The IoU metric for the TRCNNs decoder is 2.21 and 0.83 percentage points higher than the values for the TR and CNNs decoders, respectively. The TRCNNs decoder combines the advantages of the Transformer's excellent global modeling ability and the CNN's effective local feature representation. The experiments show that the TRCNNs decoder outperforms the other two decoders and better optimizes the MFM representation capability during self-supervised pretraining.

3) Comparisons Between MFM Pretraining and Supervised Training With Different Sample Amounts: To verify the effect of high-order features generated in the proposed MFM, we randomly select training samples of building semantic segmentation in different amounts according to the proportions of 5%, 10%, 20%, 40%, 60%, and 100%. This is done to obtain the accuracy of the proposed MFM on the downstream task with different sample amounts. The baseline of the experiment is obtained using the fully supervised TransUnet network. Our results are obtained by loading the pretrained MFM model with the random mask ratio strategy and the TRCNNs decoder. During the model training, the fully supervised TransUnet network loads the pretrained model of R50-ViT-B_16 on ImageNet21k as the initialization parameters. It does not fix the parameters of the encoder and decoder and conducts fully supervised training. As a comparison, we use the TransUnet network again, but this time we load the self-supervised pretrained MFM model as initialization parameters. We fix the parameters of the CNN-based feature extraction and Transformer-based encoder, and only update the parameters of the decoder. Taking IoU as the metric, we compare the semantic segmentation results on the WHU Building Dataset for different sample amounts in Table III. Table III shows that

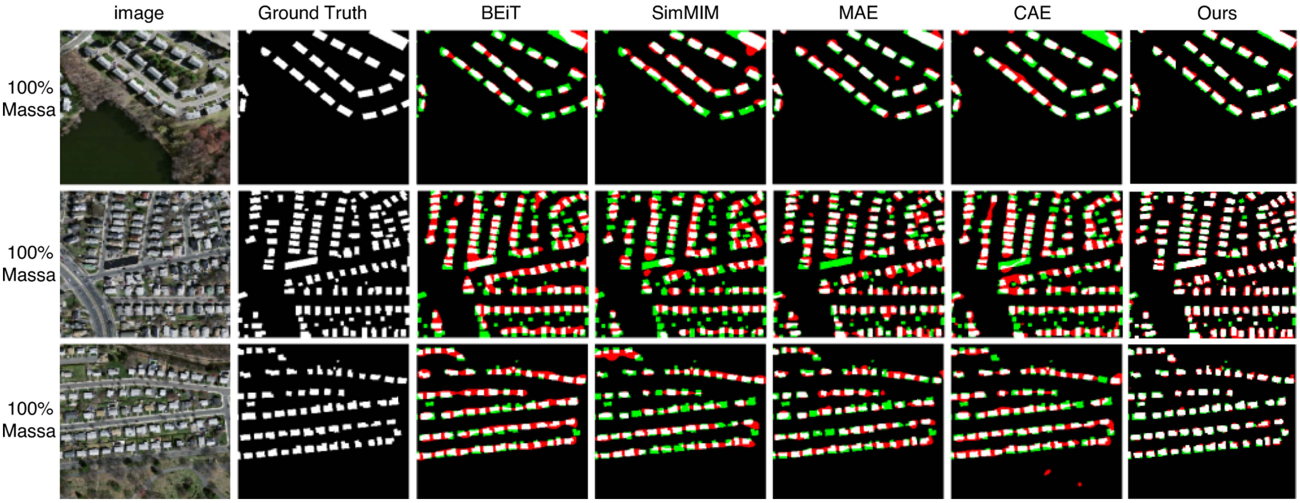


Fig. 8. Comparison of visual semantic segmentation results of the Massachusetts Building Dataset obtained using the WHU pretrained model.

the network loaded with our self-supervised pretrained MFM model achieves higher accuracy than the network loaded with the model pretrained on ImageNet21k when the number of samples is limited. Our method still achieves good accuracy with few samples, e.g., 5% of the sample size. In further experiments, we fix the CNN-based feature extraction and Transformer-based encoder and only update the decoder when training the network loaded with the self-supervised pretrained MFM model. We find that the accuracy of the network improves if the CNN-based feature extraction and Transformer-based encoder are not fixed during training and all of the parameters of the network are optimized as a whole. Detailed results with all parameters updated together are not shown as this is beyond the scope of the article. The performance of our method at a 100% sample size is poorer than that of directly using the TransUnet network for fully supervised training. There are two reasons. First, due to the lack of computing power, we do not use a large-scale dataset for training the MFM. In self-supervised pretraining, the MFM does not have sufficient representation capability. Second, to verify the quality of self-supervised pretraining, we fix the parameters of the CNN-based feature extraction and Transformer-based encoder of the downstream semantic segmentation network, and only train the decoder of the network.

E. Comparisons on the WHU Building Dataset

Comparative experiments are conducted on the WHU Building Dataset to verify the effectiveness of the proposed method against current mainstream SSL algorithms: BEiT [31], SimMIM [35], MAE [32], and CAE [37]. BEiT, the first MIM in computer vision, serves as the baseline and is used by many SSL methods for comparison. To allow a more objective evaluation, the Transformer blocks in the encoders of the SSL methods adopt the settings of ViT-B in the literature [42], and load the parameters of the model pretrained on ImageNet21k. We adopt the best settings for data augmentation and mask ratios as recommended by the authors. For all comparative methods, we use STER-PUP, commonly used in the ViT architecture as the decoder, and the same decoders based on convolutional upsampling on the downstream task for a fair comparison. During downstream training, the encoder parameters remain fixed.

We conduct comparative experiments on the WHU Building Dataset. We first use all the original images of the dataset including the training set, validation set, and test set for self-supervised pretraining, and we then use the training set with different sample amounts for fine-tuning, followed by accuracy statistics. Taking the IoU metric as the statistical accuracy, the semantic segmentation results of the different methods on the WHU Building Dataset for different sample amounts are compared in Table IV. The last column in Table IV is the time cost of pretraining 800 batches of the SSL network in upstream tasks.

Our algorithm demonstrates superior accuracy compared with four other generative SSL algorithms across varying sample sizes of the WHU Building Dataset. For instance, with a mere 5% subset of the dataset, the IoU metric of our algorithm reaches 75.39%, which is 9.16 percentage points higher than BEiT's IoU metric of 66.23%, 3.77 percentage points higher than SimMIM's IoU metric of 71.62%, 3.55 percentage points higher than MAE's IoU metric of 71.84%, and 2.38 percentage points higher than CAE's IoU metric of 73.01%. On the 100% dataset, the IoU metric of our algorithm is 5.57, 4.3, 2.59, and 1.73 percentage points higher than the IoU metrics of the BEiT, SimMIM, MAE, and CAE algorithms, respectively. Our method outperforms the other four methods in fine-tuning with different sample amounts. The proposed MFM for SSL fuses the CNN and Transformer architectures, benefiting from the multiscale features extracted by the CNN architecture and the global modeling ability of the Transformer architecture, and performs better in the remote-sensing image semantic segmentation task. The accuracy of the MAE method in fine-tuning with different sample amounts is higher than that of the SimMIM method, which is also based on the ViT architecture, which indicates that the masking strategy of the MAE method is better than that of the SimMIM method on downstream tasks.

In the MAE method, the encoder only processes unmasked patches, whereas in the SimMIM method, masked patches are replaced by learnable vectors, and the SimMIM's encoder thus processes all patches. The CAE method is better than MAE under different sample amounts. This shows that the CAE strictly separates the representation learning (encoding) from the pretext task is effective. In terms of efficiency, the image in the BEiT method was represented as a sequence of discrete tokens, and the training is faster than that of the other four methods. In our method, more patches are processed during training, and the training time is longer than that of the MAE method due to the random mask ratio used in training, whereas the SimMIM method has the longest training time due to the processing of all patches.

Visual semantic segmentation results of the different methods on the WHU Building Dataset with 5% and 100% sample amounts are compared in Fig. 6. Fig. 6 shows that when fine-tuning the network with 5% of the training dataset for semantic segmentation, the other four methods have many false positives and false negatives, whereas our method is better able to extract buildings with few training data and has obviously fewer false positives and false negatives. Moreover, the edges of buildings extracted using our method are sharper and closer to the ground truth, indicating that our generative SSL method of combining the CNN and Transformer architectures is more effective in local detail prediction. When using the 100% training dataset, the accuracy greatly improves for all five methods, but our method remains the best among the five methods.

F. *Comparisons on the Massachusetts Building Dataset*

To further verify the effectiveness of the proposed method, we conduct comparative experiments using the Massachusetts Building Dataset. We first use all of the dataset's original images for self-supervised pretraining and then use the dataset's training set with different sample amounts for fine-tuning, followed by accuracy statistics. On the Massachusetts Building Dataset, the encoder and decoder for all SSL methods, we adopted the same settings as those of the WHU Building Dataset. The semantic segmentation results of the different methods for the Massachusetts Building Dataset with different sample sizes are compared in Table IV.

Our method outperforms the other four methods on the Massachusetts Building Dataset. For example, on the dataset with 100% of the sample, the IoU metric of our method is 53.48%, which is 5.55 percentage points higher than that of the BEiT method, 9.19 percentage points higher than that of the SimMIM method, 5.52 percentage points higher than that of the MAE method, and 2.57 percentage points higher than that of the CAE method. For 5% of the sample dataset, similar results can be obtained. Our method performs better on datasets with a small sample size, low-definition images, and small targets, such as the Massachusetts Building Dataset. It is worth noting that none of the five methods has the best feature extraction ability of the encoder on the Massachusetts Building Dataset due to the limited images.

Comparisons of the visual semantic segmentation results of different methods on the Massachusetts Building Dataset with 5% and 100% sample amounts are shown in Fig. 7.

Fig. 7. shows that when using 5% of the training dataset for semantic segmentation, due to the low-definition images and the small building targets in the Massachusetts Building Dataset, the BEiT, SimMIM, and MAE methods have many false negatives and false positives. In the case of the BEiT method, large nonbuilding areas are misdetected as buildings. In the case of the SimMIM method, most buildings in the image are missed. The prediction results of our method are better than those of the other methods. Our method has better adaptability than the other methods for the 5% sample dataset with poor image quality. With the use of the complete 100% training dataset, the detection accuracy of all five methods improves significantly. However, the SimMIM method still suffers from a severe issue of false negatives (e.g., small buildings being incorrectly predicted as background) and false positives (e.g., spaces between buildings being predicted as building areas).

Due to the limited number of images in the Massachusetts Building Dataset, which can adversely affect the quality of representation learning, we adopted a consistent approach for all five algorithms (BEiT, SimMIM, MAE, CAE, and our proposed method). To mitigate the impact of the dataset size, we employed self-supervised pretraining models from Section E that underwent 800 epochs of training on the WHU building dataset. Subsequently, these models were fine-tuned for 200 epochs using the entire training set of the Massachusetts Building Dataset. During the fine-tuning process for downstream tasks, we loaded the pretrained encoder parameters and utilized the same decoder (STER-PUP) for all pretraining models. The semantic segmentation results of the Massachusetts Building Dataset obtained using the WHU-pretrained model are compared in Table V.

Table V shows that even if the self-supervised pretraining model derived using the WHU Building Dataset is used, and only the sample data of the Massachusetts Building Dataset are used to fine-tune the obtained self-supervised pretrained model, the IoU metrics are higher than those when using only the Massachusetts Building Dataset. Among the results, the IoU metric of the BEiT method reaches 49.03%, which is 1.10 percentage points higher than that of the self-supervised pretraining and fine-tuning using only the Massachusetts Building Dataset. Similarly, the IoU metric of SimMIM, MAE, and CAE are 2.83, 3.36, and 1.53 percentage points higher, respectively. Our method is 2.98 percentage points higher, and all the SSL methods are of considerable improvements.

Fig. 8 visually compares the fine-tuning semantic segmentation results of the Massachusetts Building Dataset obtained using the WHU pretrained model. The following conclusions hold for all five SSL methods. The fine-tuning results of the model pretrained using the WHU Building Dataset are better than those of the model pretrained using the Massachusetts Building Dataset (shown in Fig. 7). Meanwhile, due to the limitation of computing power and dataset size, none of the five SSL methods achieve the best results. The performances of all five generative SSL methods depend on the number of

TABLE V
COMPARISON OF SEMANTIC SEGMENTATION RESULTS OF THE MASSACHUSETTS BUILDING DATASET OBTAINED USING THE WHU-PRETRAINED MODEL (UNIT:%)

Model	IoU	Accuracy	Precision	Recall	$F1$
BEiT (baseline) [31]	49.03	86.70	63.01	68.84	65.80
SimMIM [35]	47.12	86.32	62.77	65.41	64.06
MAE [32]	51.32	87.87	67.07	68.60	67.83
CAE [37]	52.44	88.11	67.58	70.07	68.80
ours	**56.46**	**89.89**	**74.13**	**70.32**	**72.18**

The best results indicated in bold.

original images. The accuracy will further improve if training is performed on a larger remote-sensing original image dataset.

Furthermore, we have conducted comparative experiments of SSL algorithms on a land cover dataset, specifically the Gaofen Image Dataset (GID). The efficacy of our method has been validated through multiclass semantic segmentation experiments. Detailed experimental procedures and results can be found in the Appendix.

V. CONCLUSION

This article introduces a generative SSL framework for high-resolution remote sensing images, known as MFM, which integrates CNN and Transformer architectures. The proposed method leverages MFM to obtain high-level image representations. The resulting pretrained model significantly enhances the accuracy of downstream tasks, such as the semantic segmentation of high-resolution remote sensing images, while reducing reliance on annotated samples. From an algorithmic perspective, we have designed an MFM network that merges CNN and Transformer architectures, preserving the superior local feature representation and convergence performance of CNN, and incorporating the comprehensive modeling capability of the Transformer architecture. This hybrid architecture is particularly beneficial for remote-sensing image datasets with limited samples. In terms of experimentation, we have assessed the primary attributes of our method using the WHU Building Dataset, investigated the impact of masking strategy and decoder selection on the network, and confirmed the effectiveness of the pretrained model on the downstream semantic segmentation task. Furthermore, we have compared our method with four popular generative SSL methods using the WHU and Massachusetts Building Datasets. Performance comparisons on the downstream task of semantic segmentation have demonstrated the superiority of our proposed method for high-resolution remote-sensing image datasets. Future research will incorporate algorithms related to contrastive learning and large-scale data processing to further explore more efficient and universal SSL methods for remote sensing images, thereby addressing the challenges of strong dependence on high-quality large samples and domain adaptation in high-resolution remote sensing imagery.

APPENDIX A

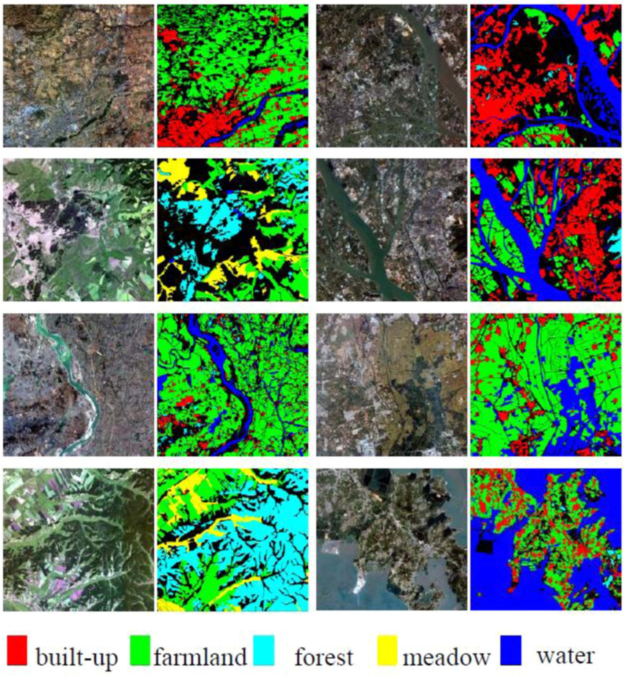


Fig. 9. Examples of the GID-5 Datasets.

In this Appendix, we extend the validation of our method to the Gaofen Image Dataset (GID). The GID is a substantial dataset designed for land use and land cover (LULC) classification. It comprises 150 high-quality Gaofen II (GF-2) images sourced from over 60 different cities across China. The GID dataset includes two sets of labels: A 5-category semantic segmentation label and a 15-category semantic segmentation label. In this article, we utilize the 5-category (i.e., GID-5) label. We partition the large images into 109 200 smaller images of 256 × 256 pixels, and distribute them into a training set (65 520 images), a validation set (21 840 images), and a test set (21 840 images) in a 6:2:2 ratio. Examples of the GID-5 Dataset are illustrated in Fig. 9. During the training process, we adhere to the paradigm outlined in the article: We first perform self-supervised pretraining using all raw images for 400 epochs, followed by fine-tuning on the downstream task for 200 epochs. The batch size is set to 96 for self-supervised pretraining and 196 for fine-tuning on the downstream task. In the fine-tuning phase on the downstream task, we load the parameters of the encoder of the pretrained SSL to execute LULC classification on the GID-5 Dataset. To verify the feature representation ability of the self-supervised pretrained model, the loaded parameters of the encoder are kept constant during fine-tuning. In the GID-5 Dataset, we employ intersection over union (IoU) as the accuracy metric for semantic segmentation for each category, and mean

TABLE VI
Comparison of Semantic Segmentation Results of the Different Methods on the GID-5 Dataset (unit:%)

Model	IoU of Built-up	IoU of Farmland	IoU of Forest	IoU of Meadow	IoU of Water	*m*IoU
BEiT (baseline) [31]	84.27	68.61	66.80	59.91	66.22	69.16
SimMIM [35]	84.99	70.56	69.01	61.69	68.97	71.03
MAE [32]	85.72	73.03	**69.10**	61.38	68.44	71.53
CAE [37]	86.31	73.06	66.67	62.67	**69.35**	71.61
ours	**86.42**	**73.13**	69.09	**64.13**	68.88	**72.33**

The best results indicated in bold.

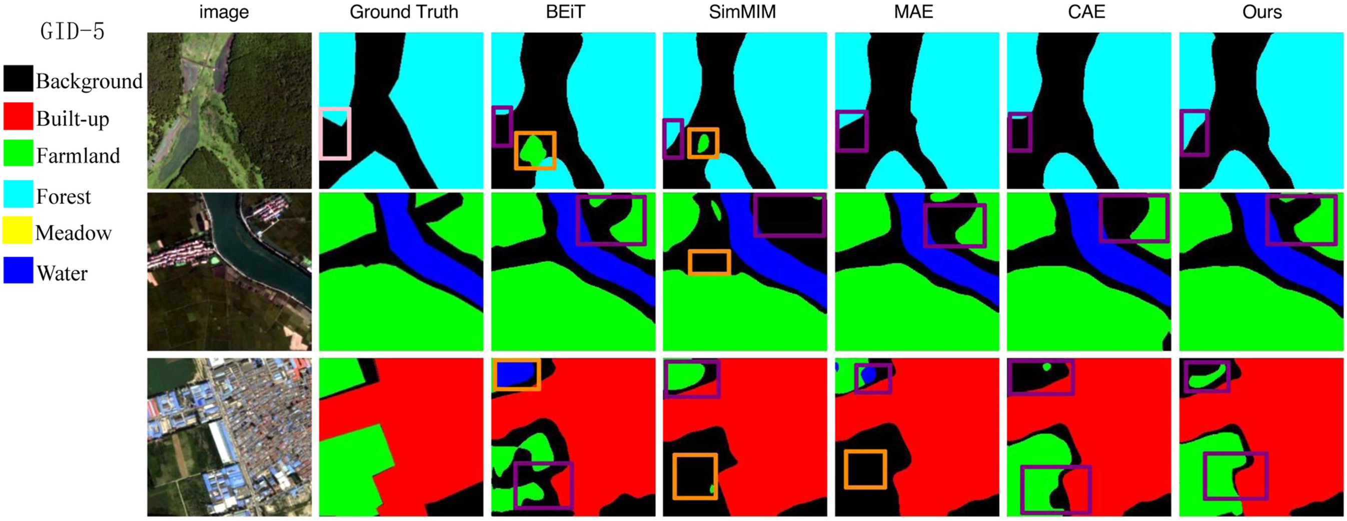


Fig. 10. Comparison of visual semantic segmentation results of the GID-5 Dataset. Mislabeled areas in the ground truth are highlighted with pink boxes. Incorrect segmentation results are highlighted with orange boxes. Competitive segmentation results are highlighted with purple boxes.

intersection over union (mIoU) is used as the metric for semantic segmentation accuracy across all categories. The calculation of mIoU is as follows:

$$\text{mIoU} = \frac{1}{k+1}\sum_{i=0}^{k}\frac{\text{TP}}{\text{FN}+\text{FP}+\text{TP}} \tag{22}$$

where TP, TN, FP, and FN represent the numbers of true positives, true negatives, false positives, and false negatives, respectively. The variable k denotes the number of all categories excluding the background, and $k+1$ represents the total number of categories, including the background. The semantic segmentation results of various methods on the GID-5 Dataset are compared in Table VI. Due to the large sample size of the GID-5 dataset, the gap in mIoU between different methods is smaller than that observed in other semantic segmentation datasets (i.e., WHU Building Dataset and Massachusetts Building Dataset) when fine-tuning on the downstream task. Our method outperforms other SSL methods, achieving the highest mIoU. For instance, it is 0.72% higher than that of CAE, which has the second-highest mIoU, and 3.17% higher than that of BEiT, which has the lowest mIoU. Visual comparisons of different methods on the GID-5 Dataset are presented in Fig. 10. As depicted in Fig. 10, our method yields superior results, whereas other methods tend to produce false positives and false negatives (e.g., competitive segmentation results in the purple boxes).

References

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

[2] J. Kang, R. F.-Beltran, P. Duan, S. Liu, and A. J. Plaza, "Deep unsupervised embedding for remotely sensed images based on spatially augmented momentum contrast," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2598–2610, Mar. 2021.

[3] H. Jung, Y. Oh, S. Jeong, C. Lee, and T. Jeon, "Contrastive self-supervised learning with smoothed representation for remote sensing," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 8010105.

[4] X. Li, D. Shi, X. Diao, and H. Xu, "SCL-MLNet: Boosting few-shot remote sensing scene classification via self-supervised contrastive learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5801112.

[5] J. Wang, X. Wang, L. Xing, B.-D. Liu, and Z. Li, "Class-shared SparsePCA for few-shot remote sensing scene classification," *Remote Sens.*, vol. 14, no. 10, 2022, Art. no. 2304.

[6] X. Xiao, C. Li, and Y. Lei, "A lightweight self-supervised representation learning algorithm for scene classification in spaceborne SAR and optical images," *Remote Sens.*, vol. 14, no. 13, 2022, Art. no. 2956.

[7] H. Zhou, X. Du, and S. Li, "Self-supervision and self-distillation with multilayer feature contrast for supervision collapse in few-shot remote sensing scene classification," *Remote Sens.*, vol. 14, no. 13, 2022, Art. no. 3111.

[8] V. Stojnic and V. Risojevic, "Self-supervised learning of remote sensing scene representations using contrastive multiview coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1182–1191.

[9] C. Tao, J. Qi, W. Lu, H. Wang, and H. Li, "Remote sensing image scene classification with self-supervised paradigm under limited labeled samples," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2020, Art. no. 8004005.

[10] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.

[11] M. Zhu, J. Fan, Q. Yang, and T. Chen, "SC-EADNet: A self-supervised contrastive efficient asymmetric dilated network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5519517.

[12] S. Hou, H. Shi, X. Cao, X. Zhang, and L. Jiao, "Hyperspectral imagery classification based on contrastive learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5521213.

[13] H. Xu, W. He, L. Zhang, and H. Zhang, "Unsupervised spectral–spatial semantic feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526714.

[14] L. Zhao, W. Luo, Q. Liao, S. Chen, and J. Wu, "Hyperspectral image classification with contrastive self-supervised learning under limited labeled samples," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 6008205.

[15] L. Song, Z. Feng, S. Yang, X. Zhang, and L. Jiao, "Self-supervised assisted semi-supervised residual network for hyperspectral image classification," *Remote Sens.*, vol. 14, no. 13, 2022, Art. no. p. 2997.

[16] P. Guan and E. Y. Lam, "Cross-domain contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5528913.

[17] Y. Chen and L. Bruzzone, "Self-supervised SAR-optical data fusion of sentinel-1/-2 images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5406011.

[18] L. Zhang, S. Zhang, B. Zou, and H. Dong, "Unsupervised deep representation learning and few-shot classification of PolSAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2020, Art. no. 5100316.

[19] H. Li et al., "Global and local contrastive self-supervised learning for semantic segmentation of HR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5618014.

[20] H. Gao, Y. Zhao, P. Guo, Z. Sun, X. Chen, and Y. Tang, "Cycle and self-supervised consistency training for adapting semantic segmentation of aerial images," *Remote Sens.*, vol. 14, no. 7, 2022, Art. no. 1527.

[21] V. Marsocci, S. Scardapane, and N. Komodakis, "MARE: Self-supervised multi-attention REsu-net for semantic segmentation in remote sensing," *Remote Sens.*, vol. 13, no. 16, 2021, Art. no. 3275.

[22] K. Cha, J. Seo, and Y. Choi, "Contrastive multiview coding with electro-optics for SAR semantic segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2021, Art. no. 4018505.

[23] S. Saha, P. Ebel, and X. X. Zhu, "Self-supervised multisensor change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4405710.

[24] A. Ciocarlan and A. Stoian, "Ship detection in sentinel 2 multi-spectral images with self-supervised learning," *Remote Sens.*, vol. 13, no. 21, 2021, Art. no. 4255.

[25] Y. Xu, H. Sun, J. Chen, L. Lei, K. Ji, and G. Kuang, "Adversarial self-supervised learning for robust SAR target recognition," *Remote Sens.*, vol. 13, no. 20, 2021, Art. no. 4158.

[26] X. Zheng, B. Kellenberger, R. Gong, I. Hajnsek, and D. Tuia, "Self-supervised pretraining and controlled augmentation improve rare wildlife recognition in UAV images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 732–741.

[27] O. Manas, A. Lacoste, X. G. Nieto, D. Vazquez, and P. Rodriguez, "Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9414–9423.

[28] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," in *Proc. 3rd Int. Conf. Learn. Representations*, Y. Bengio, Y. LeCun, Eds., San Diego, CA, USA, May 7–9, 2015. [Online]. Available: http://arxiv.org/abs/1410.8516.

[29] A. V. D. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2016, pp. 1747–1756.

[30] A. V. d. Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4797–4805.

[31] H. Bao, L. Dong, S. Piao, and F. Wei, "BEiT: BERT pre-training of image transformers," in *Proc. 10th Int. Conf. Learn. Representations*, Apr. 25–29, 2022. [Online]. Available: https://openreview.net/forum?id=p-BhZSz59o4

[32] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16000–16009.

[33] Y. Cong et al., "SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 197–211.

[34] Z. Xue, X. Yu, A. Yu, B. Liu, P. Zhang, and S. Wu, "Self-supervised feature learning for multimodal remote sensing image land cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5533815.

[35] Z. Xie et al., "SimMIM: A simple framework for masked image modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9653–9663.

[36] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14668–14678.

[37] X. Chen et al., "Context autoencoder for self-supervised representation learning," in *Proc. Int. J. Comput. Vis.*, vol. 132, no. 1, 2024, pp. 208–223.

[38] X. Sun et al., "RingMo: A remote sensing foundation model with masked image modeling," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2022, Art. no. 5612822.

[39] Y. Liu, S. Zhang, J. Chen, Z. Yu, K. Chen, and D. Lin, "Improving pixel-based MIM by reducing wasted modeling capability," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 5361–5372.

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

[41] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*. [Online]. Available: https://api.semanticscholar.org/CorpusID:231847326

[42] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Representations*, May 3–7, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

Shiyan Pang received the B.S. degree in science and technology of surveying and mapping, the M.S. and Ph.D. degrees in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2009, 2012, and 2015, respectively.

From 2016 to 2018, she was a Postdoctor with the Collaborative Innovation Center for Geospatial Technology and the School of Resource and Environmental Sciences, Wuhan University, Wuhan, China. She is currently an Associate Professor with the Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, China. Her research interests include machine learning, deep learning, and semantic segmentation and change detection of remotely sensed data.

Hanchun Hu received the B.S. degree in process equipment and control engineering from the Wuhan Institute of Technology, Wuhan, China, in 2021. He is currently working toward the M.S. degree in software engineering with the Central China Normal University, Wuhan.

His research interests include deep learning and remote sensing image processing.

Zhiqi Zuo received the B.S. degree in remote sensing science and technology and the M.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2009 and 2013, respectively.

He is an Experimentalist with the College of Informatics, Huazhong Agricultural University, Wuhan, China. His research interests include machine learning, deep learning, and experimental data processing.

Jia Chen received the master degree in electrical engineering and the Ph.D. degree in control science and engineering, with a specialization in robot vision, from the Harbin Institute of Technology, Harbin, China, in 2007 and 2012, respectively.

He was a lead Research and Development Engineer with Samsung Research Institute (Beijing) on XXXX-Avatar. After that, he was with Central China Normal University. From 2017 to 2018, he was a Visiting Scholar with the Centre for Vision, Speech, and Signal Processing, the University of Surrey, Guildford, U.K. He is currently an Associate Professor with Digital Media Technology and is with the Faculty of Artificial Intelligence in Education, CCNU, Wuhan, China. His current research interests include VR/AR, 3-D reconstruction, 3-D motion capture, and educational information technology.

Xiangyun Hu received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2001.

From 2002 to 2005, he was a Postdoctoral Research Fellow with the Department of Earth and Space Science and Engineering, Lassonde School of Engineering, York University, Canada. He has developed a feature extraction technology SmartDigitizer acquired by PCI Geomatics, Leica Geosystems, and Microsoft. From 2005 to 2010, he was a Senior Software Engineer with ERDAS, Inc., Atlanta, USA. He is currently a Professor and Head of the Department of Photogrammetry with the School of Remote Sensing and Information Engineering, Wuhan University. He is also an adjunct Professor with Hubei Luojia Laboratory, Wuhan, China. He has authored or coauthored more than 60 papers in journals and conferences in intelligent feature extraction of remotely sensed data. Recently he has been leading a team developing an open-source deep learning framework—LuojiaNET.