UNetFormer: A UNet-like Transformer for Efficient Semantic Segmentation of Remote Sensing Urban Scene Imagery

Abstract

Semantic segmentation of remote sensing images in urban scenes is of great significance, which can effectively support urban planning, environmental monitoring, and disaster emergency response.UNetFormer proposes a semantic segmentation method for remote sensing images based on an improved Transformer model, aiming to realize the accurate segmentation of complex urban scenes by combining the ResNet18 encoder and the efficient global-local attention module. accurate segmentation of complex urban scenes by combining ResNet18 encoder and efficient global-local attention module. In order to further improve the performance of the model, this paper optimizes the SAM (Self-Attention Mechanism) raw output-assisted model and conducts experiments on the Potsdam dataset. The experimental results show that the optimized model achieves significant performance improvement in the segmentation task in multiple categories, especially in the categories of buildings and trees, with larger average F1 scores and intersection ratios (mIoU) are both improved. However, there is a slight performance degradation in some categories such as Car, while the lightweighting direction still needs to be improved. Future research will further focus on optimizing the efficiency of the model and balancing the performance of different modules.

Keywords: Remote sensing images, Semantic segmentation, Global-local attention, SAM optimization.

1 Introduction

Semantic segmentation of remotely sensed images refers to the categorization of each pixel in a remotely sensed image into a specific semantic category. Due to the increasing global capture of high resolution remotely sensed urban scene images with rich spatial details and rich potential semantic content, urban scene images have been widely subjected to semantic segmentation, resulting in a variety of urban related applications including land cover mapping, change detection, road and building extraction and many other practical applications. Therefore semantic segmentation of remote sensing images is of great significance to both human beings and the state. For human beings, it plays a key role in urban planning, environmental monitoring, and disaster emergency response, and can improve the efficiency of resource management and emergency response. For the state, it is also crucial in homeland security and natural resource management, supporting the state's monitoring of critical areas and rational utilization of natural resources. Therefore, it is of great relevance to reproduce this thesis [1].

2 Related works

Under the wave of deep learning, remote sensing image semantic segmentation also has several stages of development.

2.1 CNN

In the beginning, Convolutional Neural Networks (CNNs) dominated the semantic segmentation task, and in the semantic segmentation task of remote sensing images, CNN-based approaches are able to capture finer-grained local contextual information, which underpins their great capabilities in feature representation and pattern recognition. Through multiple stacking of convolutional layers, CNNs are able to gradually learn from simple local features to more complex image structures for effective pixel-level classification. This capability is particularly applicable to complex feature classes and fine spatial structures in remote sensing images. Compared with traditional methods, CNN can automatically learn the most discriminative features from the data without relying on manual feature design, which greatly improves the accuracy and robustness of semantic segmentation.

2.2 Self-attention mechanism

Due to the fixed acceptance view of convolutional operations and its lack of ability to model global contextual information or remote dependencies, traditional convolutional neural networks (CNNs) have limited performance in dealing with some tasks that require long-distance contextual relationships. To address this problem, researchers have introduced the self-attention mechanism, which is capable of capturing the dependencies between long-distance pixels in an image, thus enhancing the global context modeling capability. However, the self-attention mechanism usually requires a large amount of computational resources and memory consumption, especially when processing high-resolution images, where the computational and memory requirements increase substantially. This makes the use of self-attention mechanisms in real-time urban applications, especially for resource-limited devices, more challenging and restrictive.

3 Method

3.1 Overview

Since the use of the self-attention mechanism leads to a much higher model complexity, especially when dealing with high-resolution images, and the computational resources and memory consumption increase, many real-time urban applications are not well adapted to this computationally intensive model. To address this problem, this paper proposes several approaches to achieve lightweighting of the model, aiming to reduce computational and memory consumption while maintaining high segmentation accuracy. In order to achieve accurate urban scene segmentation while ensuring network efficiency, the overall framework of the model is a Unet-like Transformer (UNetFormer), as shown in Fig. 1, where the encoder part is ResNet18 and the decoder consists of an efficient global-local based on the Transformer

improved attention mechanism module based on Transformer, which also contains a specially setup feature refinement header.

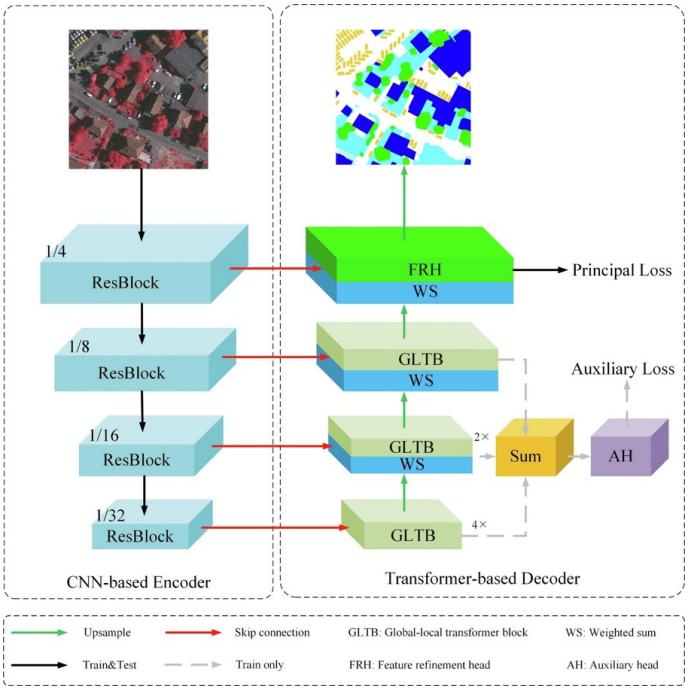

Figure 1. Model Schematic

3.2 ResNet18

The encoder part uses the ResNet18 network, which is a CNN-based encoder structure capable of efficiently extracting low- and mid-level features of an image.The ResNet18 network consists of four stages of ResBlock, each of which downsamples the feature map through a convolutional layer with a downsampling ratio of 2, the spatial resolution of the feature map is halved at each stage, while the number of channels is gradually increased. while the number of channels is gradually increased. In order to maintain the expressiveness of the features, the feature maps generated at each stage are adjusted to 64 channels by a 1x1 convolution operation to better fuse the information from different stages. These adjusted feature maps are fused with the feature maps of the corresponding stages of the decoder to form jump connections. The jump connections can effectively preserve the low-level feature information and help the decoder to recover the details of the image more finely, thus improving the segmentation precision and accuracy. Through this design, the encoder is not only able to extract rich multi-scale features, but also realize efficient information transfer and fusion through jump connections.

3.3 Global-Local Attention Module

In this paper, in order to reduce the computational complexity of Transformer and achieve the effect of lightweight, the modification of the self-attention module, in which a global-local attention is proposed to be used to replace the multi-head self-attention, as shown in Fig. 2.

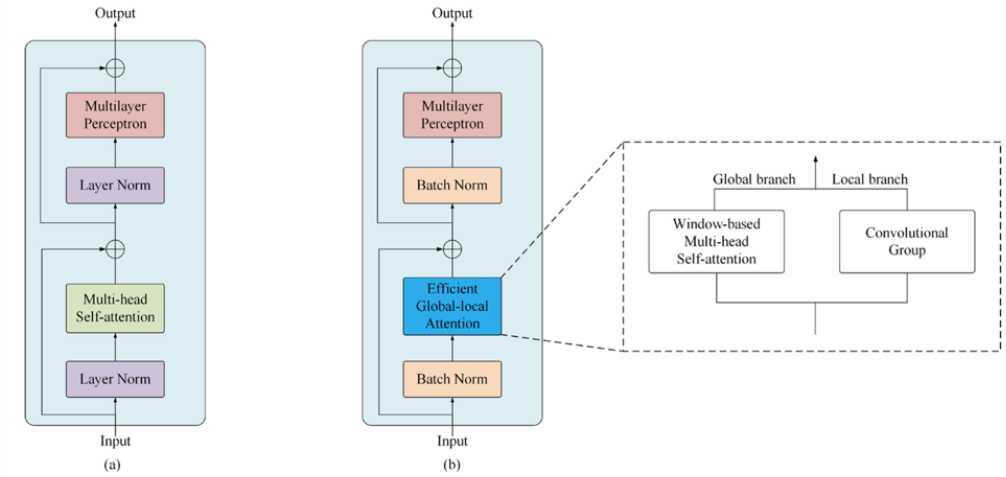

Figure 2. Improved Global Local Attention Module Based on Transformer

This global-local attention mechanism consists of two parts (Fig. 3(a)), one of which is the local branch, which focuses on the local information of the feature map and captures fine-grained local features in the image, and the other part is the global branch, which is efficiently reduced in computational by the window segmentation method (Fig. 3(b)). complexity. In global branching, the image is partitioned into multiple windows, and the attention is computed independently within each window, thus avoiding the high overhead of global computation. In order to further improve the efficiency of cross-window information transfer, the method of cross context interaction (Fig. 3(c)) is proposed, which enables the efficient exchange of information between different windows and thus enhances the model's ability to model the global context.

Furthermore, in order to take advantage of the rich spatial details generated by the first ResBlock, the corresponding global-local attention module is replaced with a feature refinement header. This feature refinement header (Fig 4) further enhances the feature representation by constructing two paths: one path mainly deals with the channel information to enhance the channel representation capability of the feature, while the other path focuses on the spatial information to improve the spatial detail representation capability. This dual-path design can better fuse local details and global semantic information to further refine the features and improve the accuracy and robustness in image segmentation tasks.

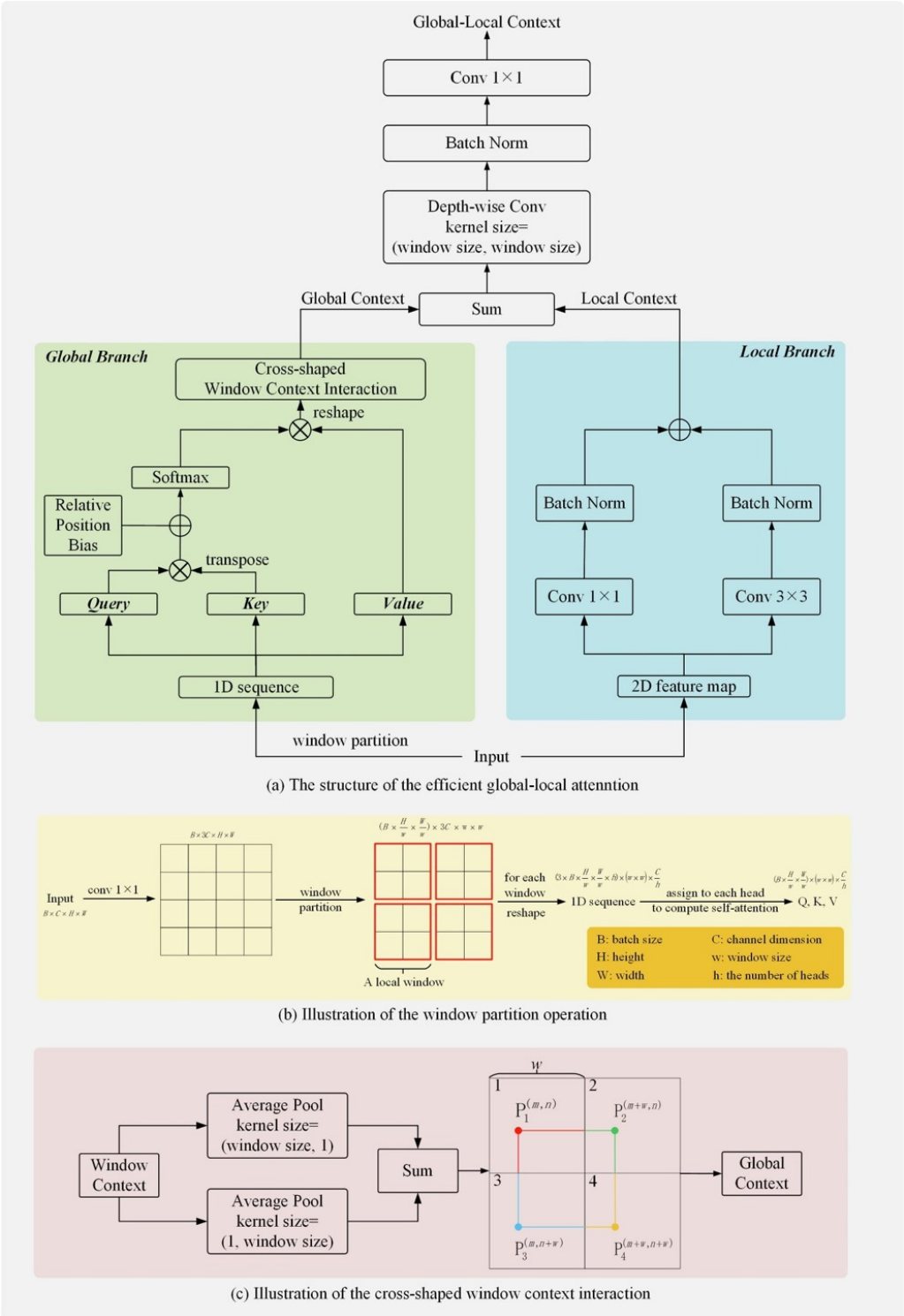

Figure 3. Global Local Attention Module

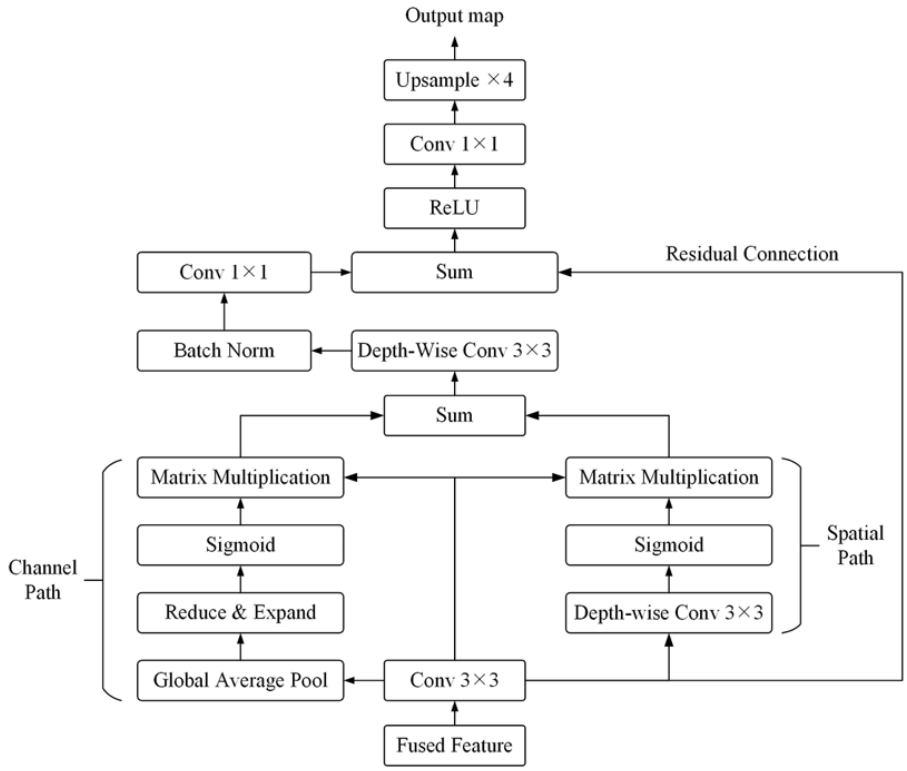

Figure 4. Feature Refinement Header

3.4 Loss

The main loss consists of two components, the cross-entropy loss and the Dice loss, which are obtained by using the final output of the model with the true value. The combination of these two loss functions can simultaneously optimize the model's performance in terms of classification accuracy and segmentation quality. The cross-entropy loss is mainly used to measure the accuracy of pixel-level classification for multi-class segmentation tasks, while the Dice loss focuses on evaluating the overlap between the predicted results and the true labels, and is especially good in dealing with the class imbalance problem because it emphasizes the similarity of the segmented regions.

In addition, to further improve the performance of the model, this paper proposes an auxiliary cross-entropy loss function that utilizes the outputs of the first three layers. Specifically, the model performs up-sampling and summing operations on the outputs of these three layers so that they have the same spatial resolution as the final output. Then, these processed feature maps are compared with the real labels to compute the auxiliary cross-entropy loss. The introduction of this auxiliary loss helps the model to utilize the intermediate layer information earlier in the training process, thus accelerating convergence and improving the model's ability to capture details. In this way, the model is not only able to optimize globally, but also gradually improve the segmentation accuracy in local regions.

4 Implementation details

4.1 Main contributions

Based on this paper, the raw output of SAM is combined with the raw output of SAM using the method proposed in SAM-Assisted Remote Sensing Imagery Semantic Segmentation With Object and Boundary Constraints [2]. UNetFormer for further optimization. By introducing the auxiliary information generated by SAM, UNetFormer is able to better capture object boundaries and structural details in the segmentation task, thus improving the performance of the model. The experiments were conducted using the Potsdam dataset, a dataset containing high-resolution remotely sensed images suitable for evaluating semantic segmentation performance in urban environments. Combined with the SAM approach, the segmentation accuracy can be further improved while ensuring computational efficiency, especially in complex feature boundary recognition and small object detection.

4.2 Comparing with the released source codes

The following code is the open source code given for SAM-Assisted Remote Sensing Imagery Semantic Segmentation With Object and Boundary Constraints, with the data processing section modified to fit the Potsdam dataset used in this paper.

```
1
2  def train(net, optimizer, epochs, scheduler=None, weights=WEIGHTS,
       save_epoch=1):
3      losses = np.zeros(1000000)
4      mean_losses = np.zeros(100000000)
5      weights = weights.cuda()
6
7      iter_ = 0
8      MIoU_best = 0.30
9      criterionb = BoundaryLoss()
10     criteriono = ObjectLoss()
11     for e in range(1, epochs + 1):
12         if scheduler is not None:
13             scheduler.step()
14         net.train()
15         for batch_idx, (data, boundary, object, target) in enumerate(
               train_loader):
16             data, target = Variable(data.cuda()), Variable(target.cuda())
17             optimizer.zero_grad()
18             output = net(data)
19             loss_ce = loss_calc(output, target, weights)
20             loss_boundary = criterionb(output, boundary)
```

```
            loss_object = criteriono(output, object)

            if LOSS == 'SEG':
                loss = loss_ce
            elif LOSS == 'SEG+BDY':
                loss = loss_ce + loss_boundary * LBABDA_BDY
            elif LOSS == 'SEG+OBJ':
                loss = loss_ce + loss_object * LBABDA_OBJ
            elif LOSS == 'SEG+BDY+OBJ':
                loss = loss_ce + loss_boundary * LBABDA_BDY + loss_object
                    * LBABDA_OBJ
            loss.backward()
            optimizer.step()

            losses[iter_] = loss.data
            mean_losses[iter_] = np.mean(losses[max(0, iter_ - 100):iter_
               ])

            if iter_ % 1 == 0:
                clear_output()
                pred = np.argmax(output.data.cpu().numpy()[0], axis=0)
                gt = target.data.cpu().numpy()[0]
                print('Train (epoch {}/{}) [{}/{} ({:.0f}%)]\tLoss_ce:
                   {:.6f}\tLoss_boundary: {:.6f}\tLoss_object: {:.6f}\
                   tLoss: {:.6f}\tAccuracy: {}'.format(
                    e, epochs, batch_idx, len(train_loader),
                    100. * batch_idx / len(train_loader), loss_ce.data,
                       loss_boundary.data, loss_object.data, loss.data,
                       accuracy(pred, gt)))
            iter_ += 1

            del (data, target, loss)

        if e % save_epoch == 0:
            net.eval()
            MIoU = test(net, test_ids, all=False, stride=Stride_Size)
            net.train()
            if MIoU > MIoU_best:
                                if DATASET == 'Potsdam':
```

```
54                          torch.save(net.state_dict(), './resultsp/{}_epoch
                                {}_{}'.format(MODEL, e, MIoU))
55                  MIoU_best = MIoU
```

5 Results and analysis

The evaluation metrics used for the experiments are the average F1 score and the average intersection ratio (IoU), which are commonly used in semantic segmentation tasks to evaluate the classification accuracy and the overlap of segmented regions, respectively. As can be seen from the table below, after using the raw output of SAM to assist in optimizing the model, the overall performance of the model on several categories is significantly improved. Specifically, in the categories of buildings and trees, the optimized model improves the performance by about 1 point or so respectively. This suggests that by introducing the contextual information provided by SAM, the model is able to capture the boundaries and structures of objects more accurately, and improves the ability to distinguish between different feature categories, especially in complex urban environments.

	UNetFormer	SAM-assisted		UNetFormer	SAM-assisted
F1_ImSurf	0.9482	0.9509	IOU_ImSurf	0.8534	0.9384
F1_Building	0.9439	0.9518	IOU_Building	0.8899	0.9119
F1_LowVeg	0.8363	0.8454	IOU_LowVeg	0.7174	0.7335
F1_Tree	0.8906	0.9056	IOU_Tree	0.8109	0.8193
F1_Car	0.8670	0.8665	IOU_Car	0.7653	0.7645
F1_Clutter	0.5752	0.6086	IOU_Clutter	0.4037	0.4309
mF1	0.8921	0.9071	mIOU	0.8074	0.8335

Table 1. Performance Comparison

6 Conclusion and future work

In this study, auxiliary information based on the raw SAM output was used to optimize the UNetFormer model, and the experimental results showed that the overall performance was improved, especially in the categories such as Buildings and Trees. However, in some categories such as Car, the model effectiveness is rather reduced, which may be due to the fact that the SAM auxiliary information is not effective in improving the performance of this category. Despite the improved performance of UNetFormer, the lightweight effect of the model was weakened in real training and testing. Future research needs to further optimize the computational efficiency of the model and explore how to reduce the computational burden while maintaining accuracy. Further research can focus on a more efficient attention mechanism and lightweight design to enhance the model's performance for applications in complex remote sensing image segmentation tasks.

References

[1] Libo Wang, Rui Li, Ce Zhang, Shenghui Fang, Chenxi Duan, Xiaoliang Meng, and Peter M. Atkinson. Unetformer: A unet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 190:196–214, 2022.

[2] Xianping Ma, Qianqian Wu, Xingyu Zhao, Xiaokang Zhang, Man-On Pun, and Bo Huang. Sam-assisted remote sensing imagery semantic segmentation with object and boundary constraints. IEEE Transactions on Geoscience and Remote Sensing, 62:1–16, 2024.