Guided Patch-Grouping Wavelet Transformer with Spatial Congruence for Ultra-High Resolution Segmentation

Deyi Ji\textsuperscript{1,2}, Feng Zhao\textsuperscript{1,}\textsuperscript{*}, Hongtao Lu\textsuperscript{3,4}
\textsuperscript{1}University of Science and Technology of China
\textsuperscript{2}Alibaba Group
\textsuperscript{3}Department of Computer Science and Engineering, Shanghai Jiao Tong University
\textsuperscript{4}MOE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
jideyi@mail.ustc.edu.cn, fzhao956@ustc.edu.cn, htlu@sjtu.edu.cn

Abstract

Most existing ultra-high resolution (UHR) segmentation methods always struggle in the dilemma of balancing memory cost and local characterization accuracy, which are both taken into account in our proposed Guided Patch-Grouping Wavelet Transformer (GPWFormer) that achieves impressive performances. In this work, GPWFormer is a Transformer (T)-CNN (C) mutual learning framework, where T takes the whole UHR image as input and harvests both local details and fine-grained long-range contextual dependencies, while C takes downsampled image as input for learning the category-wise deep context. For the sake of high inference speed and low computation complexity, T partitions the original UHR image into patches and groups them dynamically, then learns the low-level local details with the lightweight multi-head Wavelet Transformer (WFormer) network. Meanwhile, the fine-grained long-range contextual dependencies are also captured during this process, since patches that are far away in the spatial domain can also be assigned to the same group. In addition, masks produced by C are utilized to guide the patch grouping process, providing a heuristics decision. Moreover, the congruence constraints between the two branches are also exploited to maintain the spatial consistency among the patches. Overall, we stack the multi-stage process in a pyramid way. Experiments show that GPWFormer outperforms the existing methods with significant improvements on five benchmark datasets.

1 Introduction

The analysis of ultra-high resolution (UHR) geospatial image with millions or even billions of pixels has opened new horizons for the computer vision community, playing an increasingly important role in a wide range of geosciences and urban construction applications, such as disaster control, environmental monitoring, land resource protection and urban planning [Chen et al., 2019; Guo et al., 2022; Ji et al., 2023]. The focus of this paper is on semantic segmentation, providing a better understanding by assigning each pixel into a specified category.

Fully convolution neural networks (FCN) based methods have driven rapid growth in segmentation for regular resolution images, but overlook the feasibility of larger scale input. Due to the memory limitation, earliest works for UHR image segmentation basically follow two paradigms: (1) downsampling the image to a regular resolution, or (2) cropping the image into small patches, feeding them to network sequentially and merging their predictions. Intuitively both the two paradigms will result in inaccurate results, the former loses many local details while the latter lacks of global context. After that, methods specially designed for UHR images are proposed and most of them follow the global-local collaborative framework to preserve both global and local information with two deep branches, taking the downsampled entire image and cropped local patches as inputs respectively. The most representative works are GLNet [Chen et al., 2019] and FCtL [Li et al., 2021]. Despite the considerable performance, their memory cost is high and inference speed is very low, due to the deep branches and sequentially inference. Later, following the bilateral architecture [Yu et al., 2018], ISDNet [Guo et al., 2022] proposes to combine a shallow and a deep branch. The shallow branch takes the whole UHR image as input and extracts multi-scale shallow features, while the deep branch takes the highly downsampled image as input to extract one deep feature, then the features are fused for final prediction with a specially designed fusion module. This type of architecture avoids local patches cropping and sequentially prediction thus improves the inference speed with a large margin, but also results in weaker performance, especially in local characterization, since the local details will never be fully addressed in the shallow branch with UHR input. In addition, their proposed fusion module also introduces much extra memory cost. In a word, due to enormous pixels, existing methods for UHR segmentation usually struggle in the pressing dilemma of balancing memory cost and local characterization accuracy.

To this end, in this paper we propose a novel Guided Patch-Grouping Wavelet Transformer (GPWFormer) network to address the above balance problem for UHR image segmentation. In general, we formulate a hybrid CNN-Transformer

\textsuperscript{*}Corresponding Author.
framework in dual-branch style, where the Transformer branch takes locally cropped UHR image as input and harvests both local details and fine-grained long-range dependencies, while the CNN branch takes downsampled UHR image as input for learning the category-wise deep context. For the sake of high inference speed and low computation complexity, different from the classical global-local framework, the local patches are fed into the Transformer branch all at once and dynamically grouped, then each group is fed into a different Transformer head to extract local texture details. Meanwhile fine-grained long-range dependencies are also captured during this process, since patches that are far away in spatial domain can also be assigned to a same group. In addition, inspired by [Yao et al., 2022], invertible down-sampling operations with dense wavelets are integrated into the Transformer for lightweight memory cost, and masks produced by CNN branch are utilized to guide the patch grouping process, providing a heuristics decision. Moreover, the congruence constraint between the two branches are also exploited to maintain the spatial consistency among the patches. In general, we stack the multi-stage process in a pyramid way.

Overall, our contributions are summarized as follows:

- We propose a novel Patch-Grouping Wavelet Transformer (GPWFormer) network for ultra-high resolution image segmentation, which is a hybrid CNN-Transformer in dual-branch style to harvest fine-grained both low-level and high-level context simultaneously in an efficient way.
- Specifically, we introduce a Wavelet Transformer to integrate the Transformer branch with dense wavelets for lightweight memory cost, and further decrease its computation complexity with heuristics grouping masks guided by the corresponding deep features of CNN branch.
- Moreover, the congruence constraints between the two branches are also exploited to maintain the spatial consistency among the patches.
- Extensive experiments demonstrate that the proposed GPWFormer obtains an excellent balance between memory cost and local segmentation accuracy, and outperforms the existing methods with significant improvements on five public datasets.

2 Related Work

2.1 Generic Semantic Segmentation

Deep learning based methods have taken a big step forward on computer vision [Goodfellow et al., 2016; He et al., 2016; Ji et al., 2019; Feng et al., 2018; Wang et al., 2021a; Wang et al., 2021b; Ji et al., 2021]. The development of deep CNN and Transformer over the past few years has driven rapid growth of methods in generic semantic segmentation for natural images and daily photos [Hu et al., 2020; Chen et al., 2018; Zhu et al., 2021]. Earlier semantic segmentation models for generic images was mainly based on the fully convolutional networks (FCN) [Long et al., 2015] and recent ones followed the design of Transformer network. FCN-based methods usually relied on large receptive field and fine-grained deep features, such as DeepLab [Chen et al., 2017; Chen et al., 2018], DANet [Fu et al., 2019], OCRNet [Yuan et al., 2020a]. While Transformer-based networks viewed segmentation as a Sequence-to-Sequence perspective and have become a new research hotspot. Representative works included SETR [Zheng et al., 2021] and Swin [Liu et al., 2021]. However, Transformer networks usually took amounts of memory cost and computation complexity, limiting its development on UHR image segmentation. In this paper, we aim to take advantage of the strong representation ability of Transformer meanwhile decrease its memory cost for UHR segmentation. In addition, knowledge distillation methods [Ji et al., 2022] have also been applied to make the network lightweight.

2.2 Ultra-High Resolution Image Segmentation

Benefited from the advancement of photography and sensor technologies, the accessibility and analysis of ultra-high resolution geospatial images has opened new horizons for the computer vision community, playing an increasingly important role in a wide range of geosciences and urban construction applications, including but not limited to disaster control, environmental monitoring, land resource protection and urban planning [Ji et al., 2023]. According to [Ascher and Pin-cus, 2007; Chen et al., 2019], an image with at least $4.1 \times 10^6$ pixels can reach the minimum bar of ultra-high definition media, usually deriving from a wide range of scientific applications, for example, geospatial and histopathological images. For ultra-high resolution image segmentation, CascadePSP [Cheng et al., 2020] proposed to improve the coarse segmentation results with a pretrained model to generate high-quality results. GLNet [Chen et al., 2019] incorporated both global and local information deeply in a two-stream branch manner. FCtL [Li et al., 2021] exploited a squeeze-and-split structure to fuse multi-scale features information. ISDNet [Guo et al., 2022] integrated the shallow and deep networks. These existing works lack of a further in-depth analysis and have obvious drawbacks analyzed in the Introduction section. WSDNet [Ji et al., 2023] was proposed as an efficient and effective framework for UHR segmentation especially with ultra-rich context.

3 Method

In this section, we introduce the proposed Guided Patch-Grouping Wavelet Transformer (GPWFormer) in detail. Firstly, we introduce the overall structure. Subsequently, we illustrate the Guided Patch-Grouping strategy and Wavelet Transformer respectively. Next the details of spatial congruence are explained.

3.1 Overview

Figure 1 shows the overall framework of the proposed GPWFormer, which consists of dual branches. The upper branch is a lightweight Transformer network taking the original UHR image as input to harvest both local structured details and fine-grained long-range spatial dependencies, while the lower is a deep CNN network taking the downsampled UHR image as input to learn category-wise deep context. For simplicity,
Figure 1: The overview of the proposed GPWFormer. Upper: Transformer branch (T); Lower: CNN branch (C). In each stage of T, the input UHR image/feature is evenly partitioned into patches and fed into the network all at once. Then, they are grouped guided by a mask generated from the corresponding features in C. Next, intra-group relations in each group are learned with their respective WFormer heads. 

We denote the two branches as T and C respectively. In C, any classical generic segmentation architecture can be utilized, here DeepLabV3+ with a lightweight backbone ResNet18 is employed. Besides, in order to further reduce its computation complexity, we integrate each stage of ResNet18 with discrete wavelet transform (DWT) to reduce the dimension of intermediate features, followed by the Atrous Spatial Pyramid Pooling (ASPP) module and inverse wavelet transform (IWT).

In T, the input UHR image is firstly evenly partition into local patches. Different from existing global-local frameworks that take into patches sequentially, all the patches are fed into T at once in our framework, which can greatly accelerate the inference process. In order to decrease the memory cost in this situation, we divide these patches into multiple groups, then intra-group relations for each group are learned with a corresponding shallow Wavelet Transformer (WFormer) head respectively, which is proposed for more effective and efficient learning. The relations consist of both local structured relations and long-range dependencies, since both spatially adjacent and nonadjacent patches can be assigned to a same group. Meanwhile all the low-level pixel-wise details are also captured during the process. Multiple stages can be stacked in a pyramid way for fully characterization, and the stage number is set to 2 here for higher inference speed, which can also be set to a flexible number according to the practical needs. The number of groups among stages are set in a pyramid manner to obtain representations of different granularity. It is worthy noted that the corresponding features from C are employed to guide the patch-grouping, providing a heuristics decision. Moreover, after the learning of each stage in T, we also add a congruence constraint to the inter-patch relations by the corresponding stage of C to maintain the spatial consistency. Finally the output of T and C are fused and supervised with a focal loss.

3.2 Guided Patch-Grouping

In each stage of T, we produce a guidance for patch-grouping from the corresponding feature from C, as shown in Figure 1, and the first stage is taken as example for the following illustration. Given the UHR input \( I \in \mathbb{R}^{H_1 \times W_1} \) in T, we evenly partition it into \( m \times n \) patches, which are then divided into \( G \) groups for the subsequent WFormer heads. Let \( h, w \) denote the height and width of each patch respectively. Our aim is to produce a guidance mask \( M \in \mathbb{R}^{m \times n} \) from the C, \( M_{i,j} \in [1, G] \) denotes the group index of patch \((i, j)\), where \( \{i \in [1, m], j \in [1, n]\} \).

Concretely, we employ the low-frequency subband \( F_{\text{LL}} \in \mathbb{R}^{C_1 \times H_1 \times W_1} \) of wavelet transform for the feature after ResNet-S1 in C to produce \( M \), since low-frequency subband is able to preserve more spatial details. \( C_1, H_1, \) and \( W_1 \) denote the channel, height, width of \( F_{\text{LL}} \) respectively. Firstly, channel of \( F_{\text{LL}} \) is decreased to \( G \) with an intuitive PCA method instead of a typical convolution operator, so that the whole mask-producing process is not necessary to be learnable, thereby reducing the amount of calculation, while avoiding the design of complex forward propagation and gradient backward processes. Next following the same partition operations for \( I \) in T, we apply a patch-wise average pooling to \( F_{\text{LL}} \), resulting in the mask feature \( F^M \in \mathbb{R}^{G \times m \times n} \). Formally, the process of patch-wise average pooling is denoted as,

\[
F^M_{i,j} = \text{GAP}(F_{\text{LL}}^{i \times h_1 : (i+1) \times h_1, j \times w_1 : (j+1) \times w_1}),
\]

\[
h_1 = H_1 / m, w_1 = W_1 / n, i \in [1, m], j \in [1, n],
\]

where \( \text{GAP}(\cdot) \) denotes global average pooling. The above process can also be viewed as an average pooling with
row_stride = h_1 and column_stride = w_1. Then we apply a softmax function to F^M, generate the score mask S ∈ \mathbb{R}^{G \times m \times n}, and argmax S along the last two dimension to generate the mask M.

\[ M = \text{arg max}_{m,n} (S). \tag{2} \]

Noted that M may not divide the groups evenly, that is, not all groups may have equal number of patches, and some groups may have more than \( \frac{\text{m.n}}{c} \) patches, thus we sort the patches in descending order of scores in these groups respectively, then re-distribute their last extra patches to other groups based on scores.

### 3.3 Wavelet Transformer

In our framework, the input to each Transformer head is a group of patch features. Taking any group in first stage of T as example for illustration, let \( \{ F_u \} \in \mathbb{R}^{h \times w \times C} \) denote the set of input patch features in the group, where \( u \) is patch index. Following the design of [Yao et al., 2022], we introduce the Wavelet Transformer to make the network lightweight. For each \( F_u \), we first reduce its channel to \( C/4 \) with a convolution layer, then 2D DWT is utilized to decompose its four wavelet subbands \( \{ F_u^{LL}, F_u^{LH}, F_u^{HL}, F_u^{HH} \} \) with four filters \( f_{LL}, f_{LH}, f_{HL}, f_{HH} \). The dimension of both the subbands is \( \frac{h}{2} \times \frac{w}{2} \times C \). \( F_u^{LL} \) is the low-frequency subband while the others are high-frequency ones, next the four subbands are concatenated to \( F_u \in \mathbb{R}^{\frac{h}{2} \times \frac{w}{2} \times C} \). By this way, the dimension of all patch features are reduced so that the computation complexity is squarely decreased. Noted that despite the downsampling operation is deployed, due to the biorthogonal property of DWT, the original feature can be accurately reconstructed by the inverse wavelet transform (IWT) in subsequent steps. Finally, \( \{ \tilde{F}_u \} \) is projected into a sequence \( \tilde{F} \in \mathbb{R}^{L \times C} \) and embedded spatial information as previous general Transformer networks [Zheng et al., 2021], and formulated as input to corresponding Transformer head.

As shown in Figure 2, following the general design of Transformer for segmentation, WFormer also contains an encoder and a decoder, where encoder consists of \( L_e \) layers of multi-head self-attention (MSA) and MLP blocks. The input to self-attention of each layer \( l \) is in a triplet of \( (Q, K, V) \), calculated by the current input \( Y^{l-1} \in \mathbb{R}^{L \times C} \) (the input for the first layer is \( \tilde{F} \)), with three learnable linear projection layers \( W_Q, W_K, W_V \in \mathbb{R}^{C \times d} \) (\( d \) is the dimension), as:

\[ Q = Y^{l-1}W_Q, K = Y^{l-1}W_K, V = Y^{l-1}W_V. \tag{3} \]

3.4 Spatial Consistency Congruence

With multiple Transformer heads, the patches of input feature are learned in groups, which affects the translation-invariant and results in the spatial in-consistency among the patches. Since \( C \) learns the input image globally and is able to preserve the overall spatial consistency, we propose to utilize congruence constraints from \( C \) to each stage of \( T \), to maintain the spatial congruence of inter-patch relations in \( T \).

Considering the stage \( s \) in \( T \), given the patch features set \( \{ P_k^{T_s} \} (k \in \{ 1, m_s \times n_s \}) \) after WFormer heads, where \( m_s \times n_s \) is the number of patches in stage \( s \) of \( T \) (denoted as \( T_s \)), we first calculate their inter-patch relations \( C_s^T \), which defines as the sum of relations between any two patch centers. The center of a patch is defined as the average of its all pixel features. To capture the high-order relations in an efficient manner, inspired by the theory of SVM [Han et al., 2012], we employ the metric with Gaussian Radial Based Function (RBF) kernel to calculate the inter-patch relations. Therefore
\( C_s^T \) can be formulated as,
\[
C_s^T = \sum_{k_1=1}^{m_s} \sum_{k_2=1}^{n_s} R(A(P_{k_1}^{T_s}), A(P_{k_2}^{T_s})),
\]
where \( R(\cdot), A(\cdot) \) are the RBF kernel and Average function respectively. To simplify the calculation, we replace the RBF calculation with its sum of \( T \)-order Taylor series,
\[
R(A(P_{k_1}^{T_s}), A(P_{k_2}^{T_s})) = e^{-\theta \cdot ||A(P_{k_1}^{T_s}) - A(P_{k_2}^{T_s})||^2},
\]
where \( \theta \) is temperature parameter, \( T \) is matrix transposition operator.

Similarly, given the feature in corresponding stage of \( C \), we partition it as the same style of \( \{ P_{k}^{T_s} \} \), and calculate its inter-patch relations \( C_s^C \), so the spatial consistency constraint can be formulated,
\[
C = \sum_s \frac{1}{m_s \times n_s} ||C_s^C - C_s^T||^2,
\]
which is implemented as a loss cooperating with the main focal loss.

So the overall loss \( L \) is a weighted combination of a main focal loss and the above spatial consistency constraint,
\[
L = L_{focal} + \alpha C,
\]
where \( \alpha \) is the loss weight and set to 0.8.

4 Experiments
4.1 Datasets and Evaluation Metrics
In order to validate the effectiveness of our proposed method in a wide perspective, we perform experiments on five datasets, including DeepGlobe, Inria Aerial, Cityscapes, ISIC and CRAG.

DeepGlobe
The DeepGlobe dataset [Demir et al., 2018] have 803 UHR images, with 455/207/142 images for training, validation and testing. Each image contains 2448 × 2448 pixels and the dense annotation contains seven classes of landscape regions.

Inria Aerial
The Inria Aerial Challenge dataset [Maggiiori et al., 2017] has 180 UHR images captured from five cities. Each image contains 5000 × 5000 pixels and is annotated with a binary mask for building/non-building areas, with 126/27/27 images for training, validation and testing.

CityScapes
The Cityscapes dataset [Cordts et al., 2016] has 5,000 images of 19 semantic classes, with 2,979/500/1,525 images for training, validation and testing.

ISIC
The ISIC Lesion Boundary Segmentation Challenge dataset [Tschandl et al., 2018] contains 2596 UHR images, with 2077/260/259 images for training, validation and testing.

<table>
<thead>
<tr>
<th>CNN</th>
<th>Transformer</th>
<th>P.G.</th>
<th>Wave.</th>
<th>Cong.</th>
<th>mIoU (%)</th>
<th>Mem (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>74.1</td>
<td>924</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>74.0</td>
<td>924</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>74.8</td>
<td>75.8</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>75.9</td>
<td>3730</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>76.1</td>
<td>6090</td>
</tr>
</tbody>
</table>

Table 1: Effectiveness of each component in GPWFormer. “P.G., Wave., Cong.” indicate Patch-Grouping, Wavelet Transform, and Spatial Congruence, respectively.

<table>
<thead>
<tr>
<th>Grouping Strategy</th>
<th>Linear (row)</th>
<th>Linear (column)</th>
<th>Rectangle</th>
<th>Guided</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>74.1</td>
<td>74.0</td>
<td>74.8</td>
<td>75.8</td>
</tr>
</tbody>
</table>

Table 2: Comparisons of patch-grouping strategy.

CRAG
CRAG [Graham et al., 2019] dataset includes two classes and exhibits different differentiated glandular morphology, with 173/40 images for training/testing. Their average size is 1512 × 1516.

Evaluation Metrics
In all experiments, we adopt the mIoU, F1 score, Accuracy and memory cost to study the effectiveness.

4.2 Implementation Details
We use the MMSegmentation codebase [MMSegmentation, 2022] following the default augmentations without bells and whistles, and train on a server with Tesla V100 GPUs with batch size 8. During the image cropping of Transformer branch, the size of patches are 500 × 500 pixels and neighboring patches have 120 × 580 pixels overlap region in order to avoid boundary vanishing. The input image of CNN branch is also downsampled to size 500 × 500 to trade-off performance and efficiency. We employ three layers in WFormer encoder and pre-train the Transformer on the Imagenet-1K dataset. During training, Focal loss [Lin et al., 2017] with \( \lambda = 2 \) is used to supervise training, and Adam [Kingma and Ba, 2014] optimizer is used in the optimization process. We set the initial learning rate to 5 × 10^{-5}, and it is decayed by a poly learning rate policy where the initial learning rate is multiplied by \( (1 - \frac{iter}{total\_iter})^{0.9} \) after each iteration. The maximum iteration number set to 40k, 80k, 160k and 80k for Inria Aerial, DeepGlobe, Cityscapes and ISIC respectively. We use the command line tool “gpustat” to measure the GPU memory, with the mini-batch size of 1 and avoid calculating any gradients.

4.3 Ablation Study
In this section, we delve into the modules and settings of our proposed model and demonstrate their effectiveness. All the ablation studies are performed on DeepGlobe test set.
Effectiveness of GPWFormer

We conduct experiments to verify the effectiveness of different components in GPWFormer, as shown in Table 1. Firstly, only CNN branch or Transformer branch will lead to lightweight memory cost but lower performance, indicating the effectiveness of the hybrid CNN-Transformer architecture. Then dual-branch with all components achieves impressive and balanced results. Removing “Spatial Congruence” module leads to a lower mIoU since the spatial consistency among groups is mismatched, while memory cost is steady as this module is only used in training. Using an ordinary Transformer without wavelets has a less memory, showing that WFormer is able to improve the efficiency while maintaining a comparable performance. Applying WFormer on the whole image results in better performance while severely degraded inference speed, proving that patch-grouping strategy indeed decreases the computation complexity.

Comparisons of Patch-Grouping Strategies

We show the superiority of the proposed guided patch-grouping strategy in Table 3, and several other strategies are also utilized for comparison. “Linear (row)” and “Linear (column)” mean grouping the patches in order of rows and columns respectively, and “Rectangle” means in the order of rectangular boxes. Experiments show that guided grouping shows best performance, proving that CNN branch indeed provides an effective guidance for patch grouping of Transformer branch.

Comparisons of Downsampling Methods in WFormer

We show the comparison of self-attention block with different downsampling methods in Table 4. Classical self-attention with ordinary average pooling or transposed convolution shows much lower performance, since both of them are irreversible and lose much information during the downsampling operations, while wavelet transform is invertible thus all the information can be persevered. A self-attention block with transposed convolution achieves a high mIoU than with average pooling, but also introducing extra memory cost.

The Impact of Settings in Spatial Congruence

The comparison of different congruence methods is shown in Table 3. Here besides the Gaussian RBF kernel, we also implement some other metrics, including “Mean” and “Inner-Dot”. The former indicates calculating inter-patch relation of two patches as distance between their mean feature, while the latter as the Inner-Product. Experiments shown that Gaussian RBF is more flexible and powerful in capturing the complex non-linear relationship between high-dimensional patch features.

4.4 Comparison with State-of-the-Arts

In this section, we compare the proposed framework with existing state-of-the-art methods, including U-Net [Ronneberger et al., 2015], ICNet [Zhao et al., 2018], PPN [Wu et al., 2020], PSPNet [Zhao et al., 2017], SegNet [Badrinarayanan et al., 2017], DeepLabv3+ [Chen et al., 2018], FCN-8s [Long et al., 2015], CascadePSP [Cheng et al., 2020], BiseNet [Yu et al., 2018], PointRend [Kirillov et al., 2020], DenseCRF [Krähenbühl and Koltun, 2011], DGF [Wu et al., 2018], DANet [Fu et al., 2019], SegFix [Yuan et al., 2021], MagNet [Hyunh et al., 2021], STDC [Fan et al., 2021], GLNet [Chen et al., 2019], FCtL [Li et al., 2021], ISDNet [Guo et al., 2022] and WSDNet [Li et al., 2023], on DeepGlobe, Inria Aerial, Cityscapes, ISIC and CRAG datasets, in terms of mIoU (%), F1 (%), Accuracy (%), Memory Cost (M).

Some of these methods are specially designed for UHR images (denoted as “UHR Model”) and the others are not (denoted as “Generic Model”). We show the results of “Generic Model” on both “Global Inference” and “Local Inference”. The former obtains the prediction with downsampled global
image, and the latter obtains the prediction with local cropped patches sequentially and then merges their results by post-processing.

**DeepGlobe**

As shown in Table 5, we first compare GPWFormer with above-mentioned methods on DeepGlobe test dataset. Due to the diversity of land cover types and the high density of annotations, this dataset is very challenging. The experiments show that GPWFormer outperforms all other methods on both mIoU, F1 and Accuracy. Specifically, we outperform GLNet, ISDNet and FCtL by large margins on mIoU respectively, directly showing the segmentation effectiveness and performance improvement. Besides, the categories in the dataset are often seriously unbalanced distributed, so we exploit the F1 score and Accuracy metrics to reflect the improvements and experiment results show the proposed method also achieves the highest scores among all the models. With such impressive performance, our methods is economic in the memory cost, attaining an excellent balance among accuracy and memory cost.

**Inria Aerial**

We also show the comparisons on Inria Aerial test dataset in Table 6. This dataset is more challenging, since the number of pixels for each image reaches 25 million, which is around four times than DeepGlobe, and the foreground regions are also finer. Experiment results show that GPWFormer outperforms GLNet, ISDNet and FCtL by large margins again on mIoU respectively, with comparable memory cost.

**Cityscapes**

To further validate the generality of our method, we also show the results on Cityscapes dataset, as shown in Table 7. GPWFormer also outperforms all other methods on mIoU, with a bright results on memory cost.

**ISIC and CRAG**

The image resolution of ISIC is comparable to Inria Aerial, and CRAG is of lower image resolution than other datasets. Table 8 shows the experimental results. GPWFormer once achieves excellent performances.

### Table 6: Comparison with state-of-the-arts on Inria Aerial test set.

<table>
<thead>
<tr>
<th>Generic Model</th>
<th>mIoU (%)↑</th>
<th>F1 (%)↑</th>
<th>Acc (%)↑</th>
<th>Mem (M)↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepLabv3+</td>
<td>55.9</td>
<td>-</td>
<td>-</td>
<td>5122</td>
</tr>
<tr>
<td>FCN-8s</td>
<td>69.1</td>
<td>81.7</td>
<td>93.6</td>
<td>2447</td>
</tr>
<tr>
<td>STDC</td>
<td>72.4</td>
<td>-</td>
<td>-</td>
<td>7410</td>
</tr>
</tbody>
</table>

**UHR Model**

| CascadePSP    | 69.4      | 81.8    | 93.2     | 3236     |
| GLNet         | 71.2      | -       | -        | 2663     |
| ISDNet        | 74.2      | 84.9    | 95.6     | 4680     |
| FCtL          | 73.7      | 84.1    | 94.6     | 4332     |
| WSDNet        | 75.2      | 86.0    | 96.0     | 4379     |
| GPWFormer (Ours) | **76.5** | **86.2** | **96.7** | 4710     |

### Table 7: Comparison with state-of-the-arts on Cityscapes test set.

<table>
<thead>
<tr>
<th>Generic Model</th>
<th>mIoU (%)↑</th>
<th>Mem (M)↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiseNetV1</td>
<td>74.4</td>
<td>2147</td>
</tr>
<tr>
<td>BiseNetV2</td>
<td>75.8</td>
<td>1602</td>
</tr>
<tr>
<td>PSPNet</td>
<td>74.9</td>
<td>1584</td>
</tr>
<tr>
<td>DeepLabv3</td>
<td>76.7</td>
<td>1468</td>
</tr>
</tbody>
</table>

**UHR Model**

| DenseCRF      | 62.9      | 1575     |
| DGF           | 63.3      | 1727     |
| SegFix        | 65.8      | 2033     |
| MagNet        | 67.6      | 2007     |
| MagNet-Fast   | 66.9      | 2007     |
| ISDNet        | 76.0      | 1510     |
| GPWFormer (Ours) | **78.1** | 1897     |

### Table 8: Comparison with state-of-the-arts on CRAG and ISIC test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>ISIC mIoU (%)</th>
<th>CRAG mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSPNet</td>
<td>77.0</td>
<td>88.6</td>
</tr>
<tr>
<td>DeepLabV3+</td>
<td>70.5</td>
<td>88.9</td>
</tr>
<tr>
<td>DAnet</td>
<td>51.4</td>
<td>82.3</td>
</tr>
<tr>
<td>GLNet</td>
<td>75.2</td>
<td>85.9</td>
</tr>
<tr>
<td>GPWFormer (Ours)</td>
<td><strong>80.7</strong></td>
<td><strong>89.9</strong></td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, we focus on the ultra-high resolution image segmentation and develop a novel Guided Patch-Grouping Wavelet Transformer network. We firstly analyze the limitations of existing state-of-the-art methods, and the unique difficulties of ultra-high resolution image segmentation. With CNN-Transformer dual-branch, we propose the Wavelet Transformer with a Guided Patch-Grouping strategy to learn local details and long-range spatial dependencies simultaneously, while the CNN branch takes the downsampled input UHR image with wavelet transform to capture deep category-wise context. Moreover, the congruence constraint is also introduced to maintain the spatial consistency from CNN branch to Transformer branch. Our proposed framework achieves new state-of-the-art results on five benchmark datasets.

**Acknowledgements**

This work was supported by the JKW Research Funds under Grant 20-163-14-LZ-001-004-01, Anhui Provincial Natural Science Foundation under Grant 2108085UD12, National Key R&D Program of China under Grant 2020AAA0103902, NSFC (No. 62176155), Shanghai Municipal Science and Technology Major Project, China (2021SHZDZX0102). We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.
References


[Cordts et al., 2016] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In IEEE Conference on Computer Vision and Pattern Recognition, 2016.


