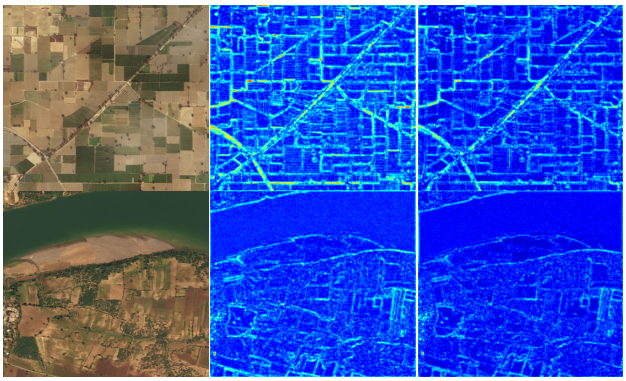
Supplementary Materials for "ISDNet: Integrating Shallow and Deep Networks for Efficient Ultra-high Resolution Segmentation"



(a) (b) (c)

Figure 1. Comparison of feature maps for different input types. (a) Input images. (b) Feature map from the high frequency inputs. (b) Feature map from the image input.

1. Experiment setting details

1.1. FPS Test

We design a Frames-Per-Second (FPS) test script (Algorithm 1) to evaluate the inference running time. The FPS script calculates the average running time of 100 consecutive model inferences without calculating gradients. The input tensor $x_0 \in \mathbb{R}^{1\times3\times H\times W}$ is a random tensor, which does not have impact on performance. We use Python *time.perf_counter()* to calculate time, rather than *time.time()*, since it is more precise. Besides, we wait for all kernels in all streams to finish on a CUDA device, then we record the start and end time, denoted by t_{start} and t_{end}.

Note that all methods do **not** use optimization from packages like TensorRT. Before each test, at least 20 forward pass is conducted as warm-up of the device. For each new method to be tested, we keep running warm-up trials of a recorded method until the recorded FPS is reached again, so we can guarantee a similar peak machine condition as before. The evaluation platform is a 2080 Ti GPU, with CUDA 10.1, CuDNN 7.6.5, PyTorch 1.6.0.

1.2. The image size for testing

For a fair comparison, we align the settings with most previous methods. Specifically, full-scale inputs are used in Cityscapes [4] for all previous and our methods. For DeepGlobe [5] and Inria Aerial [10], downsampling rate is 0.8 for ours, Deeplabv3 [1], and FCN8s [9], while 0.6 is used in UNet [11]. For the Global Local Refinement methods, *i.e.* GLNet [2] and FCtL [8], we keep their original settings, full scale for global and $\frac{1}{4}$ scale for patches.

Algorithm 1: The pseudo code for FPS test

Input: net: The test model; $x_0 \in \mathbb{R}^{1\times3\times H\times W}$: The Input tensor;
Output: FPS: the FPS of the test model
// load the model and tensor into the GPU device
$net.cuda()$
$x_0.cuda()$
// Warm-up the deivce
for $i \leftarrow 0$; $i < 20$; $i \leftarrow i + 1$ **do**
  out $\leftarrow net.forward(x_0)$;
// compute the FPS
$t_{total} \leftarrow 0$;
for $j \leftarrow 0$; $j < 100$; $i \leftarrow j + 1$ **do**
  with $torch.no\_grad()$:
  torch.cuda.synchronize();
  $t_{start} \leftarrow time.perf\_counter()$;
  out $\leftarrow net.forward(x_0)$;
  torch.cuda.synchronize();
  $t_{end} \leftarrow time.perf\_counter()$;
  $t_{total} \leftarrow t_{total} + (t_{end} - t_{start})$;
FPS $\leftarrow 100/t_{total}$;
final;

2. Experimental results

2.1. Fine details in residual inputs

The Laplacian pyramid decomposition guarantees that no information has been lost, *i.e.* the original image can be reconstructed from the inputs of $\mathcal{D}$ branch and $\mathcal{S}$ branch. Therefore, the information of two sides is complementary. On the other hand, from the perspective of only one branch,

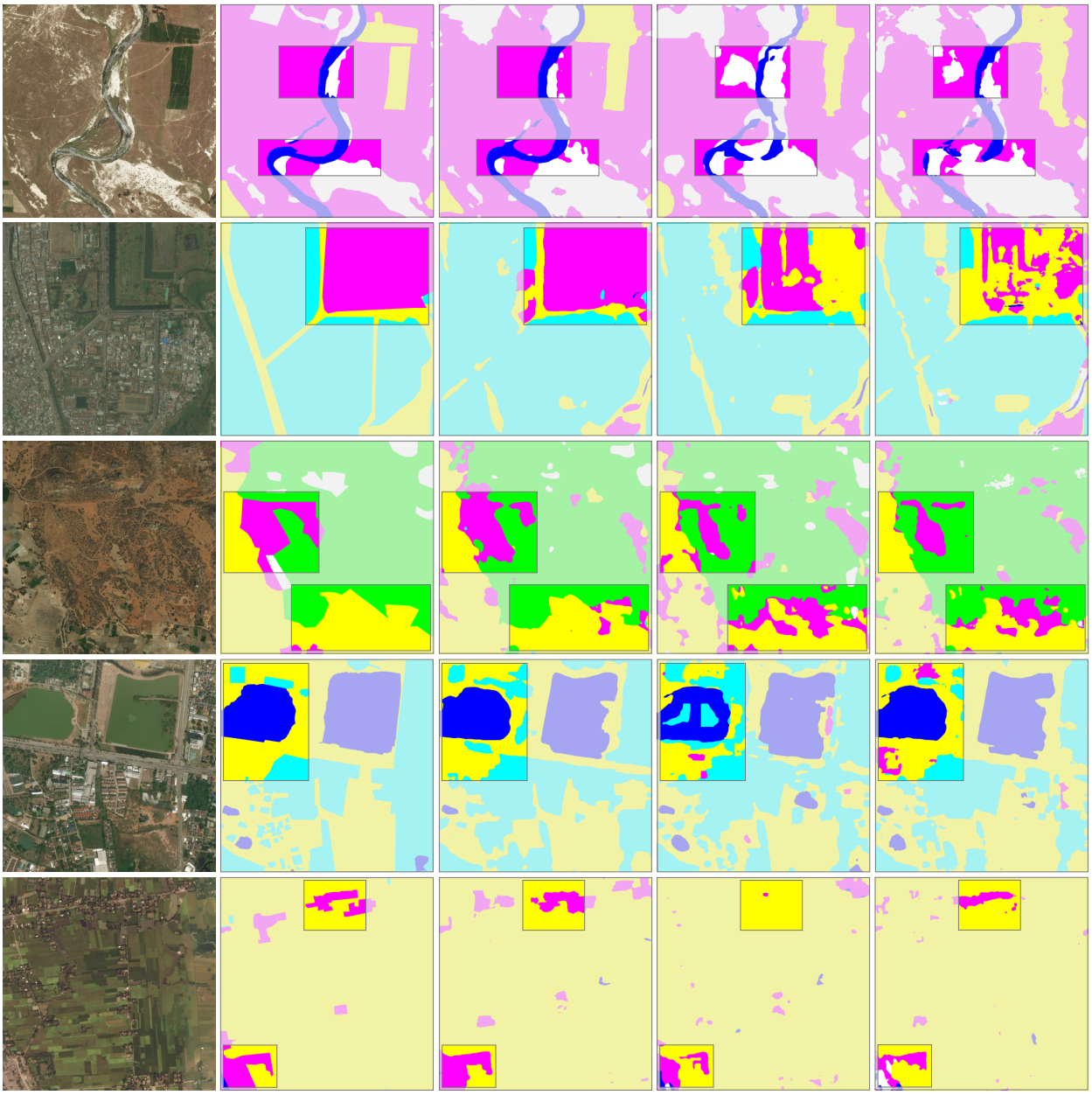


Figure 2. We illustrate several examples of the DeepGlobe dataset, comparing with the SOTAs. In this figure, masks with varied colors represent different semantic regions. Particularly, cyan represents "urban", yellow represents "agriculture", purple represents "rangeland", green represents "forest", blue represents "water", white represents "barren" and black represents "unknown".

it is indeed lost, but making information loss is common in regularization to help training (*e.g.* dropout). Besides, Fig. 1(b) shows clearer boundaries than (c), which proves that residuals help the network to learn fine-grained details.

2.2. Other metrics

As shown in Table 1, we also employ pixel acc (Acc). and F1 score to validate the clarity of semantic boundary. Our method reaches 88.7 pixel acc. and 84.0 F1 score on DeepGlobe dataset. Both metrics are better and much faster inference than previous methods, *e.g.* FCtL (88.3 pixel acc., 83.8 F1). Moreover, on our approach achieves a clear improvement on Inria Aerial dataset. In sum, the above results show the superiority of our method.

2.3. Comparison with different backbones

We choose ResNet18 as the backbone for the sake of speed and accuracy. We have tried to replace it with HR-

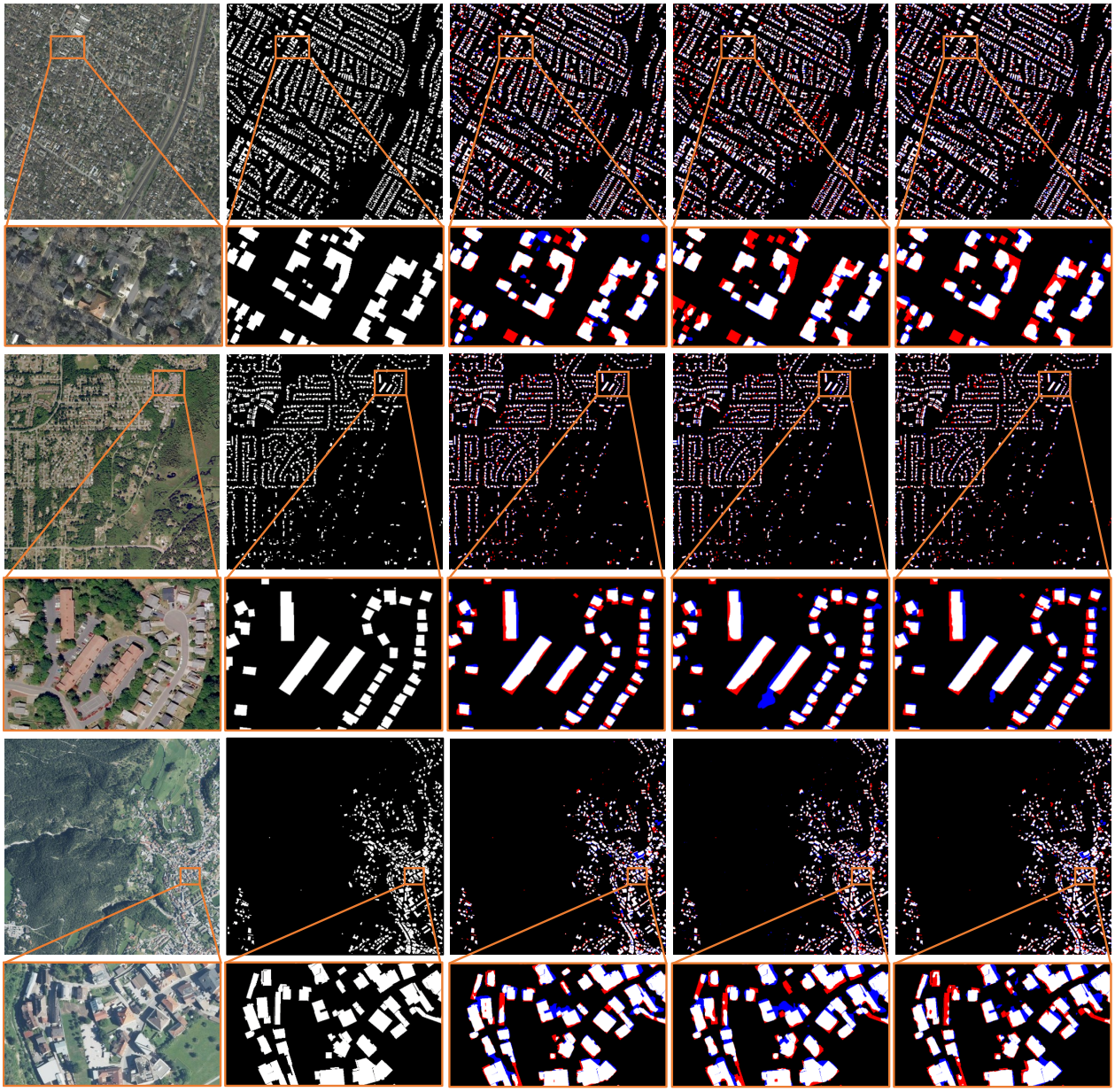


Figure 3. We illustrate several examples of the Inria Aerial dataset, comparing with the SOTAs. In this figure, white and black represent building and non-building respectively. Besides, in the segmentation results, we employ red and blue to mark the area with misclassification. Specifically, red represents foreground is classified into background, and vice versa for blue.

Net18 [12]. As shown in Table 2, our method still reaches 72.81 mIoU and 12.75 FPS on Deepglobe, which is similar in accuracy, but much faster than MagNet [7] (72.96, 0.8).

2.4. More qualitative results

We provide more comparison results with FCtL [8] and STDC [6]. Fig. 2 and Fig. 3 shows comparison results on DeepGlobe [5] and Inira Aerial [10], respectively. As shown in Fig. 2, our results are better, in both thin, long objects (*e.g.* the first row in Fig. 2) and lager areas (*e.g.* the

	DeepGlobe		Inria Aerial	
method	F1	Acc	F1	Acc
CascadePSP [3]	79.7	85.6	81.8	93.2
GLNet [2]	83.2	88.0	-	-
FCtL [8]	83.8	88.3	84.1	94.6
Ours	**84.0**	**88.7**	**84.9**	**95.6**

Table 1. F1 and Acc on the DeepGlobe and Inria Aerial.

Backbone of deep branch	mIoU	FPS
ResNet18	73.30	27.70
HRNet18	72.81	12.75

Table 2. Comparison with different backbones on DeepGlobe.

second and fourth rows in Fig. 2). The segmentation results in Fig. 3 also show the superiority of our method. Hence, our method outperforms the compared methods.

3. Method Comparison and Runtime Analysis

Since our method is distinctly different from previous global-local refinement methods, here we only discuss the novelty compared to previous lightweight models. Bilateral architecture is widely used for real-time segmentation *e.g*. BiSeNet [13]. However, we aim at designing a generic framework for UHR image segmentation, which brings the major difference that our method can reuse the architecture and weights of existing deep and shallow segmentation models for different scale inputs to make a better balance for speed and accuracy. In addition, ICNet [14] proposes a cascade pipeline to iterative refine the prediction from multiple scales, in which the final prediction is highly dependent on the prediction from small scales. In contrast, we used a parallel prediction with heterogeneous input, and the segmentation head of low resolution is discarded.

Our method benefits from the heterogeneous inputs for deep and shallow branches, while the high frequency residual inputs for deep branches require additional computation. In this paper, we keep a small kernel size to reduce the computation. Besides, Gaussian blur has been highly optimized in OpenCV and it only takes around 5ms to build the Laplacian pyramid on CPU, so it can be merged to the data prefetch stage, and parallel with GPU inference.

References

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[2] Wuyang Chen, Ziyu Jiang, Zhangyang Wang, Kexin Cui, and Xiaoning Qian. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8924–8933, 2019.

[3] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: toward class-agnostic and very high-resolution segmentation via global and local refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8890–8899, 2020.

[4] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 172–181, 2018.

[6] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9716–9725, 2021.

[7] Chuong Huynh, Anh Tuan Tran, Khoa Luu, and Minh Hoai. Progressive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16755–16764, 2021.

[8] Qi Li, Weixiang Yang, Wenxi Liu, Yuanlong Yu, and Shengfeng He. From contexts to locality: Ultra-high resolution image segmentation via locality-aware contextual correlation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7252–7261, 2021.

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[10] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3226–3229. IEEE, 2017.

[11] O Ronneberger, P Fischer, and T Brox. U-net: Convolutional networks for biomedical image segmentation. arxiv. *Lecture Notes in Computer Science*, 2015, 2015.

[12] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019.

[13] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018.

[14] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 405–420, 2018.