Supplementary Material: Automatic High Resolution Wire Segmentation and Removal

A. Comparison with Pixel 6

We show a visual comparison between our model and Pixel 6's "Magic Eraser" feature in Figure 1. Without manual intervention, Google Pixel 6's "Magic Eraser" performs well on wires with clean background, but suffers from thin wires that are hardly visible ((A) upper), and also on wires with complicated background ((A) lower). We also pass our segmentation mask to our wire inpainting model to acquire the wire removal result, as shown in the lower image of (B).

B. Failure cases

We show some challenging cases where our model fails to predict accurate wire masks in Figure 2. These include regions that are very similar to wires (top row), severe background blending (middle row) and extreme lighting conditions (bottom row).

C. Panorama

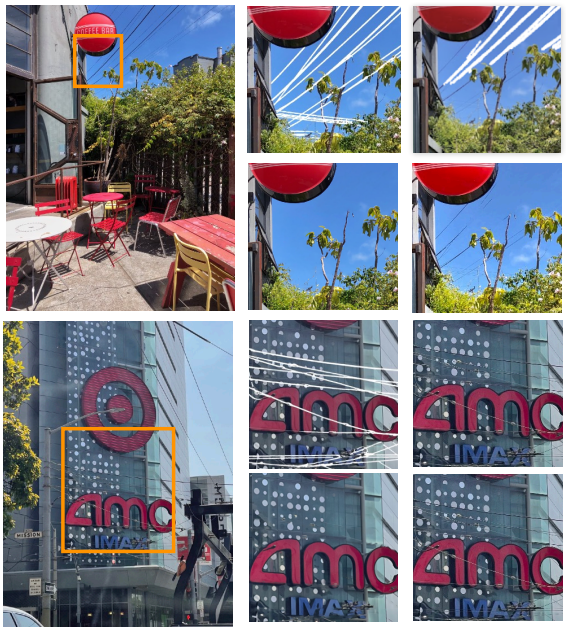
Our two-stage model leverages the sparsity of wires in natural images, and efficiently generalizes to ultra-high resolution images such as panoramas. We show one panoramic image of 11K by 1.5K resolution in Figure 3. Note that our method produces high-quality wire segmentation that covers wires that are almost invisible. As a result, our proposed wire removal step can effectively remove these regions.

D. Segmentation and inpainting visualizations

We show our wire segmentation and inpainting results in several common photography scenes as well as in some challenging cases in Figure 4. We provide more visualizations of wire segmentation and subsequent inpainting results. Our model successfully handles numerous challenging scenarios, including strong backlit (top row), complex background texture (2nd row), low light (3rd row), and barely visible wires (4th row). A typical use case is shown in the last row.

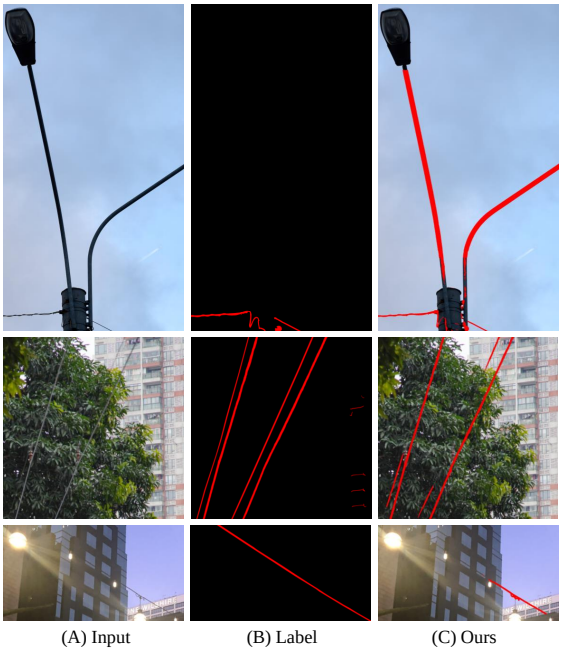
E. Experiments on other datasets

Most existing wire-like datasets either are at low resolutions or are for specific purposes (e.g., aerial imaging) and thus do not contain the scene diversity like WireSegHR does. The suggested TTPLA [2] dataset shares the Power Lines class with our dataset, although it only contains aerial images. Table 1 shows evaluation results of the TTPLA dataset on our model and also our WireSegHR dataset on the TTPLA model.



(A) Input (B) Ours (C) Pixel 6

Figure 1. **Comparison with Pixel 6**. Our model can pick up hardly visible wires that even in complicated backgrounds



(A) Input (B) Label (C) Ours

Figure 2. **Failure cases**. In some challenging cases, our model fails to predict accurate masks. Zoom in to see detailed wire masks in ground truth and prediction.

Figure 3. **Segmentation and inpainting result for a panoramic image.** Our model is scalable to very large images with very thin wires.

Dataset	Model	IoU (%)
TTPLA (Power Line only)	TTPLA (ResNet-50, 700× 700)	18.9
	Ours (ResNet-50)	33.1
	Ours (MiT-b2)	42.7
WireSegHR	TTPLA (ResNet-50, 700× 700)	3.5
	Ours (ResNet-50)	47.8
	Ours (MiT-b2)	60.8

Table 1. Comparison with TTPLA.

TTPLA is trained on fixed resolution (700 $\times$ 700) and takes in the entire image for inference, which requires significant downsampling of our test set. As a result, the quality of thin wires deteriorates in both the image and the label. Our model drops in performance on the TTPLA dataset due to different annotation definitions: we annotate all wire-like objects while TTPLA only annotates power lines.

F. Additional training details

CascadePSP [1] We follow the default training steps provided by the CascadePSP code[1]. During training, we sample patches in the image that contain at least 1% of wire pixels. During inference, we feed the predictions of the global DeepLabv3+ to the pretrained/retrained CascadePSP model to get the refined wire mask. In both cases, we follow the default inference code[1] to obtain the final mask.

MagNet [3] MagNet[2] obtains the initial mask predictions from a single backbone trained on all refinement scales. For a fair comparison, we adopt a 2-scale setting of MagNet, similar to our two-stage model, where the image is downsampled to 1024×1024 in the global scale, and is kept at the original resolution in the local scale. To this end, we train a single DeepLabv3+ model by either downsampling the sample image to 1024×1024 or randomly cropping 1024×1024 patches at the original resolution. The sampled patches contain at least 1% of wire pixels. We then train the refinement module based on the predictions from the DeepLabv3+ model, following the default setting. Inference is kept the same as the original MagNet model.

ISDNet [2] ISDNet[3] performs inference on the entire image without sliding window. As a result, during training, we resize all images to 5000×5000 and randomly crop 2500×2500 windows, such that the input images can fit inside the GPUs. Sampled patches should contain 1% wire pixels. During inference, all images are resized to 5000×5000. We observe that this yields better results than if we keep images below 5000×5000 at their original sizes.

[1] https://github.com/hkchengrex/CascadePSP

[2] https://github.com/VinAIResearch/MagNet

[3] https://github.com/cedricgsh/ISDNet

References

[1] 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. 2

[2] Shaohua Guo, Liang Liu, Zhenye Gan, Yabiao Wang, Wuhao Zhang, Chengjie Wang, Guannan Jiang, Wei Zhang, Ran Yi, Lizhuang Ma, et al. Isdnet: Integrating shallow and deep networks for efficient ultra-high resolution segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4361–4370, 2022. 2

[3] 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. 2

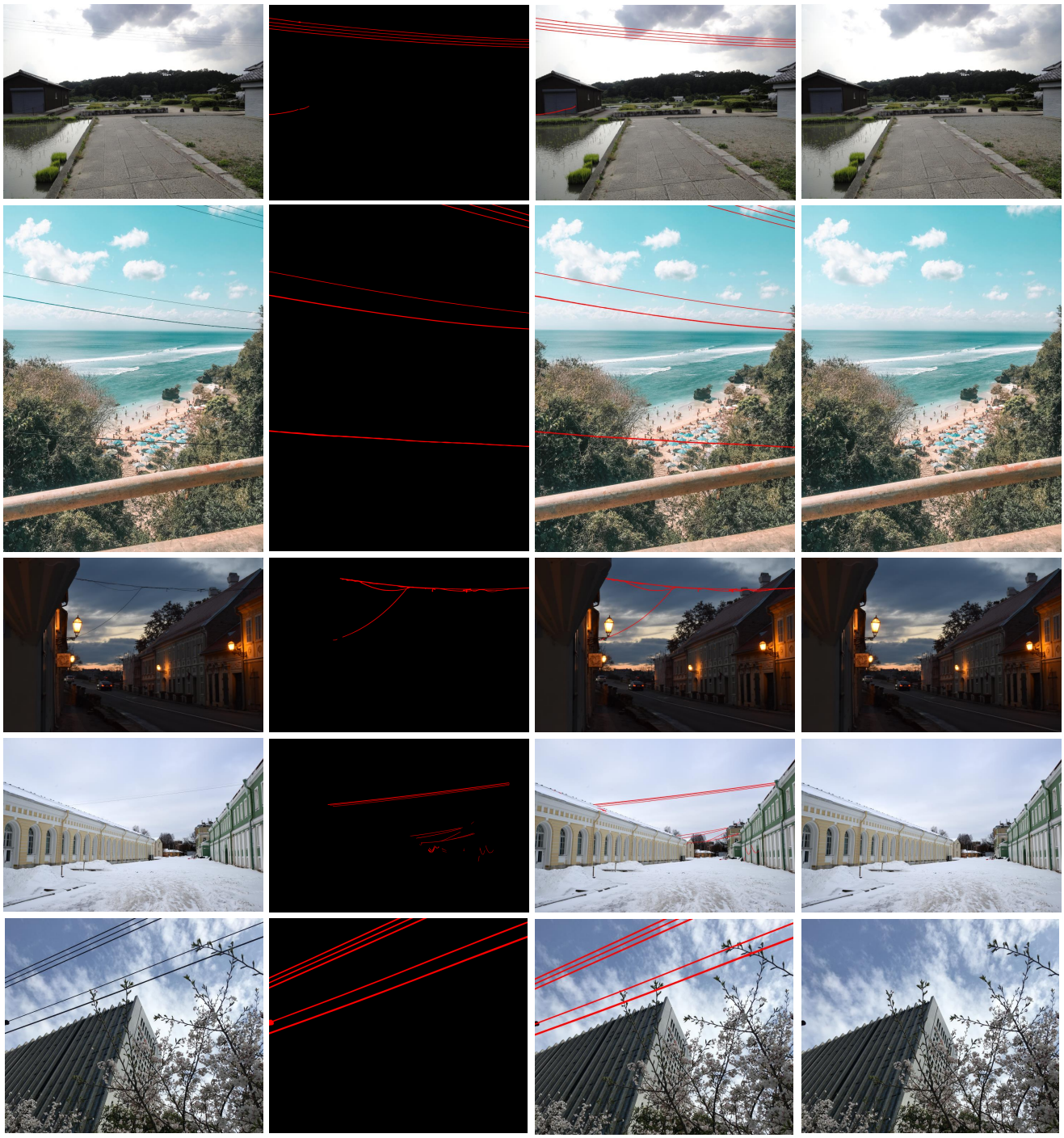


Figure 4. **Segmentation and inpainting visualizations**. Our model can handle several challenging scenes, including strongly backlit (top row), background with complex texture (2nd row), low light (3rd row), and barely visible wires (4th row)