

# Automatic High Resolution Wire Segmentation and Removal

Mang Tik Chiu<sup>1,2</sup>, Xuaner Zhang<sup>2</sup>, Zijun Wei<sup>2</sup>, Yuqian Zhou<sup>2</sup>, Eli Shechtman<sup>2</sup>, Connally Barnes<sup>2</sup>, Zhe Lin<sup>2</sup>, Florian Kainz<sup>2</sup>, Sohrab Amirghodsi<sup>2</sup>, Humphrey Shi<sup>1,3</sup>

<sup>1</sup>UIUC, <sup>2</sup>Adobe, <sup>3</sup>University of Oregon

<https://github.com/adobe-research/auto-wire-removal>



Figure 1. We present an automatic high-resolution wire segmentation and removal pipeline. Each triad shows the high-resolution input image, our automatic wire segmentation result masked in red, and our full-resolution wire removal result. The visual quality of these photographs is greatly improved with our fully-automated wire clean-up system.

## Abstract

Wires and powerlines are common visual distractions that often undermine the aesthetics of photographs. The manual process of precisely segmenting and removing them is extremely tedious and may take up hours, especially on high-resolution photos where wires may span the entire space. In this paper, we present an automatic wire clean-up system that eases the process of wire segmentation and removal/inpainting to within a few seconds. We observe several unique challenges: wires are thin, lengthy, and sparse. These are rare properties of subjects that common segmentation tasks cannot handle, especially in high-resolution images. We thus propose a two-stage method that leverages both global and local contexts to accurately segment wires in high-resolution images efficiently, and a tile-based inpainting strategy to remove the wires given our predicted segmentation masks. We also introduce the first wire segmentation benchmark dataset, *WireSegHR*. Finally, we demonstrate quantitatively and qualitatively that our wire clean-up system enables fully automated wire removal with great generalization to various wire appearances.

## 1. Introduction

Oftentimes wire-like objects such as powerlines and cables can cross the full width of an image and ruin an otherwise beautiful composition. Removing these “distractors” is thus an essential step in photo retouching to improve the visual quality of a photograph. Conventionally, removing a wire-like object requires two steps: 1) segmenting out the wire-like object, and 2) removing the selected wire and inpainting with plausible contents. Both steps, if done manually, are extremely tedious and error-prone, especially for high-resolution photographs that may take photographers up to hours to reach a high-quality retouching result.

In this paper, we explore a fully-automated wire segmentation and inpainting solution for wire-like object segmentation and removal with tailored model architecture and data processing. For simplicity, we use *wire* to refer to all wire-like objects, including powerlines, cables, supporting/connecting wires, and objects with wire-like shapes.

Wire semantic segmentation has a seemingly similar problem setup with generic semantic segmentation tasks; they both take in a high-resolution image and generate dense predictions at a pixel level. However, wire semantic

segmentation bears a number of unique challenges. First, wires are commonly long and thin, oftentimes spanning the entire image yet having a diameter of only a handful of pixels. A few examples are shown in Figure 2. This prevents us from getting a precise mask based on regions of interest. Second, the input images can have arbitrarily high resolution up to  $10k \times 10k$  pixels for photographic retouching applications. Downsampling such high-resolution images can easily cause the thin wire structures to disappear. This poses a trade-off between preserving image size for inference quality and run-time efficiency. Third, while wires have regular parabolic shapes, they are often partially occluded and can reappear at arbitrary image location, thus not continuous. (e.g. [20, 36]).

To account for these challenges, we propose a system for automatic wire semantic segmentation and removal. For segmentation, we design a two-stage coarse-to-fine model that leverages both pixel-level details in local patches and global semantics from the full image content, and runs efficiently at inference time. For inpainting, we adopt an efficient network architecture [35], which enables us to use a tile-based approach to handle arbitrary high resolution. We design a training strategy to enforce color consistency between the inpainted region and the original image. We also present the first benchmark dataset, WireSegHR, for wire semantic segmentation tasks, where we collect and annotate high-resolution images with diverse scene contents and wire appearances. We provide analyses and baseline comparisons to justify our design choices, which include data collection, augmentation, and our two-stage model design. Together, these design choices help us overcome the unique challenges of accurately segmenting wires. Our contributions are as follows:

- **Wire segmentation model:** We propose a two-stage model for wire semantic segmentation that leverages global context and local information to predict accurate wire masks at high resolution. We design an inference pipeline that can efficiently handle ultra-high resolution images.
- **Wire inpainting strategy:** We design a tile-based inpainting strategy and tailor the inpainting method for our wire removal task given our segmentation results.
- **WireSegHR, a benchmark dataset:** We collect a wire segmentation benchmark dataset that consists of high resolution images, with diversity in wire shapes and scene contents. We also release the manual annotations that have been carefully curated to serve as ground truths. Besides, we also propose a benchmark dataset to evaluate inpainting quality.

## 2. Related Work

**Semantic segmentation** Semantic segmentation has been actively researched over the past decade. For example, the

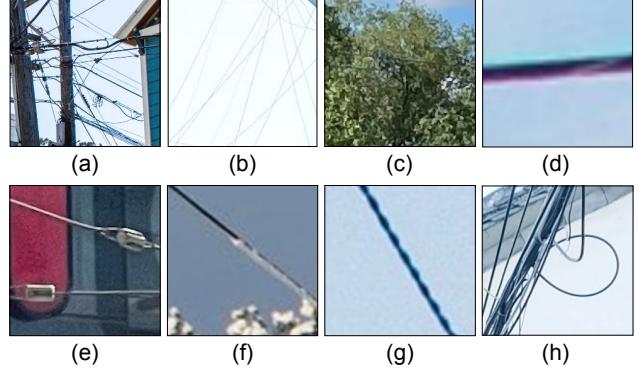


Figure 2. **Challenges of wire segmentation.** Wires have a diverse set of appearances. Challenges include but are not limited to (a) structural complexity, (b) visibility and thickness, (c) partial occlusion by other objects, (d) camera aberration artifacts, and variations in (e) object attachment, (f) color, (g) width and (h) shape.

DeepLab series [4–6] has been one of the most widely used set of semantic segmentation methods. They leverage dilated convolutions to capture long-range pixel correlations. Similarly, CCNet [14] attend to non-local regions via a two-step axial attention mechanism. PSPNet [48] use multi-scale pooling to extract high-resolution features.

Recently, the self-attention mechanism [37] has gained increasing popularity. Transformer-based models for semantic segmentation [11, 12, 17, 18, 26, 31, 39, 51] significantly outperform convolution-based networks since the attention modules benefit from their global receptive fields [39], which let the models attend to objects that span across larger portions of the feature map.

While these above methods work well in common object semantic segmentation, when applied to our task of wire segmentation in high-resolution images, they either drop significantly in segmentation quality or require long inference times. We show in Section 6 that directly applying these methods to our task yields undesirable results.

**High-resolution image segmentation** Segmentation in high-resolution images involves additional design considerations. It is computationally infeasible to perform inference on the full-resolution image with a deep network. As a result, to maximally preserve image details within the available computation resources, many methods employ a global-local inference pipeline. For instance, GLNet [7] simultaneously predict a coarse segmentation map on the downsampled image and a fine segmentation map on local patches at the original resolution, then fuse them to produce the final prediction. MagNet [15] is a recent method that proposes to iteratively predict and refine coarse segmentation maps at multiple scales using a single feature extractor and multiple lightweight refinement modules. CascadePSP [8] train a standalone class-agnostic model to refine predictions at a higher resolution from a pretrained segmentation model. ISDNet [10] propose to use an extremely

lightweight subnetwork to take in the entire full-resolution image. However, the subnetwork is limited in capacity and thus segmentation quality. We share the same idea with these past works on using a coarse-to-fine approach for wire segmentation, but modify the architecture and data processing to tailor to wires.

**Wire/Curve segmentation** While few works tackle wire segmentation in high-resolution images, there are prior works that handle similar objects. For example, Transmission Line Detection (TLD) is an actively researched area in aerial imagery for drone applications. Convolutional neural networks are used [2, 23, 28, 46] to segment overhanging power cables in outdoor scenes. However, wire patterns in TLD datasets are relatively consistent in appearance and shape – evenly spaced and only spanning locally. In contrast, we handle more generic wires seen in regular photographic contents, where the wire appearance has much higher variety.

Some other topics are loosely related to our task. Lane detection [20, 34, 36] aims to segment lanes for autonomous driving applications. These methods benefit from simple line parameterization (e.g., as two end-points), and strong positional priors. In contrast, as shown in Figure 2, wires vary drastically in shapes and sizes in our task, thus making them difficult to parameterize.

**High-Resolution Image Inpainting** Image inpainting has been well-explored using patch synthesis-based methods [3, 9, 22, 38] or deep neural networks [16, 25, 29, 40, 43, 44]. Zhao *et al.* leveraged the powerful image synthesis ability of StyleGAN2 [21] and proposed CoModGAN [49] to push the image generation quality to a newer level, and was followed by [19, 50]. Most of these deep models cannot be applied to inpainting tasks at high-resolution images. The latest diffusion-based inpainting model like DALLE-2 [30], LDM [32], and StableDiffusion etc. also suffer from long inference time and low output resolution. ProFill [45] was first proposed to address high resolution inpainting via a guided super resolution module. HiFill [42] utilized a contextual residual aggregation module and the resolution can be up to 8K. LaMa [35] applied the fourier convolutional residual blocks to make the propagation of image structures well. LaMa was trained on only  $256 \times 256$  images, but can be used for images up to 2K with high quality. Recently, Zhang *et al.* [47] proposed to use guided PatchMatch for any-resolution inpainting and extended the deep inpainting results from LaMa to modern camera resolution. The textures are better reused, while the structure and line completion at high-resolution can still be challenging. In this paper, we aim at removing wires from high resolution photos. The problem can become easier if we run inpainting in a local manner since wires are usually thin and long. Therefore, we propose to revisit LaMa for wire removal, and run the inference in a tile-based fashion.

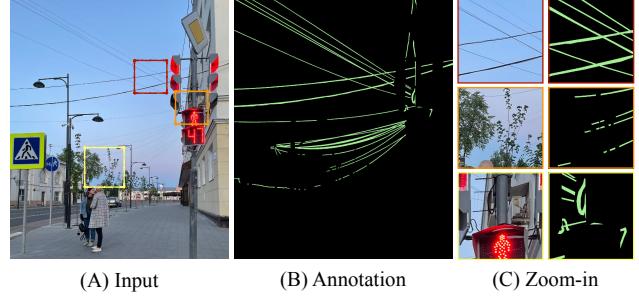


Figure 3. **Wire Annotation Example.** An example wire annotation in our dataset. Our annotation (B) is accurate in different wire thicknesses (red), variations in wire shapes (orange) and accurate wire occlusions (yellow).

### 3. Dataset Collection and WireSegHR

#### 3.1. Image Source and Annotations

Our definition of wires include electrical wires/cables, power lines, supporting/connecting wires, and any wire-like object that resemble a wire structure. We collect high-resolution images with wires from two sources: 80% of the images are from photo sharing platforms (Flickr, Pixabay, etc.), and 20% of the images are captured with different cameras (DSLRs and smartphones) in multiple countries on our own. For the internet images, we collect 400K candidate images by keyword-searching. Then, we remove duplicates and images where wires are the only subjects. We then curate the final 6K images that cover sufficient scene diversity like city, street, rural area and landscape.

Our wire annotation process contains two rounds. In the first round, annotators draw detailed masks over wires at full-resolution. The annotated masks enclose the main wire body and the boundary, oftentimes including a gradient falloff due to aliasing or defocus. The boundary region annotation is crucial so as to avoid residual artifacts during wire removal. In the second round, quality assurance is carried out to re-annotate unsatisfactory annotations. We show an example of our high-quality wire annotations in Figure 3.

#### 3.2. Dataset Statistics

In Table 1, we list the statistics of our dataset and compare them with existing wire-like datasets. Our dataset is the first wire dataset that contains high-resolution photographic images. The dataset is randomly split into 5000 training, 500 validation, and 500 testing images. We release 420 copyright-free test images with annotations.

### 4. High-Resolution Wire Segmentation

Wires appear visually different from common objects – being thin, long, sparse and oftentimes partially occluded. We find the following two design choices crucial to building an effective wire segmentation system: 1) having a two

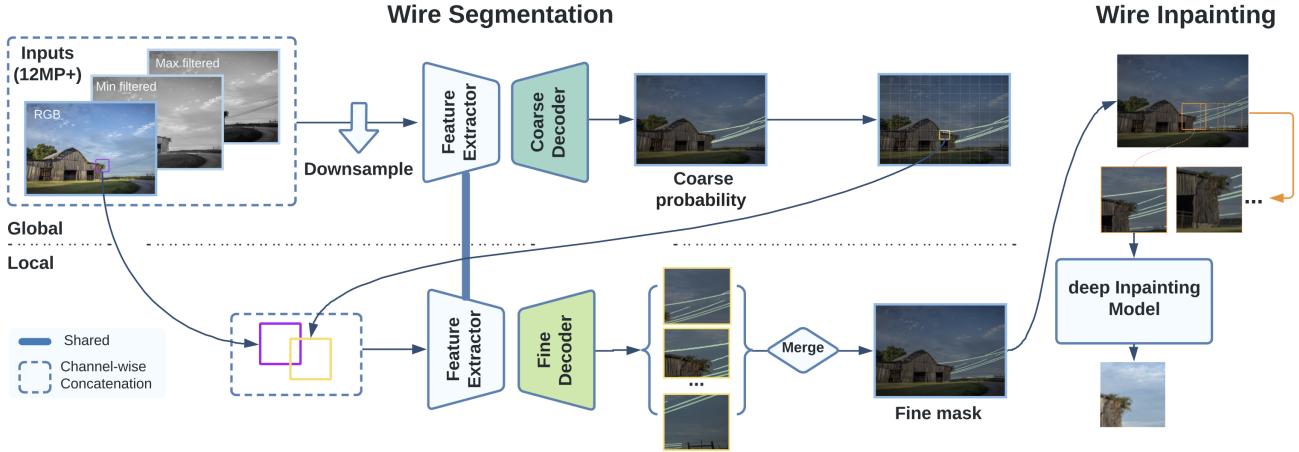


Figure 4. **Our wire removal system.** A system overview of our wire segmentation and removal for high resolution images. Input is concatenated with min- and max-filtered luminance channels. The downsampled input is fed into the coarse module to obtain the global probability. In the local stage, original-resolution patches are concatenated with the global probability map to obtain the local logit map. After a segmentation mask is predicted, we adopt LaMa architecture and use a tile-based approach to achieve wire removal. See Section 4, 5 for details.

Dataset	# Wire Images	Min. Image Size	Max. Image Size	Median Image Size
Powerline [41]	2000	128×128	128×128	128×128
PLDU [46]	573	540×360	540×360	540×360
PLDM [46]	287	540×360	540×360	540×360
TTPLA [2]	1100	3840×2160	3840×2160	3840×2160
<b>Ours</b>	6000	360×240	15904×10608	5040×3360

Table 1. Statistics of our wire dataset compared to others.

stage framework so that coarse prediction from global context guides precise segmentation from local patches and 2) maximally preserving and augmenting image features and annotations of wires throughout the pipeline.

#### 4.1. The Two-stage Coarse to Fine Model

Figure 4 shows the two-stage segmentation pipeline. It consists of a coarse and a fine module, which share an encoder  $E$  and have their own decoder  $D_C$  and  $D_F$ . Intuitively, the coarse module aims to capture the global contextual information from the entire image and highlight the image regions possibly containing wires. Conditioned on the predictions from the coarse module, the fine module achieves high-resolution wire segmentation by only looking at local patches likely containing wires.

Given a high-resolution image  $I_{\text{glo}}$ , we first bilinearly downsample it to  $I_{\text{glo}}^{ds}$  with a fixed size  $p \times p$  and feed it into the coarse module. The module predicts the global probability map  $P_{\text{glo}} = \text{SoftMax}(D_C(E(I_{\text{glo}}^{ds})))$  containing the activation of the wire regions.

For each patch  $I_{\text{loc}}$  of size  $p \times p$  cropped from the full-resolution image  $I_{\text{glo}}$ , and the corresponding conditional probability map  $P_{\text{con}}$  cropped from  $P_{\text{glo}}$ , we predict the lo-

cal probability  $P_{\text{loc}} = \text{SoftMax}(D_F(E(I_{\text{loc}}, P_{\text{con}})))$ . Note that  $E$  is shared between the coarse and the fine module, thus it should take inputs with the same number of channels. Therefore, for the coarse module, we concatenate an additional zero channel with the input image to make the channel number consistent.

We apply Cross Entropy (CE) loss to both the global  $P_{\text{glo}}$  and local probability map  $P_{\text{loc}}$ , comparing with their ground truth annotations  $G_{\text{glo}}$  and  $G_{\text{loc}}$ .

$$\begin{aligned}\mathcal{L}_{\text{glo}} &= CE(P_{\text{glo}}, G_{\text{glo}}) \\ \mathcal{L}_{\text{loc}} &= CE(P_{\text{loc}}, G_{\text{loc}})\end{aligned}\quad (1)$$

The final loss  $\mathcal{L}$  is the sum of the two:

$$\mathcal{L} = \mathcal{L}_{\text{glo}} + \lambda \mathcal{L}_{\text{loc}}, \quad (2)$$

where we set  $\lambda = 1$  for training. Similar to Focal loss [24] and Online Hard Example Mining [33], we balance the wire and background samples in the training set by selecting patches that contain at least 1% of wire pixels.

To perform inference, we first feed the downsampled image to the coarse module, which is the same as training. Local inference is done by running a sliding window over the entire image, where the patch is sampled only when there is at least some percentage of wire pixels (determined by  $\alpha$ ). This brings two advantages: First, we save computation time in regions where there are no wires. Second, the local fine module can leverage the information from the global branch for better inference quality.

#### 4.2. Wire Feature Preservation

As wires are thin and sparse, applying downsampling to the input images may make the wire features vanish entirely.

To mitigate this challenge, we propose a simple feature augmentation technique by taking the min and max pixel luminance values of the input image over a local window. Either the local min or the max value makes the wire pixels more visually apparent. In practice, we concatenate the min- and max-filtered luminance channels to the RGB image and condition map, resulting in 6 total channels as input. We name this component MinMax.

Besides feature augmentations, we also adapt the architecture to maximally preserve the sparse wire annotations. We propose to use “overprediction” and achieve this by using max-pool downsampling on the coarse labels during training, which preserves activation throughout the coarse branch. We name this component MaxPool. We provide ablation studies for these components in Section 6.

## 5. High-Resolution Wire Inpainting

Given a full-resolution wire segmentation mask estimated by our wire segmentation model, we propose an inpainting pipeline to remove and fill in the wire regions. Our approach addresses two major challenges in wire inpainting. First, recent state-of-the-art deep inpainting methods do not handle arbitrary resolution images, which is critical for high-resolution wire removal. Second, deep inpainting methods often suffer from color inconsistency when the background has uniform (or slowly varying) colors. This issue is particularly significant for wires, as they are often in front of uniform backgrounds, such as the sky or building facades. The commonly used reconstruction loss, such as L1, is not sensitive to color inconsistency, which further exacerbates this issue.

We thus revisit the efficient deep inpainting method LaMa [35]. Compared with other inpainting models, LaMa has two major advantages. First, it contains the Fourier convolutional layers which enables an efficient and high-quality structural completion. This helps complete building facades and other man-made structures with fewer artifacts. Second, its high inference efficiency makes a tile-based inference approach possible for high resolution images.

To address color inconsistency, we propose a novel “onion-peel” color adjustment module. Specifically, we compute the mean of the RGB channels within the onion-peel regions  $M_o = D(M, d) - M$  of the wire mask  $M$ , where  $D$  is the binary dilation operator, and  $d$  is the kernel size. The color difference for each channel  $c \in R, G, B$  becomes  $\text{Bias}_c = \mathbb{E}[M_o(x_c - y_c)]$ , where  $x$  is the input image, and  $y$  is the output from the inpainting network. The final output of the inpainting model is:  $\hat{y}_c = y_c + \text{Bias}_c$ . Loss functions are then applied to  $\hat{y}_c$  to achieve color consistency while compositing the final result  $y_{out} = (1 - M) \odot x + M \odot \hat{y}$ .

## 6. Experiments

### 6.1. Implementation Details

**Wire Segmentation Network.** We experiment with ResNet-50 [13] and MixTransformer-B2 [39] as our shared feature extractor. We expand the input RGB channel to six channels by concatenating the conditional probability map, min- and max-filtered luminance channels. For the min and max filtering, we use a fixed 6x6 kernel. We use separate decoders for the coarse and fine modules, denoted as  $D_C$  and  $D_F$  respectively.

We use the MLP decoder proposed in [39] for the Mix-Transformer segmentation model, and the ASPP decoder in [6] for our ResNet-50 segmentation model. In both the segmentation and inpainting modules, we take the per-pixel average of the predicted probability when merging overlapping patches. To crop  $P_{con}$  from  $P_{glo}$ , we upsample the predicted  $P_{glo}$  to the original resolution, then crop the predicted regions according to the sliding window position.

To train the segmentation module, we downsample the image  $I_{glo}$  to  $p \times p$  to obtain  $I_{glo}^{ds}$ . From  $I_{glo}$ , we randomly crop one  $p \times p$  patch  $I_{loc}$  that contains at least 1% wire pixels. This gives a pair of  $I_{glo}^{ds}$  and  $I_{loc}$  to compute the losses. During inference,  $I_{glo}^{ds}$  is obtained in the same way as training, while multiple  $I_{loc}$  are obtained via a sliding window sampled only when the proportion of wire pixels is above  $\alpha$ . All feature extractors are pretrained on ImageNet.

We train our model on 5000 training images. The model is trained for 80k iterations with a batch size of 4. We set patch size  $p = 512$  during training. For all ResNet models, we use SGD with a learning rate of 0.01, a momentum of 0.9 and weight decay of 0.0005. For MixTransformer models, we use AdamW [27] with a learning rate of 0.0002 and weight decay of 0.0001. Our training follows the “poly” learning rate schedule with a power of 0.9. During inference, we set both the global image size and local patch size  $p$  to 1024. Unless otherwise specified, we set the percentage for local refinement to 1% ( $\alpha = 0.01$ ).

**Wire Inpainting Network.** We adopt LaMa [35] for wire inpainting by finetuning on an augmented wire dataset. To prepare the wire training set, we randomly crop ten  $680 \times 680$  patches from the non-wire regions of each image in our training partition. In total, we have 50K more training images in addition to the 8M Places2 [52] dataset, and increase its sampling rate by  $10\times$  to balance the dataset. We also use all the ground truth segmentation maps in our training set to sample wire-like masks. During training, we start from Big-LaMa weights, and train the model on  $512 \times 512$  patches. We also prepare a synthetic wire inpainting quality evaluation dataset, containing 1000 images at  $512 \times 512$  with synthetic wire masks. While running inference on full-resolution images, we apply a tile-based approach, by fixing the window size at  $512 \times 512$  with an 32-pixel overlap.

## 6.2. Wire Segmentation Evaluation

**Quantitative Evaluation** We compare with several widely-used object semantic segmentation and high-resolution semantic segmentation models. Specifically, we train DeepLabv3+ [6] with ResNet-50 [13] backbone under two settings: global and local. In the global setting, the original images are resized to  $1024 \times 1024$ . In the local setting, we randomly crop  $1024 \times 1024$  patches from the original images. We train our models on 4 Nvidia V100 GPUs and test them on a single V100 GPU. For high-resolution semantic segmentation models, we compare with CascadePSP [8], MagNet [15] and ISDNet [10]. We describe the training details of these works in the supplement.

We present the results of in Table 2 tested on WireSegHR. We report wire IoU, F1-score, precision and recall for quantitative evaluation. We also report wire IoUs for images at three scales, small ( $0 - 3000 \times 3000$ ), medium ( $3000 \times 3000 - 6000 \times 6000$ ) and large ( $6000 \times 6000+$ ), which are useful for analyzing model characteristics. Finally, we report average, minimum and maximum inference times on WireSegHR.

As shown in Table 2, while the global model runs fast, it has lower wire IoUs. In contrast, the local model produces high-quality predictions, but requires a very long inference time. Meanwhile, although CascadePSP is a class-agnostic refinement model designed for high-resolution segmentation refinement, it primarily targets common objects and does not generalize to wires. For MagNet, its refinement module only takes in probability maps without image features, thus failing to refine when the input prediction is inaccurate. Among these works, ISDNet is relatively effective and efficient at wire prediction. However, their shallow network design trades off capacity for efficiency, limiting the performance of wire segmentation that is thin and sparse.

Compared to the methods above, our model achieves the best trade-off between accuracy and memory consumption. By leveraging the fact that wires are sparse and thin, our pipeline captures both global and local features more efficiently, thus saving a lot of computation while maintaining high segmentation quality.

**Qualitative Evaluation** We provide visual comparisons of segmentation models in Figure 6. We show the “local” DeepLabv3+ model as it consistently outperforms its “global” variant given that “local” predicts wire masks in a sliding-window manner at the original image resolution. As a trade-off, without global context, the model suffers from over-prediction. CascadePSP is designed to refine common object masks given a coarse input mask, thus fails to produce satisfactory results when the input is inaccurate or incomplete. Similarly, the refinement module of MagNet does not handle inaccurate wire predictions. ISDNet performs the best among related methods, but the quality is still unsatisfactory as it uses a lightweight model with limited ca-

pacity. Compared to all these methods, our model captures both global context and local details, thus producing more accurate mask predictions.

**Ablation Studies** In Table 3, we report wire IoUs after removing each component in our model, including Min-Max, MaxPool, and Coarse condition concatenation. We find that all components play a significant role for accurate wire prediction, particularly in large images. Both MinMax and MaxPool are effective in encouraging prediction, which is shown by the drop in recall without either component, also shown in Figure 6. Coarse condition, as described in Section 4, is crucial in providing global context to the local network, without which the wire IoU drops significantly.

Table 4 shows the wire IoUs and inference speed of our two-stage model as  $\alpha$  changes. We observe a consistent decrease in performance as  $\alpha$  increases. On the other hand, setting  $\alpha$  to 0.01 barely decreases IoU, while significantly boosting inference speed, which means the coarse network is effectively activated at wire regions.

## 6.3. Wire Inpainting Evaluation

We evaluate our wire inpainting model using the synthetic dataset. Results are shown in Table 5. Our model structure is highly related to LaMa [35]. The difference is the training data and the proposed color adjustment module to address color inconsistency. We also compare our methods with PatchMatch [3] based on patch synthesis, DeepFillv2 [44] based on Contextual Attention, CMGAN [50] and FcF [19] based on StyleGAN2 [21] and LDM [32] based on Diffusion. Inference speed is measured on a single A100-80G GPU. Visual results on synthetic and real images are shown in Figure 7. PatchMatch, as a traditional patch synthesis method, produces consistent color and texture that leads to high PSNR. However, it performs severely worse on complicated structural completion. StyleGAN-based CMGAN and FcF are both too heavy for wires that are thin and sparse. Besides, diffusion-based models like LDM tends to generate arbitrary objects and patterns. DeepFill and the official Big-LaMa both have severe color inconsistency issue, especially in the sky region. Our model has a balanced quality and efficiency, and performs well on structural completion and color consistency. Note that we use a tile-based method at inference time. The reason the tile-based strategy can be employed is due to the wire characteristics: sparse, thin and lengthy. More high-resolution inpainting results are in the supplementary materials.

## 7. Discussion

### 7.1. Comparison with Google Pixel 6

Recently, Google Pixel 6 [1] announced the “Magic Eraser” photo feature that automatically detects and removes distractors. Note that this is a product feature and is not specifically designed for wires, and thus is hardly

Model	Wire IoU	F1	Precision	Recall	IoU (Small)	IoU (Medium)	IoU (Large)	Avg. Time (s/img)	Min. Time (s/img)	Max. Time (s/img)
DeepLabv3+ (Global) [6]	37.77	54.83	69.68	45.20	51.62	38.89	31.89	0.22	0.07	0.78
DeepLabv3+ (Local) [6]	48.66	65.46	68.13	63.0	60.23	51.44	40.17	3.27	0.05	16.59
CascadePSP (Pretrained) [8]	20.44	33.94	62.19	23.34	33.64	21.80	13.78	2.32	0.37	36.79
CascadePSP (Retrained) [8]	26.85	42.33	52.44	35.49	48.22	28.97	15.80	2.25	0.37	25.37
MagNet [15]	33.71	50.42	87.69	35.38	43.59	32.67	34.48	3.89	0.54	17.97
MagNet-Fast [15]	37.87	54.94	67.98	46.09	46.75	35.88	41.42	1.36	0.55	5.33
ISDNet (R-18) [10]	46.52	63.50	77.56	53.75	55.09	47.15	43.34	0.29	0.12	0.86
ISDNet (MiT-b2) [10]	47.90	64.77	77.38	55.70	54.48	46.77	49.51	0.26	0.13	1.02
Ours (R-50)	47.75	64.64	74.86	56.87	60.68	50.19	38.19	1.24	0.13	4.67
Ours (MiT-b2)	60.83	75.65	83.62	69.06	63.52	59.83	62.93	0.82	0.07	3.36

Table 2. Performances of common semantic segmentation and recent high-resolution semantic segmentation models on our dataset. We find that our dataset poses many challenges that high-resolution segmentation models fail to tackle effectively.

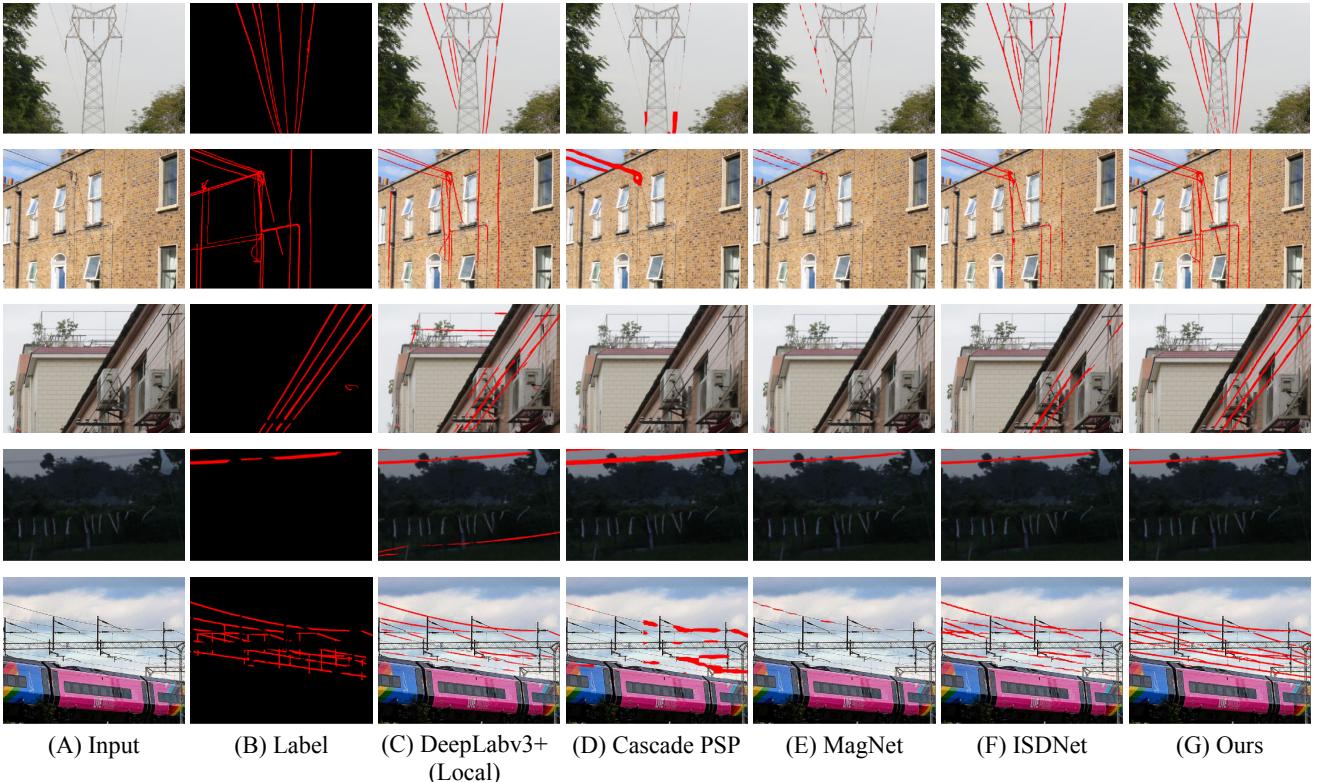


Figure 5. **Qualitative comparison of several semantic segmentation models.** A common object semantic segmentation model (DeepLabv3+) either fails to find thin wires or overpredicts due to lack of global context. On the other hand, CascadePSP and MagNet, being refinement-based models, cannot work well on wires when the predictions are inaccurate or missing. While ISDNet can capture many thin wires regions, it cannot produce a high-quality prediction. In contrast, our model is able to both capture accurate wire regions and produce fine wire masks, and maintain low inference time.

comparable with our method. We compare against this feature by uploading the images to Google Photos and applying “Magic Eraser” without manual intervention. We find that “Magic Eraser” performs well on wires with clear background, but it suffers from thin wires that are hardly visible and wires with complicated background. We show two examples in the supplementary material.

## 7.2. Failure cases

While our proposed wire segmentation model produces high-quality masks in most situations, there are still some challenging cases that our model cannot resolve. In particular, wires that are heavily blended in with surrounding structures/background, or wires under extreme lighting conditions are challenging to segment accurately. We show several examples in the supplementary material.

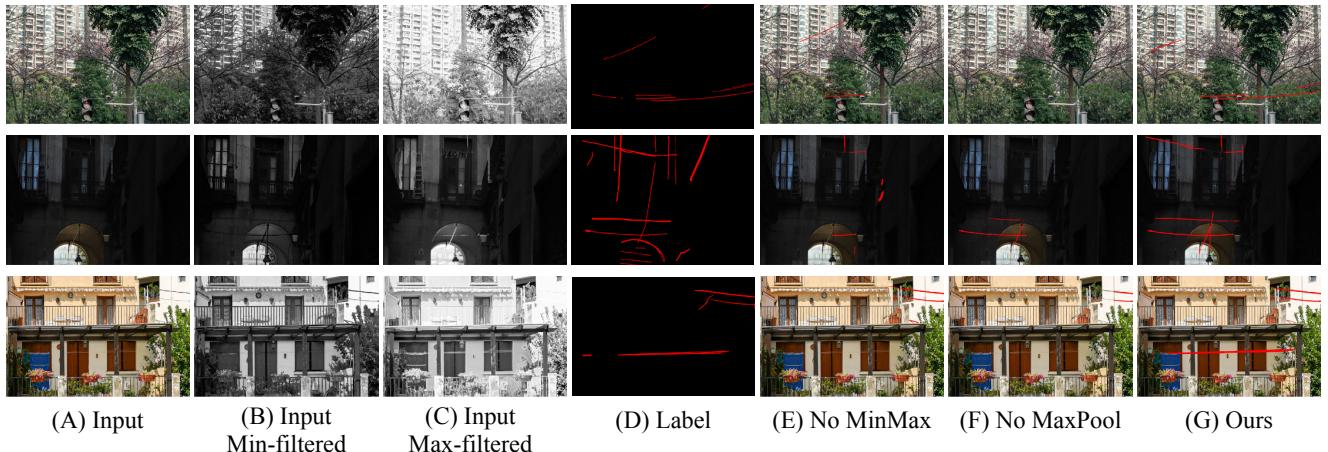


Figure 6. **Qualitative comparison of our model components.** MinMax enhances wire image features when they are too subtle to see in RGB, while MaxPool encourages aggressive predictions in the coarse branch. Both components enable the model to pick up more regions for the final wire mask prediction.

Model	Wire IoU	F1	Precision	Recall	IoU (Small)	IoU (Medium)	IoU (Large)
Ours	60.83	75.65	83.62	69.06	63.52	59.83	62.93
- MinMax	60.01	75.01	84.87	67.2	63.67	58.99	61.97
- MaxPool	59.86	74.89	85.25	66.78	61.45	59.40	60.76
- Coarse	56.92	72.55	82.91	64.49	62.83	57.42	54.47

Table 3. Ablation study of our model components.

$\alpha$	Wire IoU	F1	Precision	Recall	Avg. Time (s/img)	Speed up
0.0	60.97	75.75	82.63	69.93	1.91	1 $\times$
0.01	60.83	75.65	83.62	69.06	0.82	2.3 $\times$
0.02	60.35	75.27	83.97	68.20	0.75	2.5 $\times$
0.05	55.17	71.11	84.84	61.20	0.58	3.3 $\times$
0.1	42.44	59.59	86.06	45.57	0.4	4.8 $\times$

Table 4. Ablation on the threshold for refinement. At  $\alpha = 0.0$ , all windows are passed to the fine module.

Model	PSNR $\uparrow$	LPIPS $\downarrow$	FID $\downarrow$	Speed (s/img)
PatchMatch [3]	50.29	0.0294	5.0403	-
DeepFillv2 [44]	47.01	0.0374	8.0086	0.009
CMGAN [50]	50.07	0.0255	3.8286	0.141
FeF [19]	48.82	0.0322	4.7848	0.048
LDM [32]	45.96	0.0401	10.1687	4.280
Big-LaMa [35]	49.63	0.0267	4.1245	0.034
Ours (LaMa-Wire)	50.06	0.0259	3.6950	0.034

Table 5. Quantitative results of inpainting on our synthetic wire inpainting evaluation dataset (1000 images). Our model achieves the highest perceptual quality in terms of FID, and has a balanced speed and quality.

## 8. Conclusion

In this paper, we propose a fully automated wire segmentation and removal system for high-resolution imagery. We demonstrate a segmentation method that maximally preserves sparse wire features and annotations, with a two-

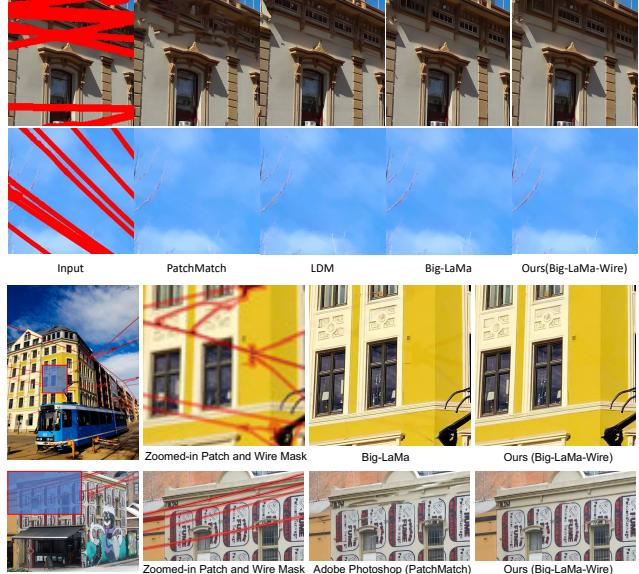


Figure 7. **Inpainting Comparison.** Our model performs well on complicated structure completion and color consistency, especially on building facades and sky regions containing plain and uniform color.

stage model that effectively uses global context and local details. The predicted segmentation mask is used in our tile-based wire inpainting model that has been demonstrated to produce seamless inpainting results. We also introduce WireSegHR, the first benchmark wire dataset with high-quality annotations. We hope our proposed method will provide insights into tackling semantic segmentation with high resolution image and annotation properties, and that our benchmark dataset encourage further research in wire segmentation and removal.

## References

- [1] Pixel 6, a smarter chip for a smarter phone - google store. [https://store.google.com/product/pixel\\_6?hl=en-US](https://store.google.com/product/pixel_6?hl=en-US). (Accessed on 11/14/2021). 6
- [2] Rabab Abdelfattah, Xiaofeng Wang, and Song Wang. Ttpla: An aerial-image dataset for detection and segmentation of transmission towers and power lines. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3, 4
- [3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 3, 6, 8
- [4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2
- [5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2
- [6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2, 5, 6, 7
- [7] 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. 2
- [8] 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, 6, 7
- [9] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics (TOG)*, 31(4):1–10, 2012. 3
- [10] 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, 6, 7
- [11] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. 2022. 2
- [12] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. 2022. 2
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 6
- [14] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. 2
- [15] 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, 6, 7
- [16] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017. 3
- [17] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. 2023. 2
- [18] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masking transformer backbones for effective semantic segmentation. *arXiv*, 2021. 2
- [19] Jitesh Jain, Yuqian Zhou, Ning Yu, and Humphrey Shi. Keys to better image inpainting: Structure and texture go hand in hand. *arXiv preprint arXiv:2208.03382*, 2022. 3, 6, 8
- [20] Oshada Jayasinghe, Damith Anhettigama, Sahan Hemachandra, Shenali Kariyawasam, Ranga Rodrigo, and Peshala Jayasekara. Swiftlane: Towards fast and efficient lane detection. *arXiv preprint arXiv:2110.11779*, 2021. 2, 3
- [21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 3, 6
- [22] Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self tuning texture optimization. In *Computer Graphics Forum*, volume 34, pages 349–359. Wiley Online Library, 2015. 3
- [23] Bo Li, Cheng Chen, Shiwen Dong, and Junfeng Qiao. Transmission line detection in aerial images: An instance segmentation approach based on multitask neural networks. *Signal Processing: Image Communication*, 96:116278, 2021. 3
- [24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4
- [25] Guilin Liu, Fitzsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018. 3
- [26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 2
- [27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

- [28] Van Nhan Nguyen, Robert Jenssen, and Davide Roverso. Ls-net: Fast single-shot line-segment detector. *arXiv preprint arXiv:1912.09532*, 2019. 3
- [29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 3
- [30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3
- [31] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 2
- [32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 3, 6, 8
- [33] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 4
- [34] Jinming Su, Chao Chen, Ke Zhang, Junfeng Luo, Xiaoming Wei, and Xiaolin Wei. Structure guided lane detection. *arXiv preprint arXiv:2105.05403*, 2021. 3
- [35] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022. 2, 3, 5, 6, 8
- [36] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 294–302, 2021. 2, 3
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2
- [38] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence*, 29(3):463–476, 2007. 3
- [39] Enze Xie, Wenhui Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 2, 5
- [40] Xingqian Xu, Shant Navasardyan, Vahram Tadevosyan, Andranik Sargsyan, Yadong Mu, and Humphrey Shi. Image completion with heterogeneously filtered spectral hints. In *WACV*, 2023. 3
- [41] Ömer Emre Yetgin, Ömer Nezih Gerek, and Ömer Nezih. Power image dataset (infrared-ir and visible light-vl). *Mendeley Data*, 8, 2017. 4
- [42] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020. 3
- [43] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018. 3
- [44] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019. 3, 6, 8
- [45] Yu Zeng, Zhe Lin, Jimei Yang, Jianming Zhang, Eli Shechtman, and Huchuan Lu. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In *European conference on computer vision*, pages 1–17. Springer, 2020. 3
- [46] Heng Zhang, Wen Yang, Huai Yu, Haijian Zhang, and Gui-Song Xia. Detecting power lines in uav images with convolutional features and structured constraints. *Remote Sensing*, 11(11):1342, 2019. 3, 4
- [47] Lingzhi Zhang, Connelly Barnes, Kevin Wampler, Sohrab Amirghodsi, Eli Shechtman, Zhe Lin, and Jianbo Shi. Inpainting at modern camera resolution by guided patchmatch with auto-curation. In *European Conference on Computer Vision*, pages 51–67. Springer, 2022. 3
- [48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2
- [49] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021. 3
- [50] Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu, Sohrab Amirghodsi, and Jebo Luo. Cm-gan: Image inpainting with cascaded modulation gan and object-aware training. *arXiv preprint arXiv:2203.11947*, 2022. 3, 6, 8
- [51] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 2
- [52] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5