Supplemental Materials: Perceptual Artifacts Localization for Inpainting

Lingzhi Zhang[1], Yuqian Zhou[2], Connelly Barnes[2], Sohrab Amirghodsi[2], Zhe Lin[2], Eli Shechtman[2], and Jianbo Shi[1]

[1] University of Pennsylvania
[2] Adobe Research

1 Mask Generation

In this work, all of our images are uniformly sampled from Places2 dataset [17]. We generate two types of masks for our experiments, which are masks on the background region and masks covering a complete object. We discuss the details of how to generate these masks in below.

Masks on the Background. Since current inpainting models still can not understand the object-level prior and thus can not properly fill the object region, we do not want to sample masks that cover partial objects, which none of the current methods can properly deal with. To this end, we first use Mask R-CNN [3] to find all object masks, and then avoid sampled holes to partially overlap with these object regions. We use both free-form masks [12] and instance masks in our experiments, where the hole size ratio over the entire image ranges from 0.08 to 0.3. The instance masks are collected from multiple segmentation datasets, such as COCO [6] and Pascal VOC [2]. Some examples of these masks are shown in Fig. 1.

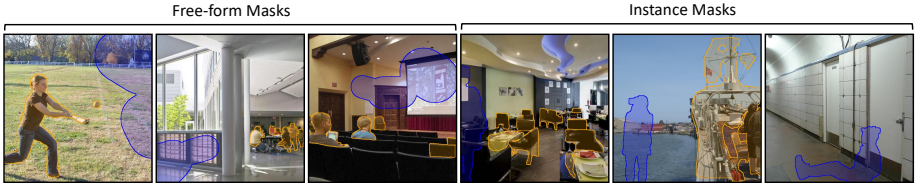

Fig. 1. Some visual examples of mask sampling for labeling and evaluation. The orange masks indicate the foreground region, where the hole masks avoid to overlap with.

Masks for Object Removal. In this work, we propose Perceptual Artifacts Ratio (PAR) metric to evaluate the inpainting quality for object removal scenario, which is often consider to be a hard but commonly used scenario for inpainting. To generate the hole mask at scale, we used Mask R-CNN [3] to first find all object masks in the image, then randomly select one object mask, and

Fig. 2. Some visual examples of mask sampling for object removal inpainting scenarios.

dilate the mask with 5×5 kernel for three iterations to increase coverage of the object. Some examples of these object removal masks are shown in Fig. 2.

For our perceptual artifacts labeling, we use a mix of both types of masks. Similarly, in our user study for evaluating "original fill vs. iterative fill" in section 6.2, half of cases use the masks sampled on the background and another half of the cases use the object removal masks. For our PAR metric study in section 5 in the main paper, we use only the masks covering complete objects, since we are evaluating the inpainting quality for object removal scenario.

2 Training Details of the Segmentation Network

In this section, we describe the training details of our perceptual artifacts segmentation network. For our final chosen model with ResNet-50 backbone [4] and PSP head [15], we also added an auxiliary FCN head [7], which has a loss ratio of 0.4 compared to the PSP head [15]. We train the segmentation network for 20,000 iterations using SGD optimizer with learning rate of 0.01, momentum of 0.9, and weight decay of 0.0005. The learning rate is schedule to decay in polynomial manner by power of 0.9, where the minimum is set to be 0.0001. For the data augmentation, we adopt random flip with a probability of 0.5 as well as JPEG compression. All of our models are trained on 4 NVIDIA RTX-Titan with a batch of 8 on each GPU. All images are trained at 512×512, which is the native resolution of inpainting outputs.

3 Perceptual Artifacts Labeling Interface

As discussed in the data labeling section in the main paper, we provide a copy of the filled image besides the image that workers actually mark on. The reason is that we want to provide workers a reference to know the original image content, which are useful to judge the perceptual artifacts region. Here, we show an actual worker's interface in Fig. 3.

4 User Study Interface

In this work, we have conducted two user studies using Amazon Mechanical Turk (AMT). In the first user study, we ask users to select the preferred image from

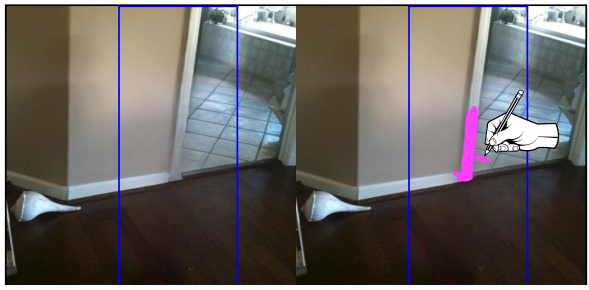

Fig. 3. User labeling interface.

two different inpainting methods, where the results are used for evaluating the metric correlation with human in section 5.3 in the main paper. In the second user study, we ask users to choose whether they think the iterative fill is better, same, or worse than the original fill, which are used for evaluation in section 6.2 in the main paper. Here, we show the user interfaces for the first and second studies in the left and right of Fig. 4, respectively. We intentionally do not show the original images to use, since we want to users to judge based on pure perceptual quality without any bias. We use bounding box to indicate the rough hole region instead of the actual hole boundary, so that users would have less bias on the boundary artifacts.

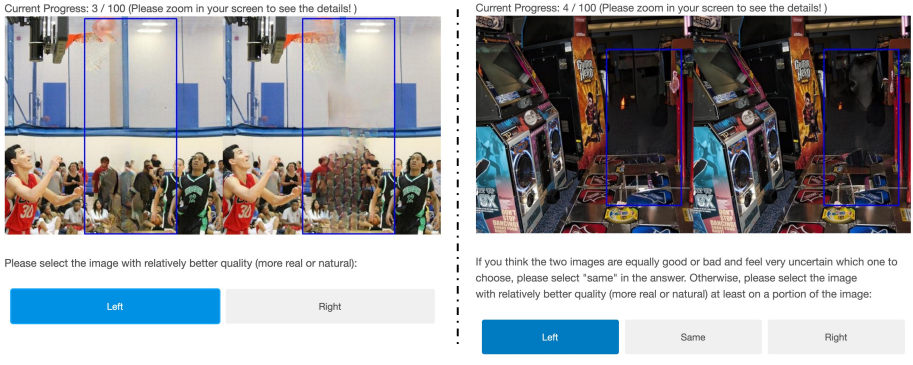

Fig. 4. Left: user study interface for evaluating between two inpainting models, which used for our PAR metric study in section 5.3. **Right**: user study interface for evaluating whether the iteratively filled images are better than the original fill, which are used as evaluation for section 6.2 in the main paper. Note that left/right images are randomized for each case.

5 More Iterations of Fill

In the main paper, we show how Perceptual Artifacts Ratio (PAR) consistently decreases over the fill iteration up to the 5^{th} iteration for 5,000 test images. However, we observe that methods like LaMa [10] still have slight tendency of decreasing PAR when increasing the fill iteration. Thus, we run a study by setting the number of iterative fill up to 20 for LaMa [11]. As shown in Table. 1, we observe that PAR still consistently decreases over the fill iteration, but the decreasing rate of PAR goes down to a very small number.

Iters.	PAR	Iters.	PAR	Iters.	PAR	Iters.	PAR
1	0.3786	6	0.1091	11	0.0707	16	0.0552
2	0.2439	7	0.0975	12	0.0666	17	0.0533
3	0.1811	8	0.0885	13	0.0628	18	0.0514
4	0.1464	9	0.0814	14	0.0600	19	0.0497
5	0.1241	10	0.0756	15	0.0575	20	0.0481

Table 1. PAR vs. Fill Iters. for LaMa [10].

6 Why Existing Metrics Are Not Suitable for Comparing "Original Fill vs. Iterative Fill"?

While previous works often use metrics, such as LPIPS [14], PSNR or FID [5][9], to evaluate the performance of different inpainting methods. We found these metrics are not suitable to compare the inpainting qualities between original fill and iterative fill in our case, since the scores are often too similar to make a judgement. The main reason is that since both original fill and iterative fill share the same inpainting algorithm, the difference between their outputs are less obvious than the difference between different inpainting methods, even though the difference might be obvious to human perception. In addition, we found that even when the holes are on the background region, reconstruction metrics LPIPS [14] and PSNR still often prefer the image that is opposite to human judgement, where the typical observations are shown in Fig. 5. Thus, we used our proposed PAR metric, which is proven to have strong correlation with human perception in paper's section 4, as well as extensive user studies to evaluate the improvement between original fill and iterative fill, as discussed in section 6.2 in the main paper.

7 More Visual Results

To help readers have better understanding and more insights on our work, we provide more visual examples of human annotations and our predictions of inpainting artifacts, and results of our iterative fill model.

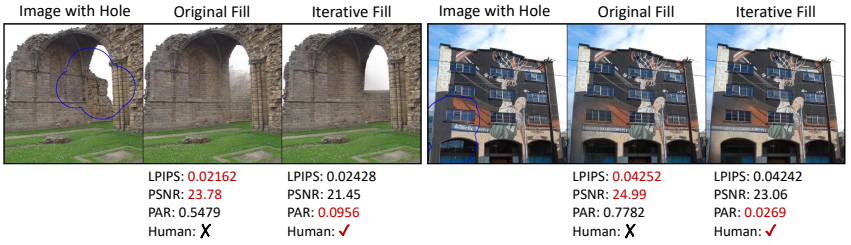

Fig. 5. Metric scores amd human preference for original fill and iterative fill.

7.1 Human Labelings

Visualize Subjective Opinions. As discussed in section 4.2 in the main paper, labeling perceptual artifacts is a highly subjective task, and different human subjects might have different opinions or standard to label. Here, we show more visual examples of different labels on the same filled images in Fig. 6.

More Human Labelings. In Fig. 7, we show more visual examples of perceptual artifacts labeling from the human professional team. The original hole masks and the artifacts regions are indicated by the blue and pink boundary, respectively. As we mentioned in section 3 in the main paper, there are 832 images that have nearly perfect fills so that human did not put any labels on these images. The visual illustration of these perfect fills are shown in the last row of Fig. 7. We can see that sometimes human's labels go outside of the actual hole mask, since we do not provide hole mask for the workers to avoid any potential bias. However, this won't be an issue, since we intersect the labels with the hole mask to clean up the overly labeled regions as a post-process.

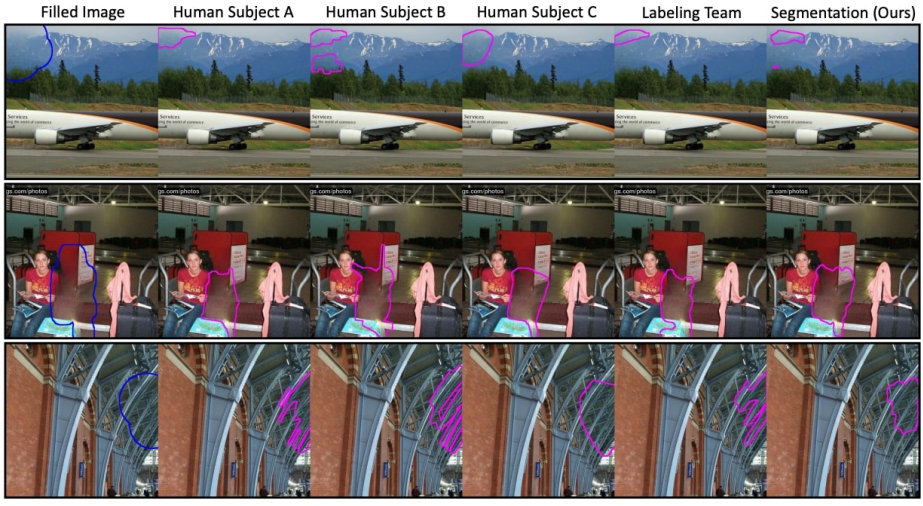

Fig. 6. More qualitative examples of visual comparison between multiple human subjects who label on the same images. The filled images with hole (blue boundary) are shown in the first column, and our segmentation results are shown in the last column.

7.2 Perceptual Artifacts Localization

In Fig. 8, we show more qualitative results of the perceptual artifacts segmentation across different inpainting methods. Note that our models are trained on filled images generated by LaMa [10], CoMod-GAN [16], and ProFill [13]. Nevertheless, our trained network generalizes reasonably well to unseen inpainting models, including EdgeConnect [8], DeepFillv2 [12], and PatchMatch [1].

7.3 Iterative Fill over Iterations

In Fig. 1, we show more examples of iteratively filled images during the process.

7.4 Original Fill vs. Iterative Fill

In Fig. 10, 11, 12, and 13, we show the comparisons between original fill and our iterative fill for LaMa [10], CoMod-GAN [16], ProFill [13], and EdgeConnect [8], respectively.

7.5 Situations where Iterative Fill Does Not Help

In section 6.2 in the main paper, we have studied how many cases that users think the iteratively filled images are better, similar, or worse than the original fills. The results show that users think iterative fill and original fill are similar for lots of cases. In this section, we take a deeper look at why this is the case.

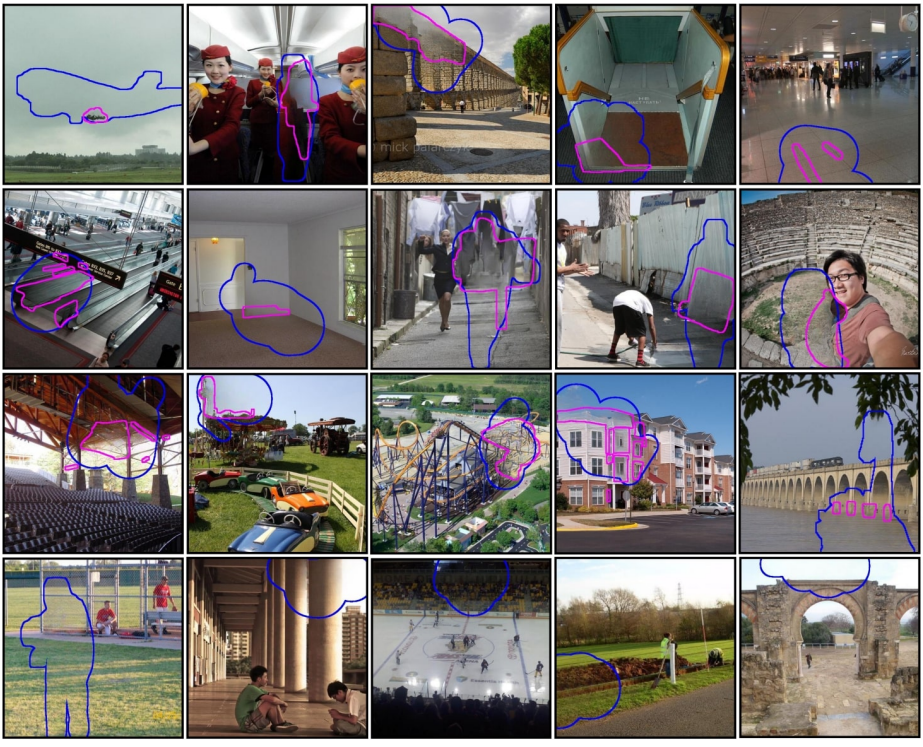

Fig. 7. More qualitative examples of perceptual artifacts labels from the professional human labeling team. The last row shows the perfectly filled images, so the workers do not label anything on them.

In general, we observe two major reasons that cause iterative fill are similar to the original fill. First, when the holes are easy, the inpainting algorithm could already fill the holes reasonably well in a single pass. In this case, our segmentation network usually would not detect much artifacts region, and thus the iterative fill would produce very similar or even identical images as the original fill, as shown in the left of Fig. 14. Second, when the holes are large and there are not enough useful context, the original fill would usually fail obviously and our artifacts segmentation network could pick up large artifacts regions, or even as large as nearly the entire hole. However, since the useful context is still very limited, the same struggle remains almost unchanged for the same inpainting algorithm and thus iterative fill would fail to produce better outputs, and thus can not further improve the inpainting quality, as shown in the right of Fig. 14.

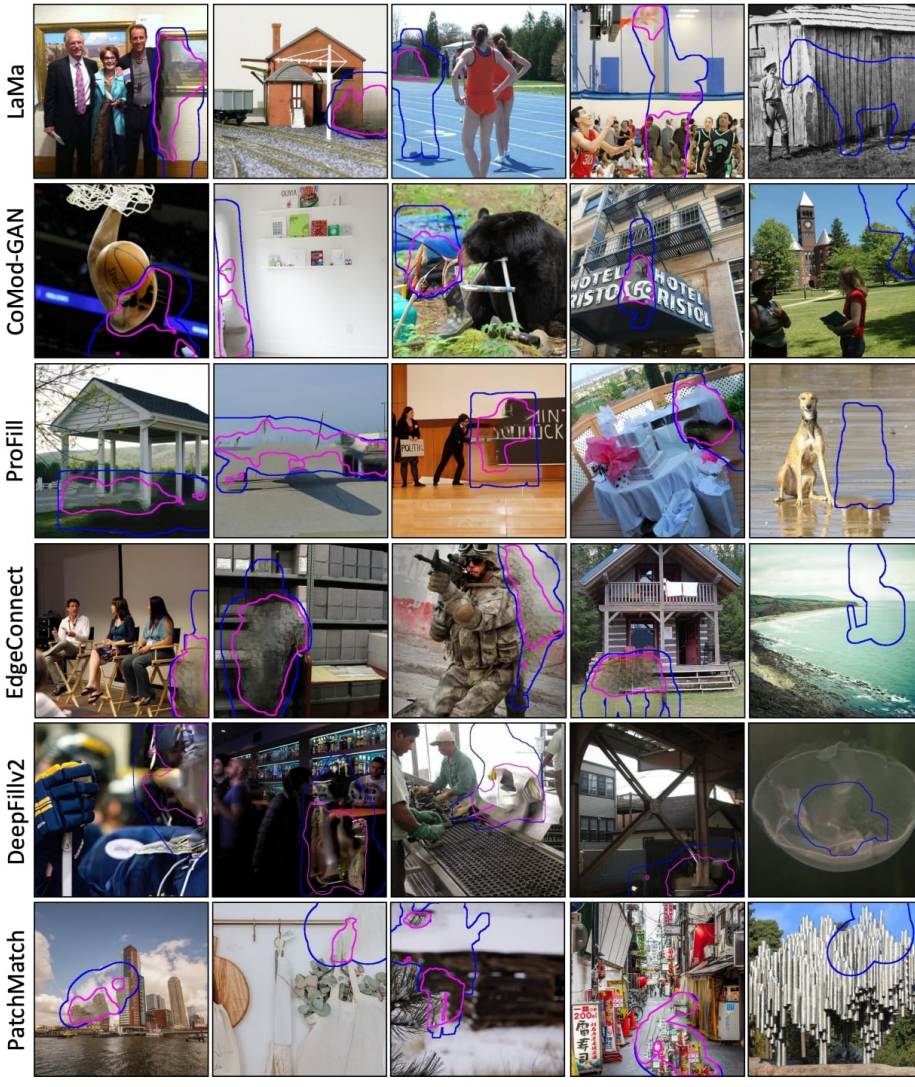

Fig. 8. More qualitative examples of the predicted perceptual artifacts localization from our segmentation network for different inpainting models. In the last column, our network does not predict any mask, since the filled images look almost perfect.

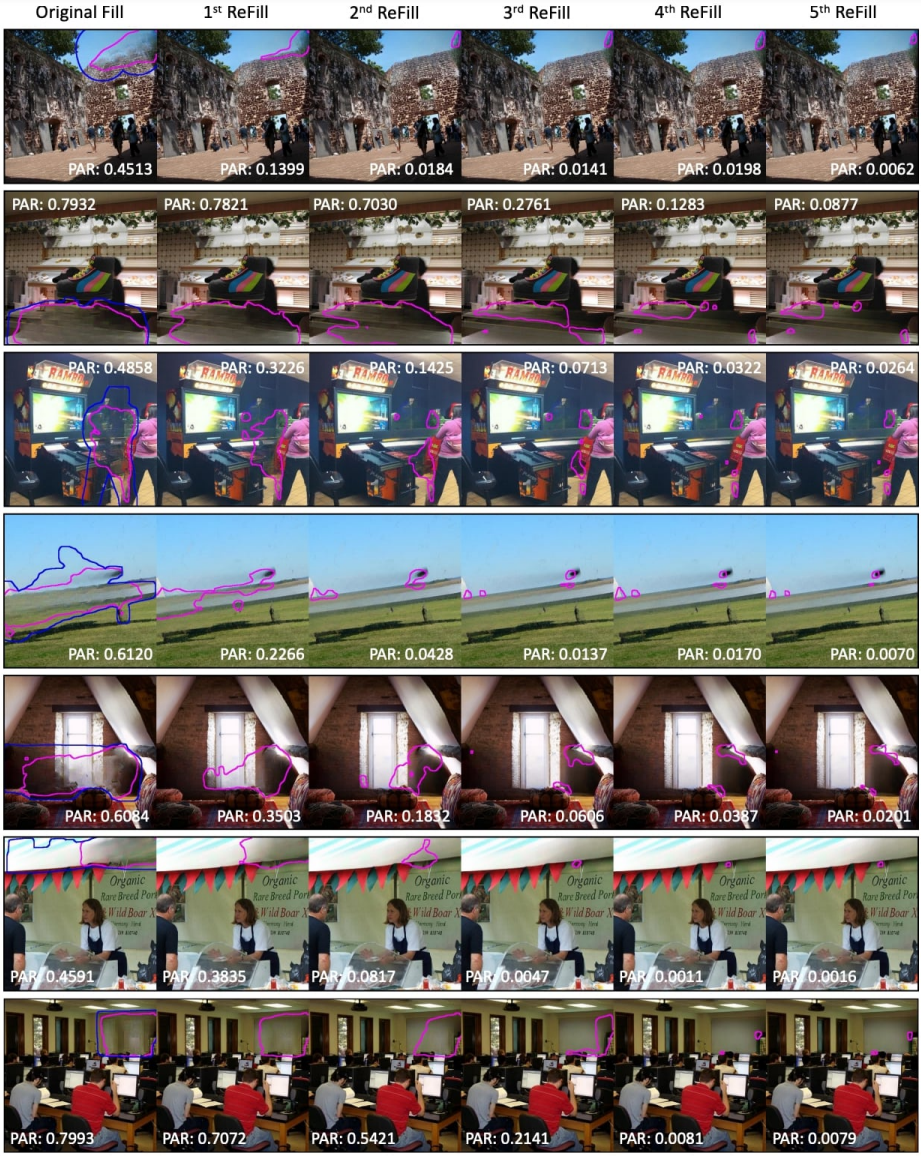

Fig. 9. More qualitative results of iterative fill by LaMa [11]. The blue and pink boundaries indicate the original hole mask and perceptual artifacts localization, respetively.

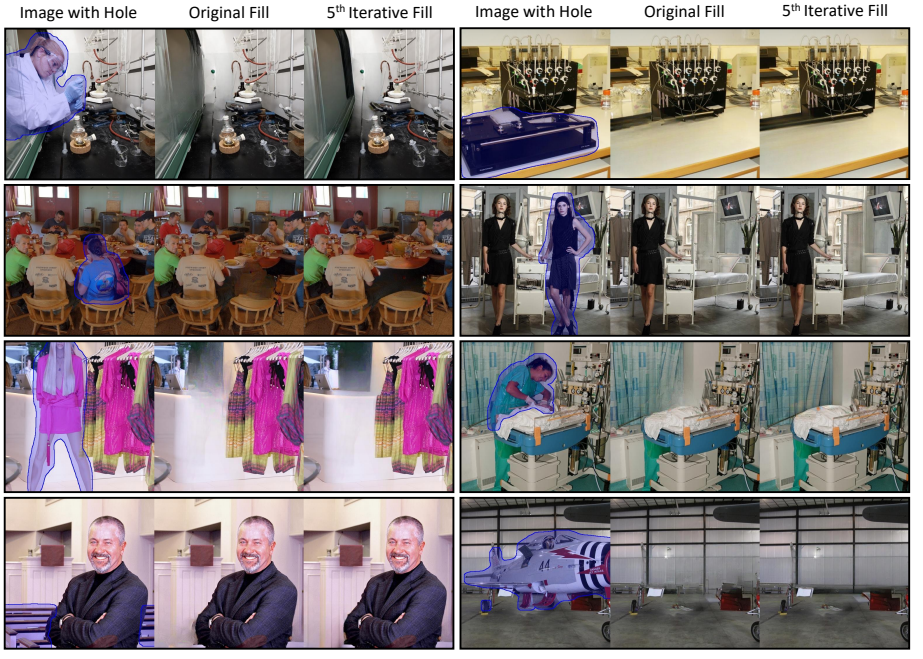

Fig. 10. Visual comparisons between the original fill and our iterative fill for LaMa [11].

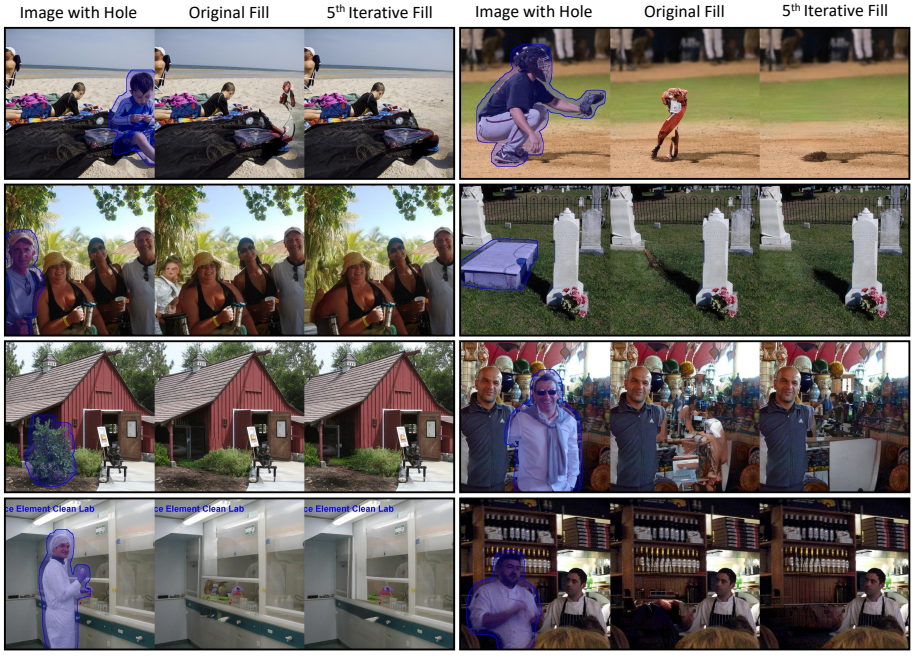

Fig. 11. Visual comparisons between the original fill and our iterative fill for CoMod-GAN [16].

Fig. 12. Visual comparisons between the original fill and our iterative fill for ProFill [13].

Fig. 13. Visual comparisons between the original fill and our iterative fill for EdgeConnect [8]. Note that the 5^{th} iterative fill still has obvious artifacts due to the limitation of the algorithm itself, but looks visually more pleasant than the original fill.

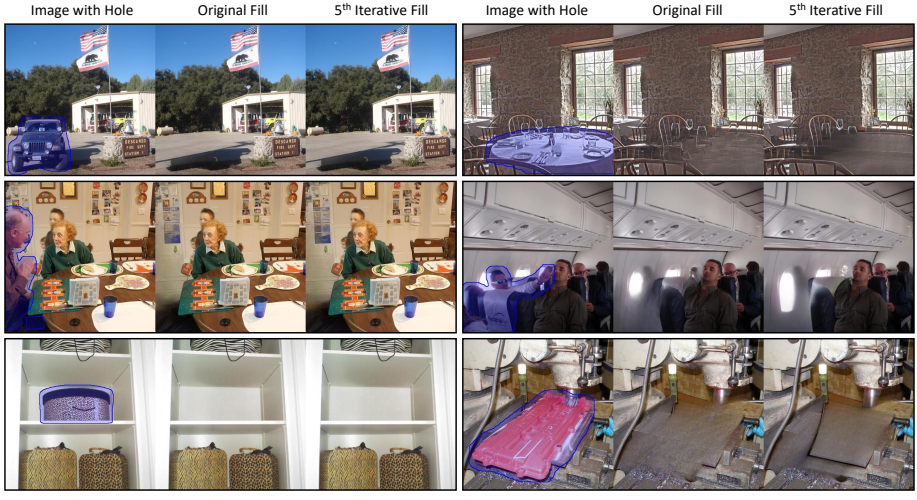

Fig. 14. Typical situations where iterative fill does not help. **Left**: when the holes are easy and original fill already looks good, our segmentation network would not detect much artifacts region and thus the iterative fill often looks very similar to the original fill. **Right**: when the holes are very large and context is limited, even if our segmentation network detects the artifacts region, iterative fill still can not properly fill the hole region and thus can not improve the inpainting quality for these hard cases.

References

1. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3), 24 (2009) 6
2. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010) 1
3. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 1
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 2
5. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 4
6. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 1
7. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) 2
8. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019) 6, 13
9. Parmar, G., Zhang, R., Zhu, J.Y.: On buggy resizing libraries and surprising subtleties in fid calculation. arXiv preprint arXiv:2104.11222 (2021) 4
10. Su, S., Yan, Q., Zhu, Y., Zhang, C., Ge, X., Sun, J., Zhang, Y.: Blindly assess image quality in the wild guided by a self-adaptive hyper network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3667–3676 (2020) 4, 6
11. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. arXiv preprint arXiv:2109.07161 (2021) 4, 9, 10
12. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4471–4480 (2019) 1, 6
13. Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: European Conference on Computer Vision. pp. 1–17. Springer (2020) 6, 12
14. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018) 4
15. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017) 2
16. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. arXiv preprint arXiv:2103.10428 (2021) 6, 11

17. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017) 1