Supplementary Material

Anonymous ECCV submission

Paper ID 3135

1 Corrections for the Main Paper

In Fig. 2 of the main paper, text labels (below images) and captions are incorrectly described. The left two images are real object removal requests; the middle two images are synthetic samples of previous approaches. The corrected figure and caption are shown here in Fig. 1

2 Results on high-resolution input

File "High-resolution-results.pdf" shows additional results for high-resolution input images of which short side is 1024.

3 Comparison with state-of-the-art methods

We show comparison of the two variants of our method *i.e.* **Ours*** and **Ours**, with more state-of-the-art methods including ContextAttention [9], PConv [6], FGAware [8], PENNet [11] and a recent progressive inpainting method FRRN [4][1].

3.1 Quantitative evaluation

Tab. 1 shows quantitative comparisons of our method with state-of-the-art methods evaluated on our test set with object-shaped holes.

3.2 Visual evaluation

We include visual comparisons of our methods with a larger set of existing methods on more synthetic samples and real object removal requests in the file "Visual-comparison.pdf". Page 1, 2 show visual comparison on synthetic test samples. Page 3, 4 shows visual comparisons on the real object removal requests on which we conduct the user study. Link to the corresponding requests are under each row. The methods for comparison are: Global&Local [5], PatchMatch [3], GConv [10], EdgeConnect [7], PENNet [11], ContextAttention [9], PConv [6], FGAware [8] and FRRN [4].

[1] Results of PConv and FGAware are provided by the authors. Results of PConv are obtained on images resized to 512×512. Other methods take input of original size.

Real requests Previous approaches Ours

Fig. 1: Comparison of input with holes. The first two columns are real object removal requests on the Web [2,1]. The second two are from PConv [6] and ContextAttention [9], respectively. The right two are our samples with object-shaped holes.

Table 1: Quantitative comparisons with state-of-the-art methods on our test set with object-shaped holes.

Method	L1 Loss	PSNR	SSIM
FFRN	.0289	25.66	.8787
ContextAttention	.0300	23.73	.8649
FGAware	.0244	26.32	.8811
PConv	.0212	27.57	.8876
PENNet	.0236	26.11	.8845
Ours*	.0194	28.20	.8985
Ours	.0205	27.67	.8949

4 Network architectures

Fig. 2 (same as Fig. 3 of the paper but with extra annotations) illustrates the overall structures of the iterative inpainting model and guided upsampling network. For convenience, we annotate convolutional blocks with dashed boxes in the figure. The coarse network and the image decoder of our iterative inpainting model have a similar architecture to GConv [10]. We also use gated convolution [10] in all of our convolution blocks and use contextual attention [9] in the bottleneck layer of our iterative inpainting model. We describe the details of these blocks shown in Fig. 2 in Tab. 2,3,4,5,6,7. As shown in Fig. 2, we use skip connections between the image decoder and confidence decoder of the iterative inpainting model as the entire generation process should be observed to evaluate confidence maps more accurately. We also use skip connections in the reconstruction module (block 5) of the guided upsampling network to preserve fine-grained details in the high-resolution input and generate a high-resolution result with more realistic textures. The dashed line with arrow in Fig .2 indicates skip connection. The dashed gray box represents feature maps copied from the skip connected layer. Input and output of skip connection layers are marked in red and blue in Tab. 2,3,4,5,6,7. Code will be made public available after the paper is published.

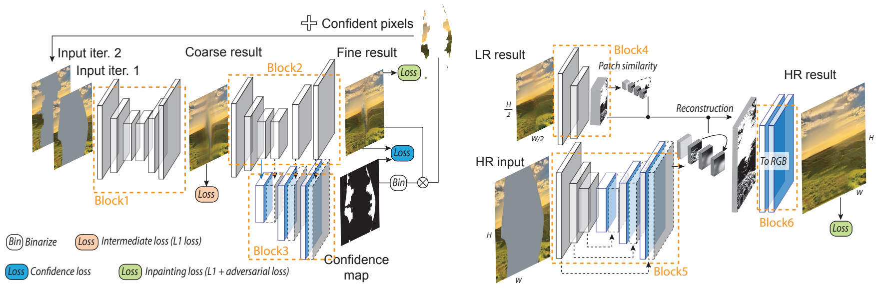

Fig. 2: Structures of the iterative model (left) and guided upsampling network (right) with extra annotations of blocks.

Table 2: Details of block 1 (coarse network) in Fig. 2.

Channel number	Kernel size	Stride	Dilation rate	Activation
48	5	1	1	ELU
96	3	2	1	ELU
96	3	1	1	ELU
192	3	2	1	ELU
192	3	1	1	ELU
192	3	1	1	ELU
192	3	1	2	ELU
192	3	1	4	ELU
192	3	1	8	ELU
192	3	1	16	ELU
192	3	1	1	ELU
192	3	1	1	ELU
2× nearest neighbor upsample				
96	1	1	1	
96	3	1	1	ELU
2× nearest neighbor upsample				
48	1	1	1	
24	3	1	1	ELU
3	3	1	1	Tanh

Table 3: Details of block 2 (image encoder and image decoder) in Fig. 2.

Branch 1					Branch 2				
C. n.	K. s.	S	D. r	A	C. n.	K. s.	S	D. r	A
48	5	1	1	ELU	48	5	1	1	ELU
48	3	2	1	ELU	48	3	2	1	ELU
96	3	1	1	ELU	96	3	1	1	ELU
96	3	2	1	ELU	192	3	2	1	ELU
192	3	1	1	ELU	192	3	1	1	ELU
192	3	1	1	ELU	192	3	1	1	ReLU
192	3	1	2	ELU	Contextual attention				
192	3	1	4	ELU					
192	3	1	8	ELU	192	3	1	1	ELU
192	3	1	16	ELU	192	3	1	1	ELU

Concatenate

Channel number	Kernel size	Stride	Dilation rate	Activation
192	3	1	1	ELU
192	3	1	1	ELU
2× nearest neighbor upsample				
96	1	1	1	
96	3	1	1	ELU
2× nearest neighbor upsample				
48	1	1	1	
24	3	1	1	ELU
3	3	1	1	Tanh

Table 4: Details of block 3 (confidence decoder) in Fig. 2.

Channel number	Kernel size	Stride	Dilation rate	Activation
192	3	1	1	ELU
192	3	1	1	ELU
2× nearest neighbor upsample				
96	1	1	1	ELU
96	3	1	1	ELU
2× nearest neighbor upsample				
48	1	1	1	ELU
24	3	1	1	ELU
1	3	1	1	Sigmoid

Table 5: Details of block 4 (similarity network) in Fig. 2.

Channel number	Kernel size	Stride	Dilation rate	Activation
48	5	1	1	ELU
96	3	2	1	ELU
96	3	1	1	ELU
192	3	2	2	ELU
48	3	1	1	ELU

Table 6: Details of block 5 (reconstruction network) in Fig. 2.

Channel number	Kernel size	Stride	Dilation rate	Activation
48	5	1	1	ELU
96	3	2	1	ELU
96	3	1	1	ELU
192	3	2	1	ELU
$2\times$ nearest neighbor upsample				
96	1	1	1	
96	3	1	1	ELU
$2\times$ nearest neighbor upsample				
48	1	1	1	
48	3	1	1	ELU

Table 7: Details of block 6 (ToRGB) in Fig. 2.

Channel number	Kernel size	Stride	Dilation rate	Activation
48	3	1	1	ELU
3	3	1	1	Tanh

References

1. Can someone please remove the backpack and 'lead' from my sons back? would love to have this picture of my kids without it! https://www.reddit.com/r/PhotoshopRequest/comments/6szh1i/specific_can_someone_please_remove_the_backpack/
2. Could someone help me remove background people - especially the guys head? will venmo $5. https://www.reddit.com/r/PhotoshopRequest/comments/b2y0o5/specific_paid_could_someone_help_me_remove/
3. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing **28**(3), 24 (2009)
4. Guo, Z., Chen, Z., Yu, T., Chen, J., Liu, S.: Progressive image inpainting with full-resolution residual network. In: Proceedings of the 27th ACM International Conference on Multimedia. ACM (2019)
5. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (ToG) **36**(4), 107 (2017)
6. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: European Conference on Computer Vision (2018)
7. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)
8. Xiong, W., Yu, J., Lin, Z., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
9. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5505–5514 (2018)
10. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: IEEE International Conference on Computer Vision (2019)
11. Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1486–1494 (2019)