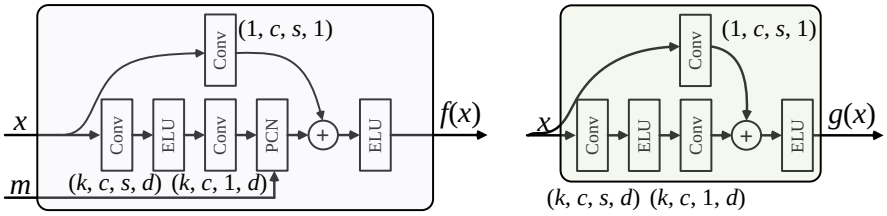
VCNet: A Robust Approach to Blind Image Inpainting Supplementary Material

Yi Wang[1], Ying-Cong Chen[2], Xin Tao[3], and Jiaya Jia[1,4]

[1]The Chinese University of Hong Kong [2]MIT CSAIL [3]Kuaishou Technology [4]SmartMore
{yiwang, leojia}@cse.cuhk.edu.hk
ycchen@csail.mit.edu jiangsutx@gmail.com

1 Network architectures

The detailed structure of our model is given as follows. We use PCB(k, c, s, d) to denote a probabilistic contextual block (PCB) block, as illustrated in Figure 1(a). k, c, s, and d denote kernel size, filter number, stride size, and dilation rate, respectively. Similarly, residual block (RB) is defined as RB(k, c, s, d), which is given in Figure 1(b). Let DB(k, c, s) denote a convolution-LeakyReLU(0.2) layer, CB(k, c, s) denote a convolution-ELU layer, Conv(k, c, s) denote a convolution layer, and $\times 4$ denote nearest-neighbor upsampling operator. Our visual consistency network consists of mask prediction network (MPN) and region inpainting network (RIN).



(a) PCBlock (b) ResBlock

Fig. 1: The illustration of our computation units.

MPN: RB(5,64,2,1)-RB(3,128,2,1)-RB(3,128,1,2)-RB(3,128,1,4)-**MPN-bottleneck**-RB(3, 128, 1, 1)-$\times 4$-RB(3, 64, 1,1)-$\times 4$-RB(3,32,1,1)-CB(3,16,1)-Conv(3,1,1)-logit-sigmoid

RIN: PCB(5,32,1,1)-PCB(3,64,2,1)-PCB(3,64,1,1)-PCB(3,128,2,1)-PCB(3,128,1,1)-PCB(3,128,1,2)-PCB(3,128,1,2)-PCB(3,128,1,4)-PCB(3,128,1,4)-PCB(3,128,1,1)- concatenated with **MPN-bottleneck**-$\times 4$-PCB(3,64,1,1)-PCB(3,64,1,1)-$\times 4$-PCB(3,32,1,1)-PCB(3,32,1,1)-Conv(3,3,1)-clipped to [-1,1]

Discriminator: DB(5,64,2)-DB(5,128,2)-DB(5,128,2)-DB(5,256,2)-DB(5,512,2)-FC (output channel 1)

2 Implementation details

Baseline methods We use CA [10], GMC [9], PC [6], and AttentiveGAN [8] for comparison. The implementations of these baselines are based on the official codes from their authors. CA, GMC, and PC are trained in the same way as ours. AttentiveGAN is trained in the author's given approach with a few data (20 images in 480×720).

Training data generation Direct blending image $\mathbf{O}$ and $\mathbf{N}$ using Eq. 1 will lead to noticeable edges, which are strong indicators for the model distinguish noisy areas. This will inevitably sacrifice the semantic understanding capability of the used model. Thus, we dilate the $\mathbf{M}$ into $\tilde{\mathbf{M}}$ by the iterative Gaussian smoothing in [9] and then apply $\mathbf{M} \leftarrow \tilde{\mathbf{M}}$ in Eq. 1 to generate degradation images for training.

Training strategy There are two training stages. In the first stage, MPN and RIN are separately trained: optimizing $\min_{\theta_F} \mathcal{L}_m(F(\mathbf{I}), \mathbf{M})$, and $\min_{\theta_G} \mathcal{L}_g(G(\mathbf{I}|\mathbf{M}), \mathbf{O})$ using Adam solver [4] ($\beta_1 = 0.5$, $\beta_2 = 0.9$) with both the learning rates as $1e-4$, $\lambda_{adv} = 0$, and batch size as 8. Then after both networks converge (around 80k iterations for FFHQ), we jointly optimize $\min_{\theta_F, \theta_G} \lambda_m \mathcal{L}_m(F(\mathbf{I}), \mathbf{M}) + \mathcal{L}_g(G(\mathbf{I}|F(\mathbf{I})), \mathbf{O})$ using Adam solver ($\beta_1 = 0.5, \beta_2 = 0.9$) with $\lambda_m = 2.0$, learning rate $1e-5$, and batch size 4. The convergence of this stage takes around another 80k iterations for FFHQ. For Places2 and ImageNet, the first stage costs at least 140k iterations.

Hyper-parameters tuning Our evaluation results are produced with the following parameters setting: $\lambda_r = 1.4$, $\lambda_s = 1e-4$, $\lambda_f = 1e-3$, and $\lambda_a = 1e-3$. Reasonable results could be expected with $\lambda_r \in [0.6, 5]$, $\lambda_s \in [1e-4, 1e-3]$, $\lambda_f \in [1e-3, 5e-2]$, and $\lambda_a \in [2e-4, 1e-3]$.

Dataset descriptions We use FFHQ[1] [3], CelebA-HQ [2] [2], ImageNet[3] [1], Places2[4] [12], and Raindrop removal[5] [8] in our experiments.

3 More experiments

3.1 Self-supervised representation learning

Similar to Pathak *et al.* [7], we train the encoder of a standard AlexNet [5] on ImageNet from scratch along with a decoder in blind inpainting setting (with only l_1 loss, and the noisy images are randomly chosen from ImageNet as well). The linear classification accuracy on ImageNet using the trained conv5 features is 16.1%, higher than random (14.1%) and Pathak *et al.* (15.5%).

[1] https://github.com/NVlabs/ffhq-dataset
[2] https://drive.google.com/drive/folders/0B4qLcYyJmiz0TXY1NG02bzZVRGs
[3] http://image-net.org/download
[4] http://places2.csail.mit.edu/download.html
[5] https://drive.google.com/drive/folders/1e7R76s6vwUJxILOcAsthgDLPSnOrQ49K

(a) (b) (c) (d) (e) (f)

Fig. 2: Applying our blind inpainting to face images (FFHQ) iteratively. (a) Input image. (b)-(e) our results by applying it iteratively from 1 to 4 times, respectively. (f) Ground truth.

Fig. 3: Blind inpainting changes by ρ in PCN on FFHQ. From the left to right: the ground truth, input, and results with ρ from 0 to 1 with step size 0.2. Best viewed with zoom-in.

3.2 Iterative processing

Fig. 2 shows the visual changes brought by applying our method iteratively to an image. It will further edit degraded regions for fitting context but compromise image fidelity as it modifies a few valid regions gradually. Generally, the model transforms the input image into the one in its learned manifold. We suppose the oil painting effects in Fig. 2(d)-(e) are caused by the used perceptual loss.

3.3 More ablation studies

How the inconsistency pixel ratio of ρ affects PCN is presented in Figure 3. As said in the paper, increasing ρ makes VCN tend to generate missing parts based on context instead of blending the introduced 'noise'.

More visual comparisons on ablation study about different VCN variants are given in Figure 4.

3.4 More visual comparisons

Graffiti removal We give more comparisons against other methods [10,9,6] on FFHQ damaged with graffiti (Figure 5 and 6). The used graffiti patterns are self-built (7 types with size 256×256) and not included in the training.

Synthetic degraded images using natural images and random strokes Synthetic experiments on FFHQ (Figure 7 and 8), Places2 (Figure 9 and 10) and ImageNet (Figure 11 and 12) are presented.

Model generalization on raindrop removal More visual evaluations about the severe raindrop removal are given in Figure 13 and 14.

Model robustness against different filling contents in the missing region Figure 15 and 16 show more examples of the results of our methods for various fillings (not included in the training) in blind inpainting.

4 Another face-swap example

Figure 17 gives more examples about face-swap on FFHQ.

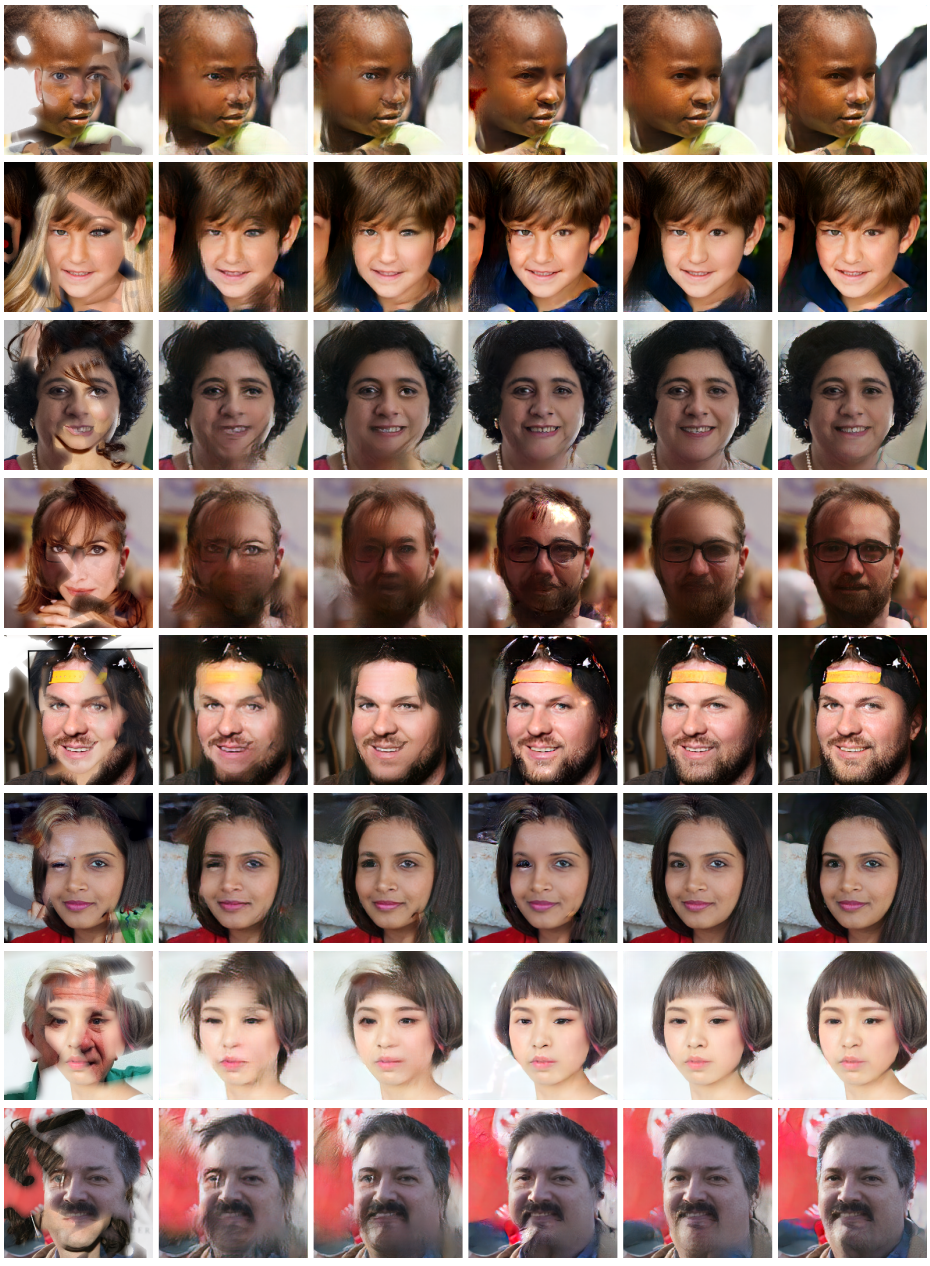


Fig. 4: Visual comparisons on FFHQ using VCN variants. (a) Input image. (b) VCN w/o MPN. (c) VCN w/o skip. (d) VCN w/o semantics. (e) VCN-RM. (f) VCN full.

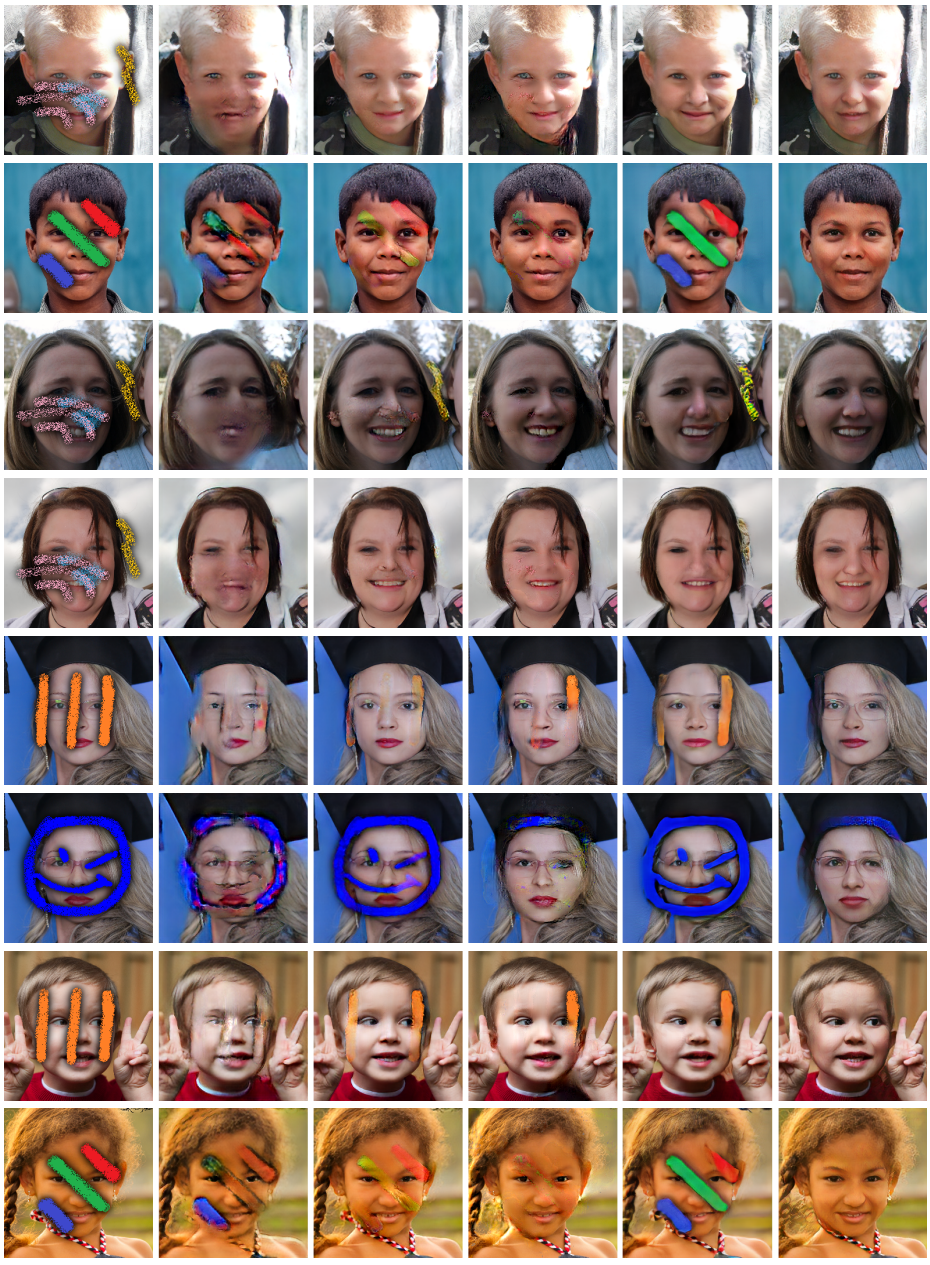


Fig. 5: Visual comparisons on FFHQ damaged by graffiti. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.

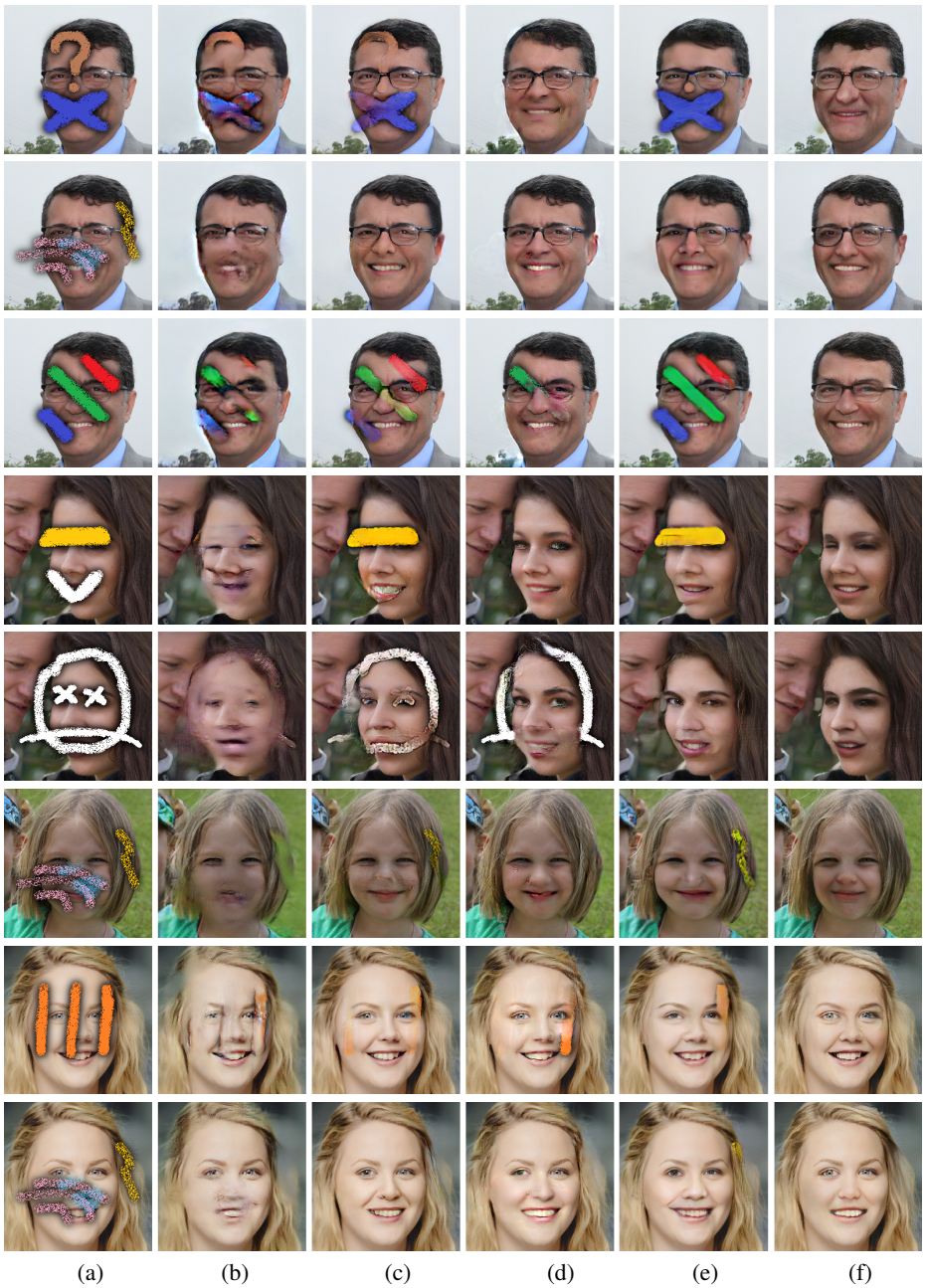
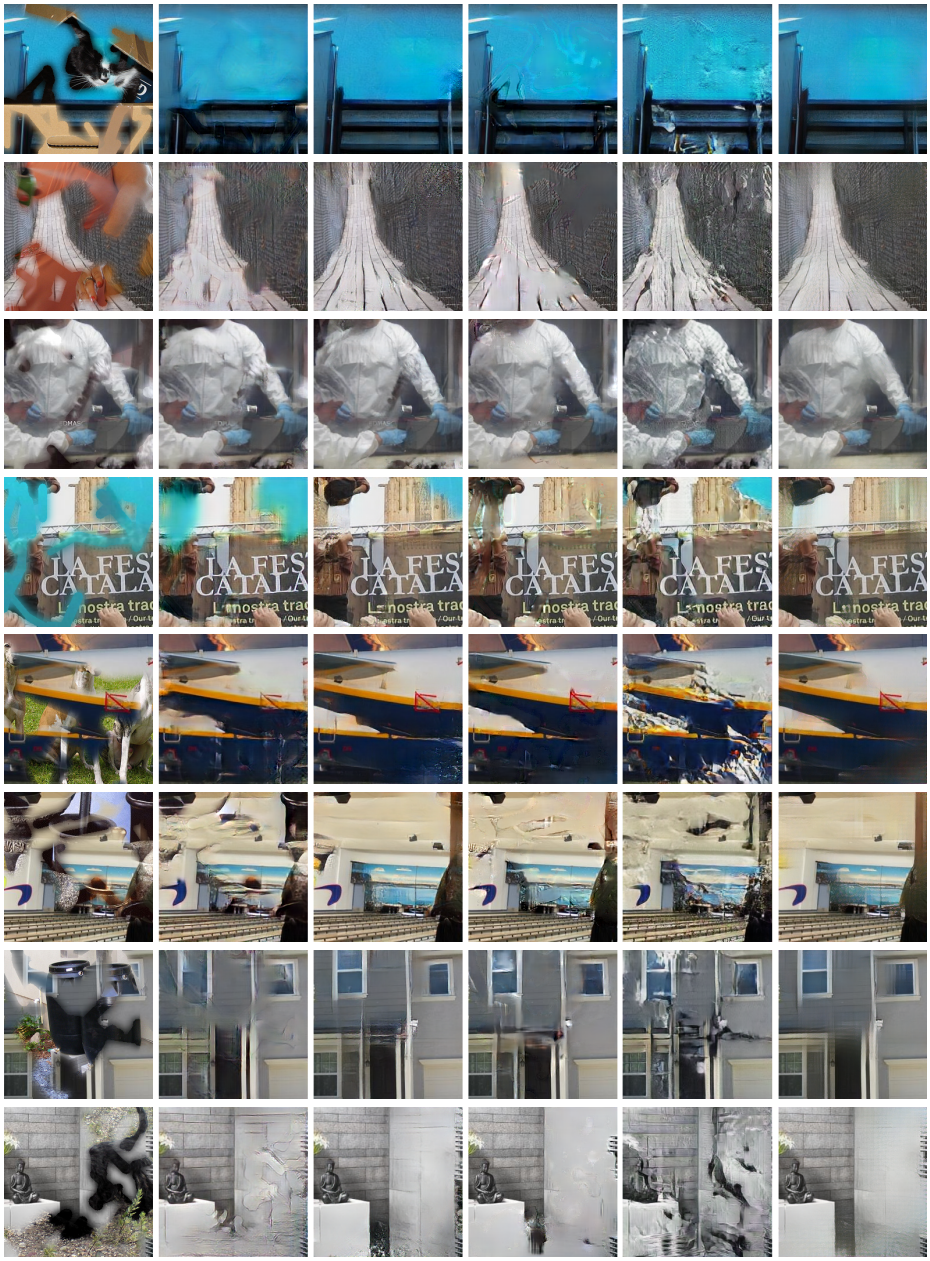


Fig. 6: Visual comparisons on FFHQ damaged by graffiti. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.

(a) (b) (c) (d) (e) (f)

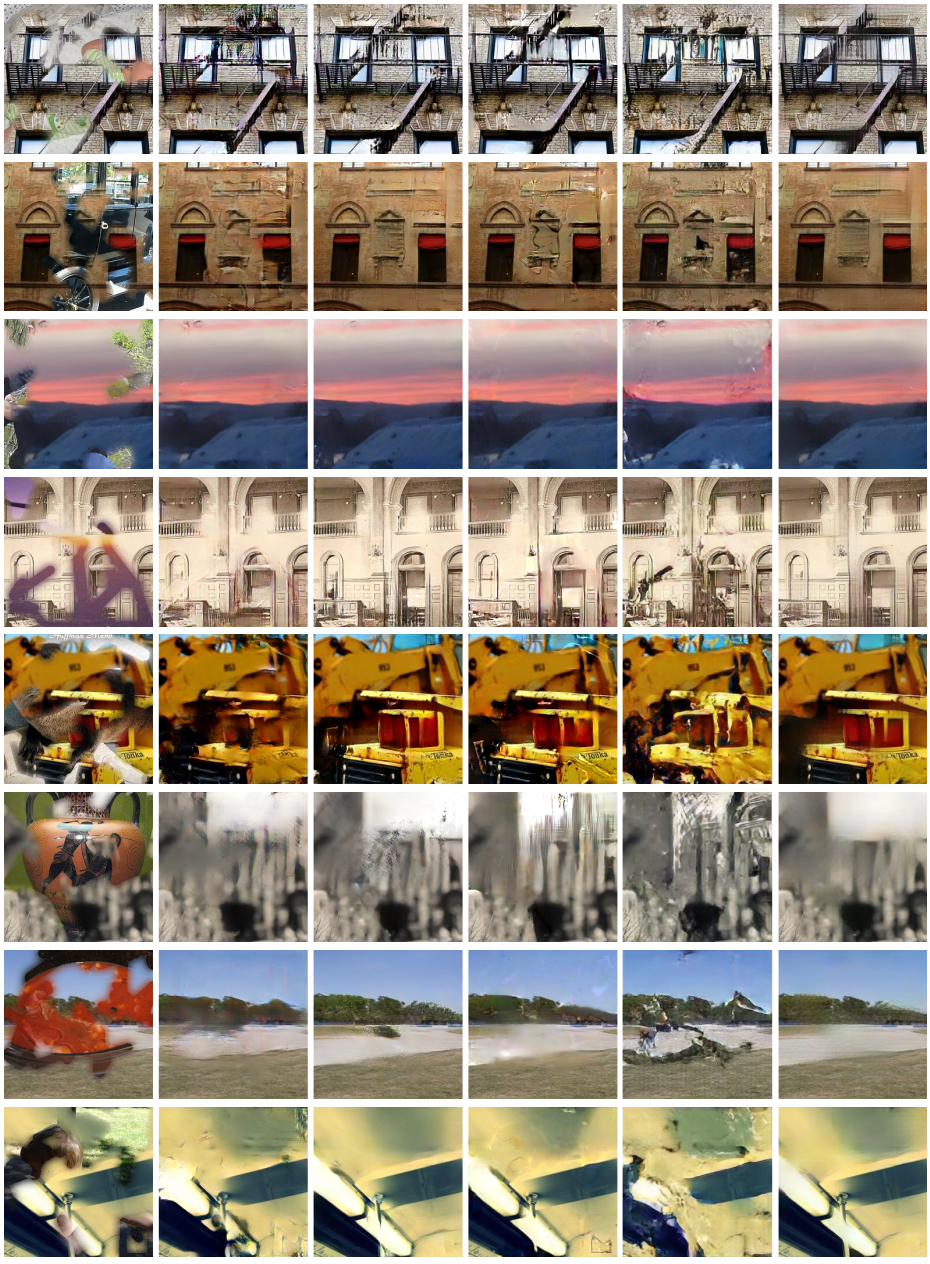
Fig. 7: Visual comparisons on FFHQ damaged by the images from CelebA-HQ testing set with random strokes. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.

Fig. 8: Visual comparisons on FFHQ damaged by the images from CelebA-HQ testing set with random strokes. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.



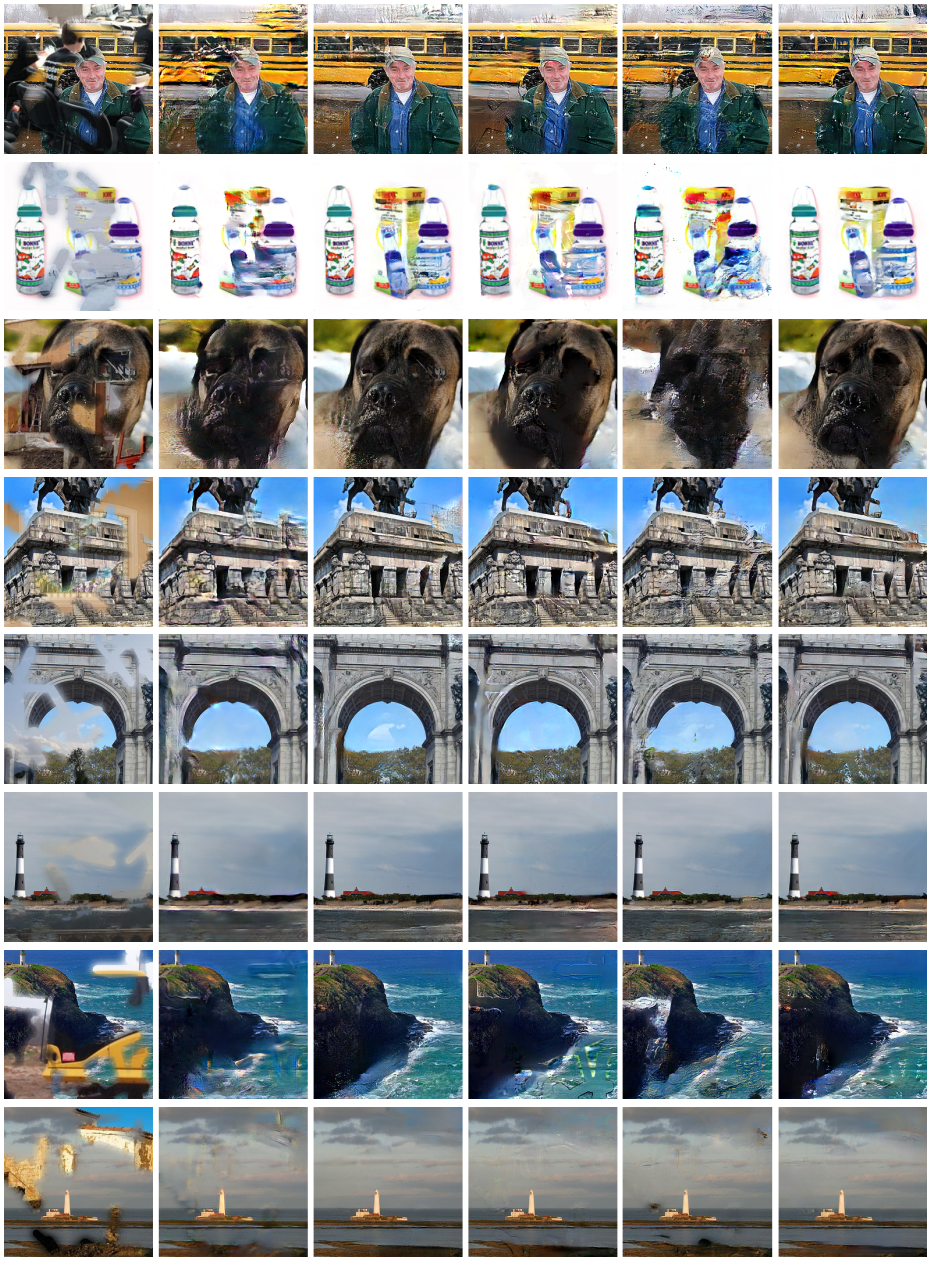
(a) (b) (c) (d) (e) (f)

Fig. 9: Visual comparisons on Places2 damaged by the images from ImageNet testing set with random strokes. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.



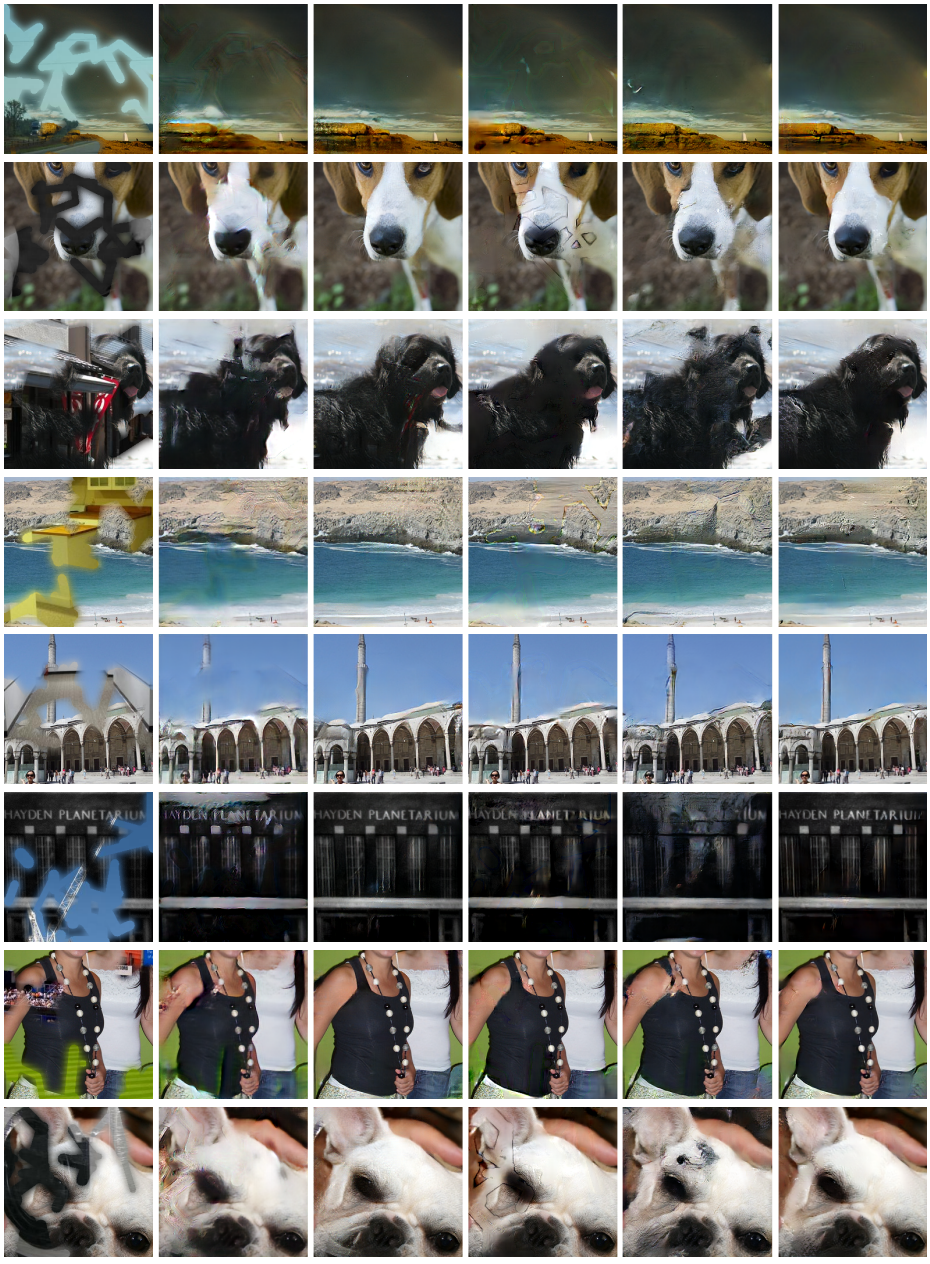
(a) (b) (c) (d) (e) (f)

Fig. 10: Visual comparisons on Places2 damaged by the images from ImageNet testing set with random strokes. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.



(a) (b) (c) (d) (e) (f)

Fig. 11: Visual comparisons on ImageNet damaged by the images from Places2 testing set with random strokes. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.



(a) (b) (c) (d) (e) (f)

Fig. 12: Visual comparisons on ImageNet damaged by the images from Places2 testing set with random strokes. (a) Input image. (b) CA [10]. (c) GMC [9]. (d) PC [6]. (e) GC [11]. (f) Our results. Best viewed with zoom-in.

(a) (b) (c)

Fig. 13: Visual evaluations on raindrop removal dataset. (a): Input image. (b): AttentiveGAN [8]. (c): Ours. Best viewed with zoom-in.

(a) (b) (c)

Fig. 14: Visual evaluations on raindrop removal dataset. (a): Input image. (b): AttentiveGAN [8]. (c): Ours. Best viewed with zoom-in.

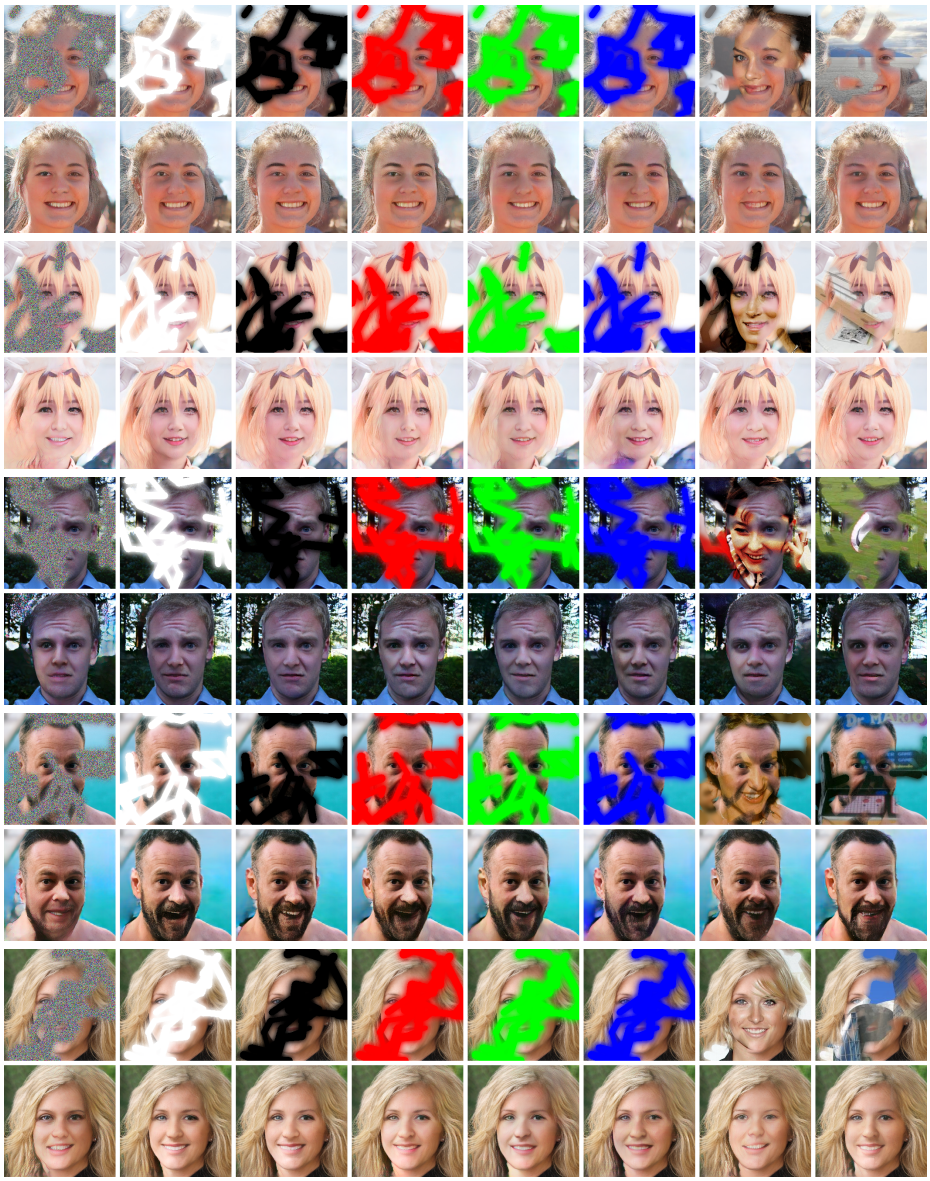


Fig. 15: Visual evaluations on FFHQ with random masks filled with different contents. The 1st, 3rd, 5th, 7th, 9th rows: input, the remaining rows: the corresponding results from our model. The last two images are filled with contents drew from CelebA-HQ and ImageNet respectively.

Fig. 16: Visual evaluations on Places2-HD (512×680) with random masks filled with different contents. (a): input. (b): the corresponding results from our model. Best viewed with zoom-in.

Fig. 17: Visual editing (face swap) on FFHQ. The first and third row: images with coarse editing where a new face is pasted at the image center, the second and fourth row: the corresponding results from our model. Best viewed with zoom-in.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
2. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
3. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948 (2018)
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
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: ECCV. pp. 85–100 (2018)
7. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR. pp. 2536–2544 (2016)
8. Qian, R., Tan, R.T., Yang, W., Su, J., Liu, J.: Attentive generative adversarial network for raindrop removal from a single image. In: CVPR. pp. 2482–2491 (2018)
9. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: NeurIPS (2018)
10. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. arXiv preprint arXiv:1801.07892 (2018)
11. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: ICCV. pp. 4471–4480 (2019)
12. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. TPAMI (2017)