Supplementary File for Wide-Context Semantic Image Extrapolation

1. Network Architectures

The specific details of our designed networks are given here. Code will be available at `https://github.com/shepnerd/outpainting_srn`.

In the following representation, K denotes kernel size, S denotes stride size, C denotes channels number, D denotes dilation ratio, $\times 4$ denotes nearest-neighbor upsampling input into 4 times size, P denotes symmetric padding, FC denotes fully connected layer with one output channel, $(top, left, bottom, right)$ denotes filling margin, h denotes input height, w denotes input width, and $r_1 r_2$ denotes expansion ratio where $r_1 = (h + top + bottom)/h$ and $r_2 = (w + left + right)/w$. Before convolution, all fed data will be zero padded except that in unfold operation. Each convolution is followed by ELU [1] activation function except those right before output. Our SRN network is sequentially connected by FEN and CPN.

FEN: K5S1C64-K3S2C128-K3S1C128-K3S2C256-K3S1C256-K3S1C256-K3S1C256D2-K3S1C256D4-K3S1C256D8-K3S1C256D16-K3S1C256-K3S1C256-$\times$4-K3S1C128-K3S1C128-$\times$4-K3S1C64-**Feature Expansion Operator**

Feature Expansion Operator:

- **Deconvolution**: K3S(r_1,r_2)C64.
- **Unfold**: P([0, 0],[top, $bottom$], [$left$, $right$],[0, 0])-K3S1C64.
- **Sub-pixel convolution**: K3S1C64$r_1 r_2$-Eq. 1.

CPN: K5S1C64-K3S2C128-K3S1C128-K3S2C256-K3S1C256-K3S1C256-K3S1C256D2-K3S1C256D4-K3S1C256D8-K3S1C256D16-K3S1C256D16-K3S1C256-**CN**-K3S1C256-$\times$4-K3S1C128-K3S1C128-$\times$4-K3S1C64-K3S1C32-K3S1C3-clip to [-1, 1]

Context Critic: K5S2C64-K5S2C128-K5S2C256-K3S1C1-Eq. 10

Global Critic: K5S2C64-K5S2C128-K5S2C256-K5S2C128-FC

2. Full Training Algorithm

The detailed training algorithm is given in Algorithm 1.

Algorithm 1 Training algorithm

for $i = 0$ to $maxIters$ **do**
  if $i \leq \lfloor maxIters/2 \rfloor$ **then**
    update G with Eq. 12 by setting λ_{mrf}, λ_{adv} to 0s.
  else
    Generate filling mask $\mathbf{M}$ by m and relative spatial variant weight matrix $\mathbf{M}_w$ by Eq. 5.
    for $j = 0$ to 5 **do**
      Sample a batch images $\mathbf{Y}$, a filling margin variable m, a random number $t \sim U[0, 1]$.
      Produce input $\mathbf{X}$ by cropping $\mathbf{Y}$ based on m.
      Infer $\hat{\mathbf{Y}} = G(\mathbf{X}, m) \odot \mathbf{M} + \mathbf{Y} \odot (\mathbf{1} - \mathbf{M})$.
      Make an interpolation $\tilde{\mathbf{Y}} = t\mathbf{Y} + (1 - t)\hat{\mathbf{Y}}$.
      Update two discriminators $D_{context/global}$ with $\mathbf{Y}$, $\hat{\mathbf{Y}}$, and $\tilde{\mathbf{Y}}$.
    end for
    update G with Eq. 12.
  end if
end for

3. Details About Experimental Settings

About CA The generative model capacity of CA [8] is doubled as that in the original paper for better generation performance and fairness.

About ED The network design details of ED (encoder-decoder) are given below (The denotations follow the same protocol as the above):

ED: K5S1C64-K3S2C128-K3S1C128-K3S1C128-K3S2C256-K3S1C256-K3S1C256-K3S1C256-K3S1C256-K3S1C256D2-K3S1C256D4-K3S1C256D8-K3S1C256D16-K3S1C256D2-K3S1C256D4-K3S1C256D8-K3S1C256D16-K3S1C256-K3S1C256-K3S1C256-K3S1C256-$\times$4-K3S1C128-K3S1C128-K3S1C128-$\times$4-K3S1C64-K3S1C64-K3S1C32-K3S1C3-clip to [-1, 1]

Notes of the Used Datasets CUB200 [6]: Training on 10000 images and testing on the left 1788 ones. Here 10000 training images include original 5994 training and randomly selected 4006 testing images. The rest 1788 test-

	CelebA-HQ	CUB200	DeepFashion	Paris street view	Places2	Cityscapes	ETHZ Synthesizability
SRN $>$ CA	97.54%	96.42%	93.68%	62.57%	84.62%	91.26%	89.84%

Table 1. User study statistics. Each entry gives the percentage of cases where results by our approach are judged more realistic than another solution.

ing images are for evaluation. We split the raw CUB200 dataset in this way since the given training set is relatively small for conditional image generation training. All images have been cropped based on the given bird location rectangle and resized to 256×256.

ETHZ Synthesizability [2]: Training on randomly chosen 90% images of each class and testing on the remaining ones. All images are resized to 256×256.

Dog [4] & Bedroom [7]: Training on their given training set and testing on the validation set. The used images are all resized into 256×256.

4. More User Studies

We conduct more user studies between CA [8] and SRN on some other datasets. The experimental protocols and results (Table 1) are all consistent with that in the paper.

5. More Ablation Studies

More visual comparisons on using different structures (Figure 1), different feature expansion operators (Figure 2), different reconstruction loss (Figure 3), and the effects of context normalization (Figure 4: w/ or w/o CN, and Figure 5: AdaIN [3] *vs.* CN), context adversarial loss (Figure 6), and ID-MRF regularization (Figure 7) are given below.

6. More Qualitative Evaluations

Image expansion on faces (Figure 1 and 8), bird (Figure 9 and 10), human pose (Figure 11), dog (Figure 12), bedroom (Figure 13), Paris street view (Figure 14), cityscapes (Figure 15), and Places2 (Figure 16).

Image expansion on random locations (Figure 8, 10, and 12).

Figure 17 shows the applicability of our method on texture synthesis.

References

[1] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[2] Dengxin Dai, Hayko Riemenschneider, and Luc Van Gool. The synthesizability of texture examples. In *CVPR*, 2014.

[3] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

[4] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[5] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *NeurIPS*, 2018.

[6] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[7] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[8] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint*, 2018.

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

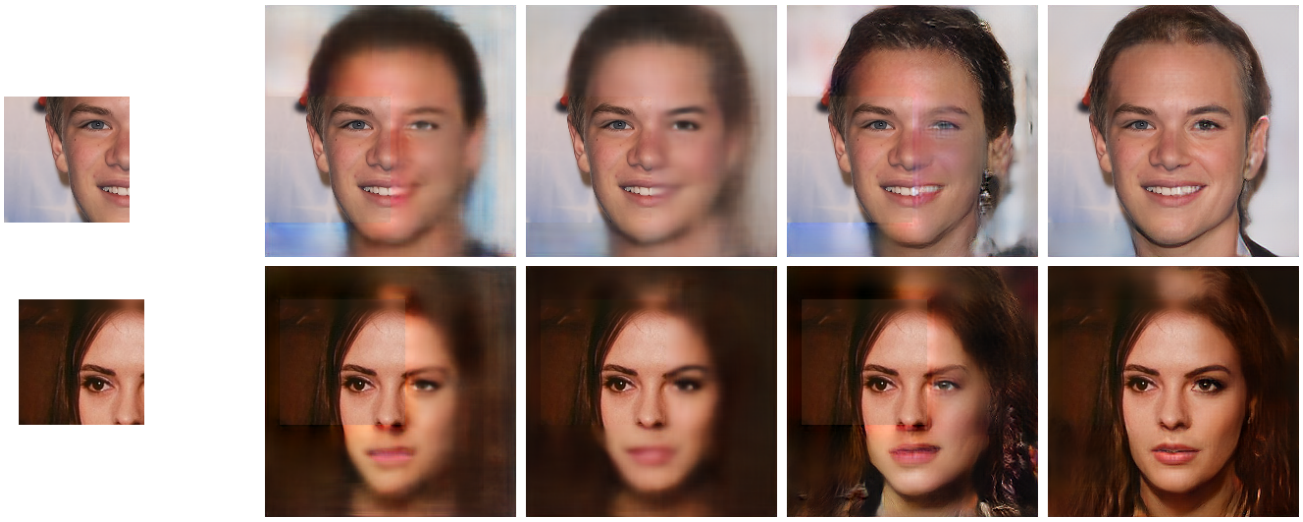
Figure 1. Visual comparisons of different network structures on CelebA-HQ. (a) Input image. (b) Coarse-to-fine (CA [8]). (c) Naive encoder-decoder. (d) SRN-HR. (e) SRN.

(a) (b) (c) (d) (a) (b) (c) (d)

Figure 2. Visual comparisons of different feature expansion operators on CelebA-HQ. (a) Input images. (b) Deconv. (c) Unfold. (d) Sub-pixel conv.

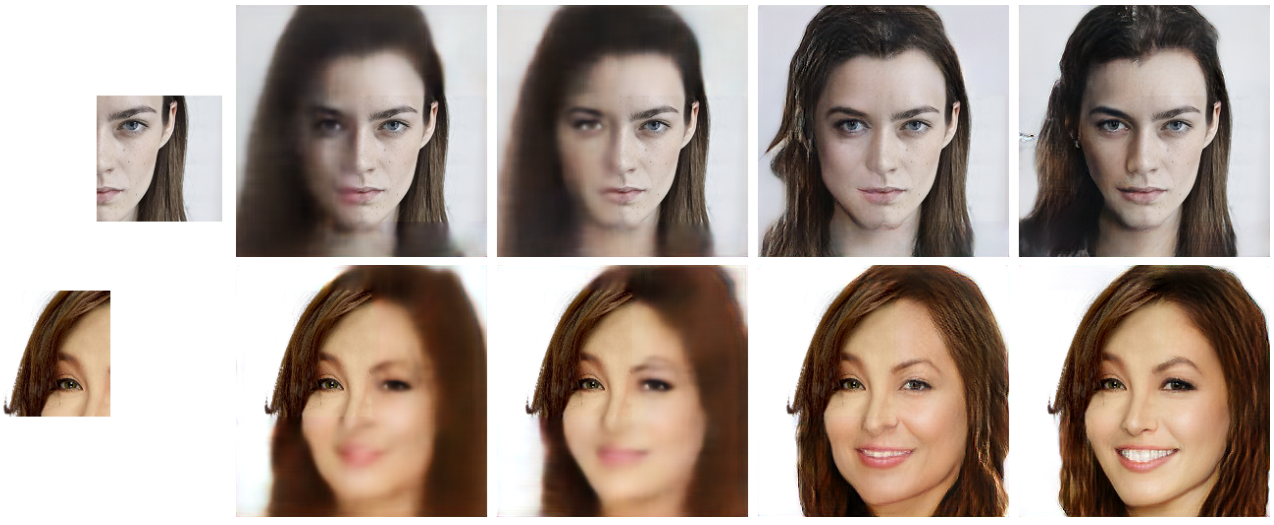
(a) (b) (c) (d) (a) (b) (c) (d)

Figure 3. Visual comparisons of SRNs only using different reconstruction losses on CelebA-HQ. (a) Input images. (b) Vanilla l_1 loss. (c) Confidence-driven loss [5]. (d) Relative spatial variant loss.



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

Figure 4. Visual comparisons of using CN (or not) on CelebA-HQ. (a) Input images. (b) SRN w/o CN in pre-training phase. (c) SRN w CN in pre-training phase. (d) SRN w/o CN. (e) SRN w CN.



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

Figure 5. Visual comparisons of using CN or AdaIN [3] on CelebA-HQ. (a) Input images. (b) SRN w AdaIN in pre-training phase. (c) SRN w CN in pre-training phase. (d) SRN w AdaIN. (e) SRN w CN.

Figure 6. Visual comparisons of different adversarial losses on CelebA-HQ. (a) Input images. (b) Vanilla global adversarial loss. (c) Context adversarial loss.

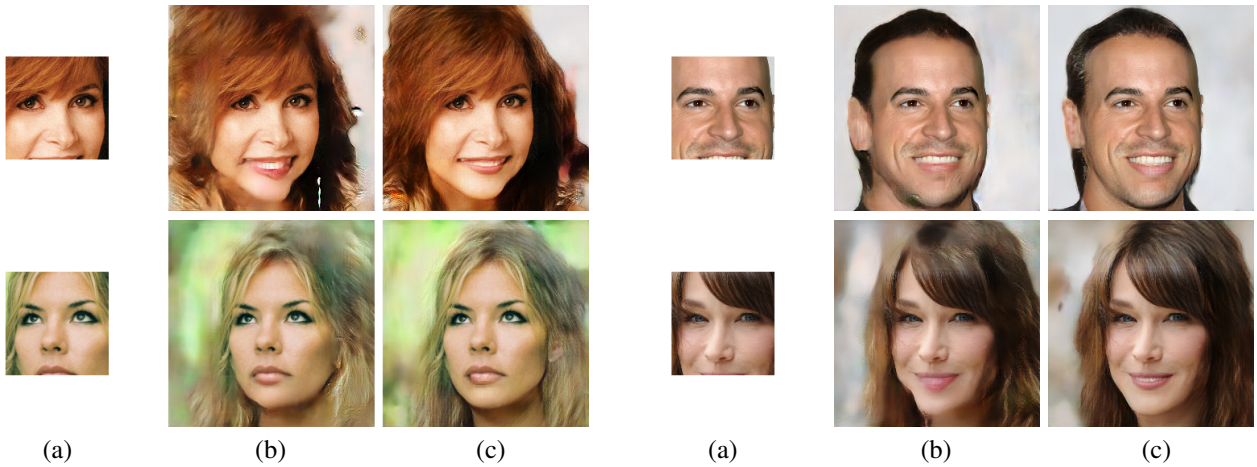


Figure 7. Visual comparisons of using ID-MRF (or not) on CelebA-HQ. (a) Input images. (b) SRN w/o ID-MRF. (c) SRN w ID-MRF.

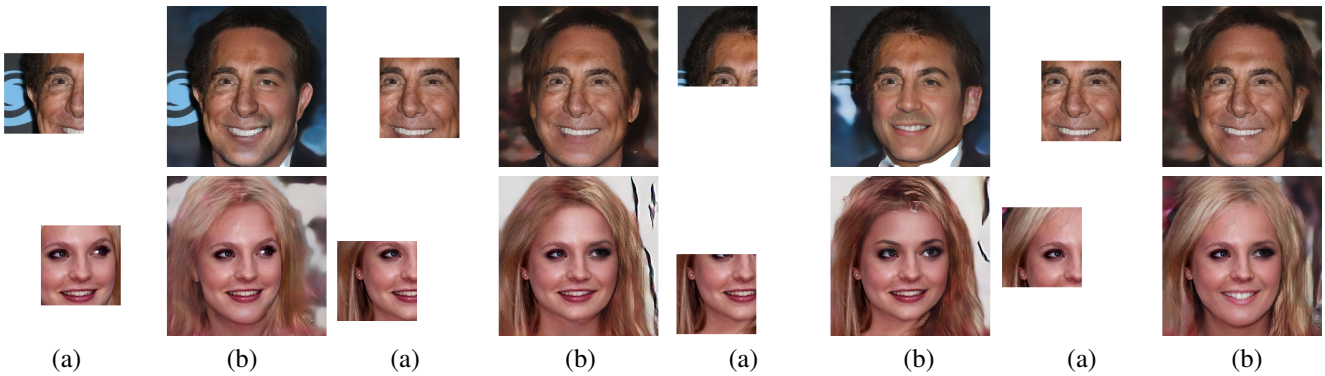


Figure 8. Extrapolation on CelebA-HQ with arbitrary filling margin. (a) Input images. (b) Our results.

Figure 9. Visual comparisons on CUB200. (a) Input images. (b) Results of CA [8]. (c) Our results.

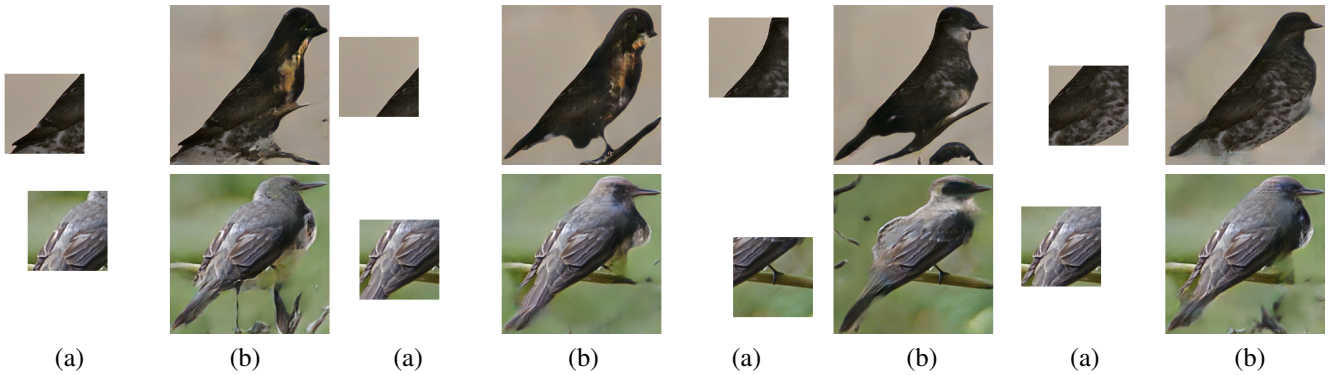
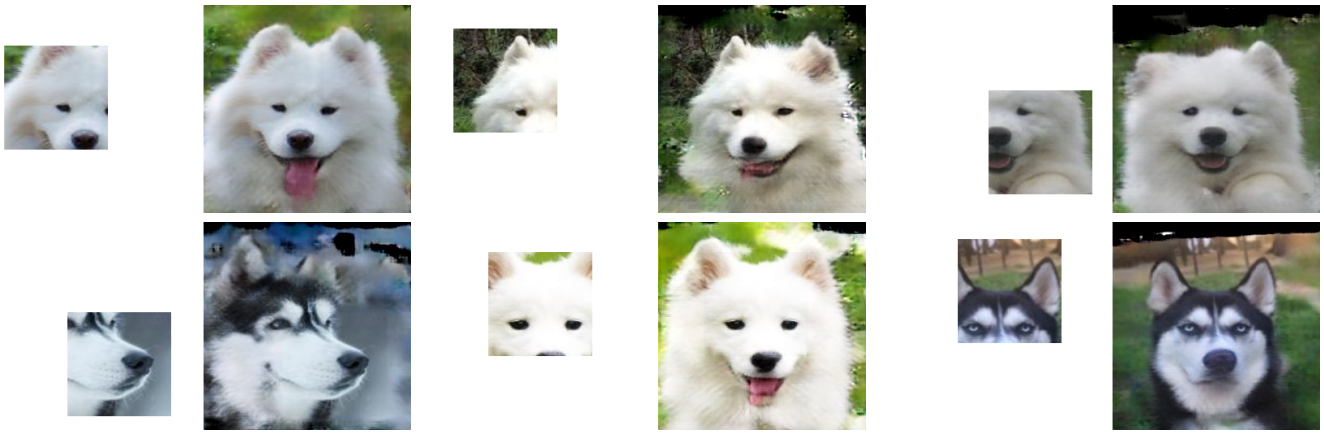


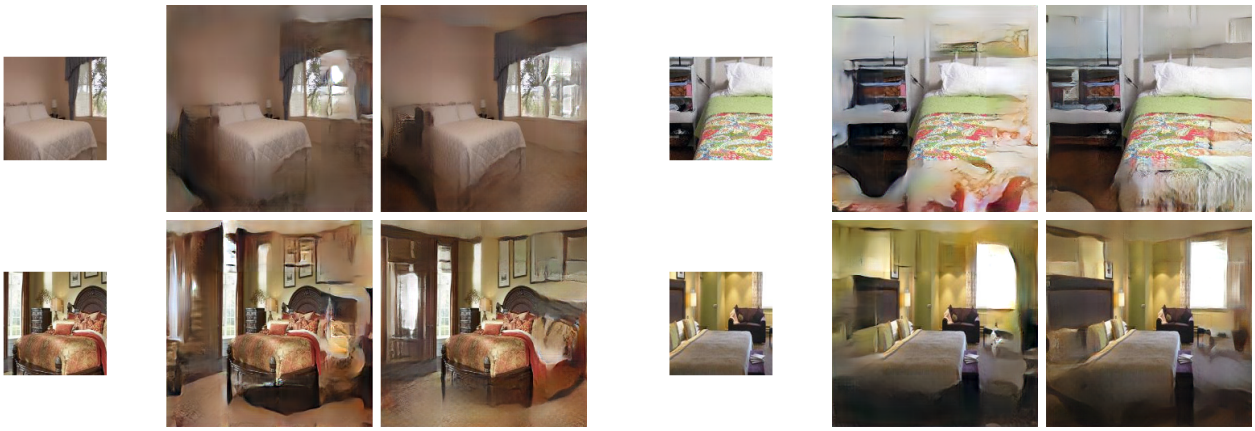
Figure 10. Extrapolation on CUB200 with arbitrary filling margin. (a) Input images. (b) Our results.

Figure 11. Visual comparisons on DeepFashion. (a) Input images. (b) Results of CA [8]. (c) Our results.



(a) (b) (a) (b) (a) (b)

Figure 12. Extrapolation on Dog with arbitrary filling margin. (a) Input images. (b) Our results.



(a) (b) (c) (a) (b) (c)

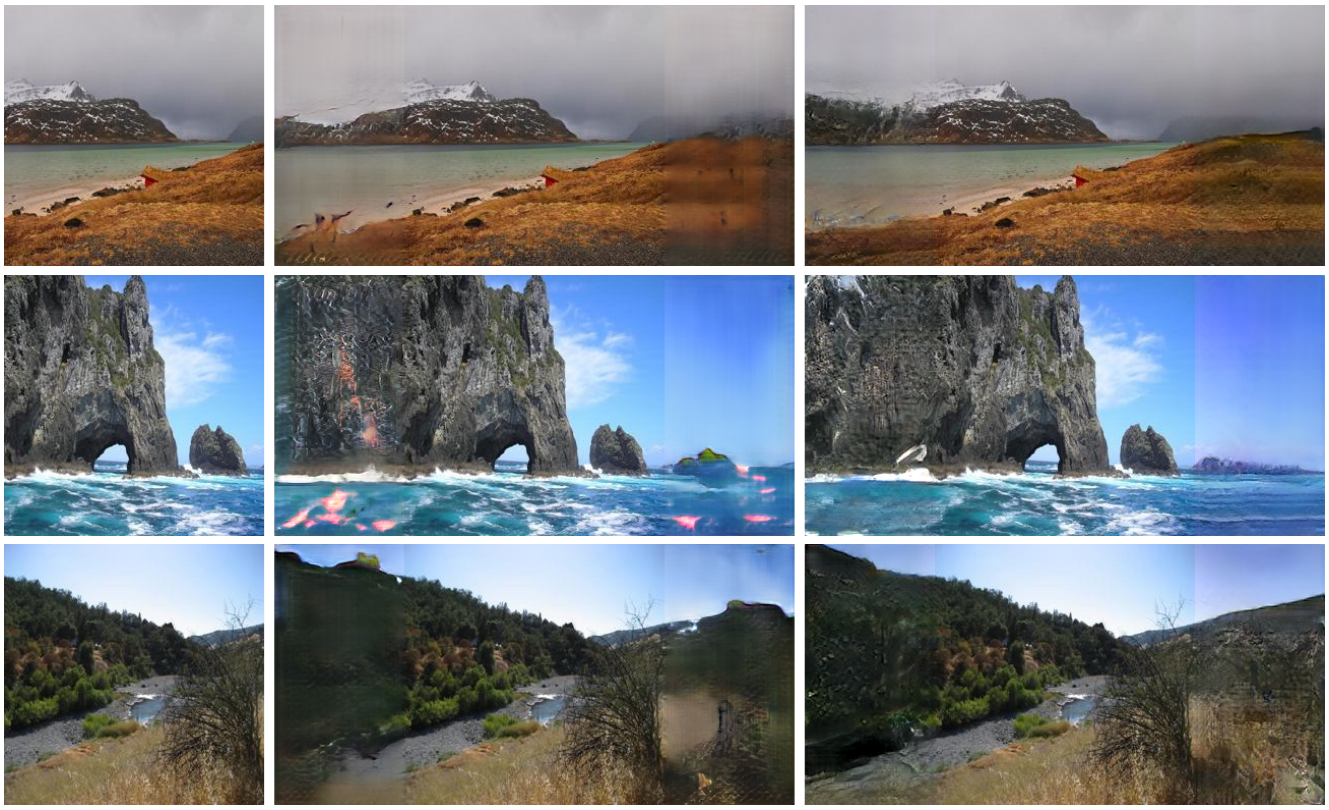
Figure 13. Visual comparisons on Bedroom. (a) Input images. (b) Results of CA [8]. (c) Our results.

(a) (b) (c)

Figure 14. Visual comparisons on Paris street view. (a) Input images. (b) Results of CA [8]. (c) Our results.

(a) (b) (c)

Figure 15. Visual comparisons on Cityscapes. (a) Input images. (b) Results of CA [8]. (c) Our results.



(a) (b) (c)

Figure 16. Visual comparisons on Places2. (a) Input images. (b) Results of CA [8]. (c) Our results.

(a) (b) (c) (a) (b) (c)

Figure 17. Visual comparison of texture synthesis on ETHZ Synthesizability. (a) Input images. (b) Results of CA [8]. (c) Our results.