Patch-Based Image Inpainting with Generative Adversarial Networks

Ugur Demir
Istanbul Technical University
ugurdemir@itu.edu.tr

Gozde Unal
Istanbul Technical University
unalgo@itu.edu.tr

Abstract

Area of image inpainting over relatively large missing regions recently advanced substantially through adaptation of dedicated deep neural networks. However, current network solutions still introduce undesired artifacts and noise to the repaired regions. We present an image inpainting method that is based on the celebrated generative adversarial network (GAN) framework. The proposed PGGAN method includes a discriminator network that combines a global GAN (G-GAN) architecture with a patchGAN approach. PGGAN first shares network layers between G-GAN and patchGAN, then splits paths to produce two adversarial losses that feed the generator network in order to capture both local continuity of image texture and pervasive global features in images. The proposed framework is evaluated extensively, and the results including comparison to recent state-of-the-art demonstrate that it achieves considerable improvements on both visual and quantitative evaluations.

1. Introduction

Image inpainting is a widely used reconstruction technique by advanced photo and video editing applications for repairing damaged images or refilling the missing parts. The aim of the inpainting can be stated as reconstruction of an image without introducing noticeable changes. Although fixing small deteriorations are relatively simple, filling large holes or removing an object from the scene are still challenging due to huge variabilities and complexity in the high dimensional image texture space. We propose a neural network model and a training framework that completes the large blanks in the images. As the damaged area(s) take up large space, hence the loss of information is considerable, the CNN model needs to deal with both local and global harmony and conformity to produce realistic outputs.

Recent advances in generative models show that deep neural networks can synthesize realistic looking images remarkably, in applications such as super-resolution [15, 18, 6], deblurring [28], denoising [39] and inpainting [25, 34, 11, 21]. One of the essential questions about realistic texture synthesis is: how can we measure ”realism” or ”naturalness”? One needs to formulate a yet inexistent formu-

lation or an algorithm that determines precisely whether an image is real or artificially constructed. Primitive objective functions like Euclidean Distance assist in measuring and comparing information on the general structure of the images, however, they tend to converge to the mean of possible intensity values that cause blurry outputs. In order to solve this challenging problem, Goodfellow *et al*. proposed Generative Adversarial Networks (GAN) [7], which is a synthesis model trained based on a comparison of real images with generated outputs. Additionally, a discriminative network is included to classify whether an image comes from a real distribution or a generator network output. During the training, the generative network is scored by an adversarial loss that is calculated by the discriminator network.

Grading a whole image as real or fake can be employed for small images [25], however high resolution synthesis needs to pay more attention to local details along with the global structure [34, 11, 21]. Isola *et al*. introduced the PatchGAN that reformulates the discriminator in the GAN setting to evaluate the local patches from the input [13]. This work showed that PatchGAN improves the quality of the generated images, however it is not yet explored for image inpainting. We design a new discriminator that aggregates the local and global information by combining the global GAN (G-GAN) and PatchGAN approaches for that purpose.

In this paper, we propose an image inpainting architecture with the following contributions:

- Combination of PatchGAN and G-GAN that first shares network layers, later uses split paths with two separate adversarial losses in order to capture both local continuity and holistic features in images;
- Addition of dilated and interpolated convolutions to ResNet [14] in an overall end-to-end training network created for high-resolution image inpainting;
- Analysis of different network components through ablation studies;
- A detailed comparison to latest state-of-the-art inpainting methods.

2. Related works

The idea of **AutoEncoders** (AE) dominated the generative modeling literature in the last decade. Theoretical developments in connecting probabilistic inference with efficient approximate optimization as in Variational AutoEncoders [17] and the intuitive expansion of AEs to Denoising Autoencoders (DAE) [31] constitute building blocks of image synthesis models both in terms of theory and neural network (NN) implementations. Particularly, the design of NN architectures has a crucial effect on texture generation as it shapes the information flow through the layers as desired. The AE framework transforms the input image to an abstract representation, then recover the image from learnt features. To improve gradient flow in backpropagations, skip connections are added to improve synthesis quality in [26]. Residual connections [9, 10, 37, 29, 33] that enhance the gradient flow are also adapted to generative models [14, 13, 39, 8, 19]. Apart from the architectural design, recently introduced components as batch normalization [12], instance normalization [30], dilated convolution [36] and interpolated convolution [24] produce promising effects on the results of image generation process [14, 26, 18, 15, 11].

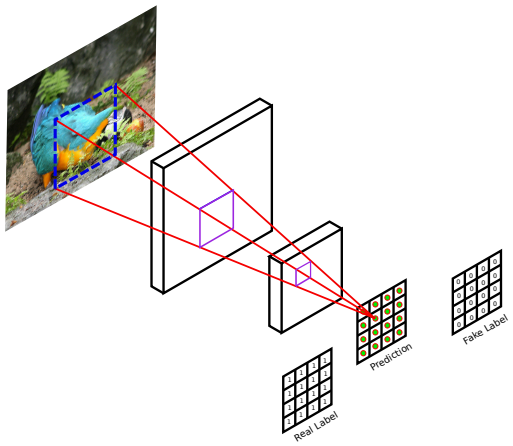


Figure 1: PatchGAN discriminator. Each value of the output matrix represents the probability of whether the corresponding image patch is real or it is artificially generated.

Adversarial training has become a vital step for texture generator Convolutional Neural Networks (CNNs). It provides substantial gradients to drive the generative networks toward producing more realistic images without any human supervision. However, it suffers from unstable discriminator behavior during training which frustrates the generator convergence. Furthermore, the GAN considers images holistically and focuses solely on the realistic image generation rather than generation of an image patch well-matched to the global image. That property of GAN is incompatible with the original goal of the inpainting. Numerous GAN-like architectures have been proposed during the last years to solve those issues to some degree [40, 23, 27, 4, 13].

Recently proposed PatchGAN [13, 20] provides a simple framework that can be adapted to various image generation problems. Instead of grading the whole image, it slides a window over the input and produces a score that indicates whether the patch is real or fake. As the local continuity is preserved, a generative network can reveal more detail from the available context as illustrated in the cover figure which presents some results of the proposed technique. To our

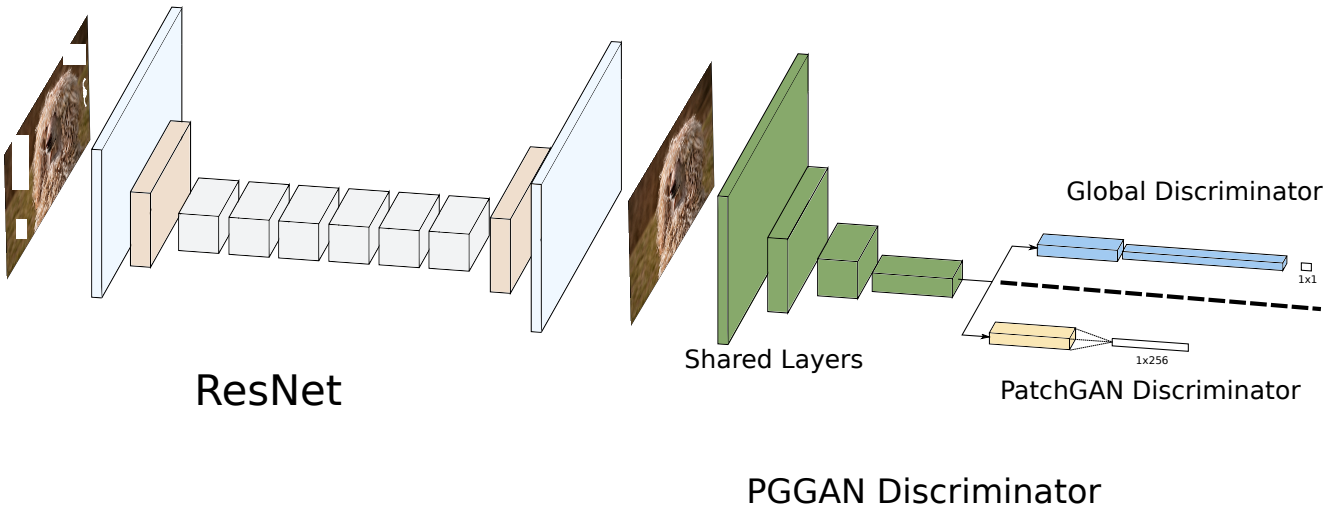


Figure 2: Generative ResNet architecture and PGGAN discriminator which is formed by combining PatchGAN and G-GAN.

knowledge, our work is the first to accommodate PatchGAN approach to work with the inpainting problem.

Inpainting: Early inpainting studies, which worked on a single image, [2, 3, 22, 1] typically created solutions through filling the missing region with texture from similar or closest image areas, hence they suffered from the lack of global structural information.

A pioneering study that incorporated CNNs into the inpainting is proposed by Pathak *et al*. [25]. They developed Context-Encoder (CE) architecture and applied adversarial training [7] to learn features while regressing the missing part of the images. Although the CE had shown promising results, inadequate representation generation skills of an AutoEncoder network in the CE led to substantial amount of implausible results as well.

An importance-weighted context loss that considers closeness to the corrupted region is utilized in [35]. In Yang *et al*. [34], a CE-like network is trained with an adversarial and a Euclidean loss to obtain the global structure of the input. Then, the style transfer method of [20] is used, which forces features of the small patches from the masked area to be close to those of the undamaged region to improve texture details.

Two recent studies on arbitrary region completion [21, 11] add a new discriminator network that considers only the filled region to emphasize the adversarial loss on top of the global GAN discriminator (G-GAN). This additional network, which is called the local discriminator (L-GAN), facilitates exposing the local structural details. Although those works have shown prominent results for the large hole filling problem, their main drawback is the LGAN's emphasis on conditioning to the location of the mask. It is observed that this leads to disharmony between the masked area where the LGAN is interested in and the uncorrupted texture in the unmasked area. The same problem is indicated in [11] and solved by applying post-processing methods to the synthesized image. In [21], LGAN pushes the generative network to produce independent textures that are incompatible with the whole image semantics. This problem is solved by adding an extension network that corrects the imperfections. Our proposed method on the other hand explores every possible local region as well as dependencies among them to exploit local information to the fullest degree.

3. Proposed Method

We introduce a generative CNN model and a training procedure for the arbitrary and large hole filling problem. The generator network takes the corrupted image and tries to reconstruct the repaired image. We utilized the ResNet [14] architecture as our generator model with a few alterations. During the training, we employ the adversarial loss to obtain realistic looking outputs. The key point of our work is the following: we design a novel discriminator network that combines G-GAN structure with PatchGAN approach which we call PGGAN. The proposed network architecture is shown in Figure 2.

3.1. Generator network

The generative ResNet that we compose consists of down-sampling, residual blocks and up-sampling parts using the architectural guidelines introduced in [14]. Down-sampling layers are implemented by using strided convolutions without pooling layers. Residual blocks do not change the width or height of the activation maps. Since our network performs completion operation in an end-to-end man-

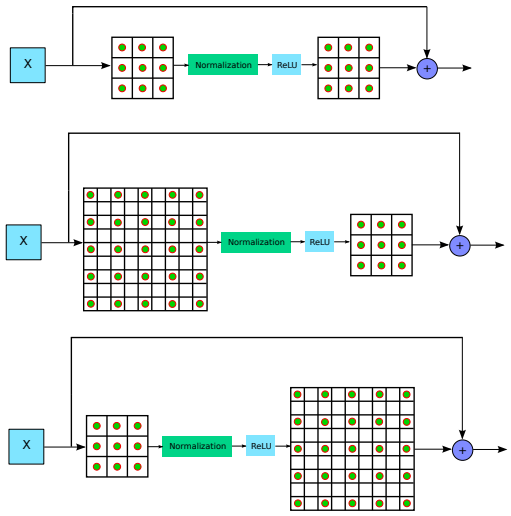


Figure 3: Residual block types. a: standard residual block. b: dilated convolution is placed first. c: Dilated convolution is placed second.

ner, the output must have the same dimension with the input. Thus, in the configuration of all our experiments, the number of down-sampling and up-sampling layers are selected as equal.

Receptive field sizes, which dictate dependency between distant regions, have a critical effect on texture generation. If the amount of sub-sampling is raised to increase the receptive field, the up-sampling part of the generator network will be faced with a more difficult problem that typically leads to low quality or blurry outputs. The **dilated convolution** operation is utilized in [36] in order to increase the receptive field size without applying sub-sampling or adding excessive amount of convolution layers. Dilated convolution spreads out the convolution weights to over a wider area to expand the receptive field size significantly without increasing the number of parameters. This was first used by [11] for inpainting. We also investigate the effect of the dilated convolution for texture synthesis problem. Three different residual block types are used in our experiments as shown in the Figure 3. First residual block which is called type-a contains only two standard convolutions, normalization, activation and a residual connection. Other types introduce dilated convolution. Type-b block places dilation before the normalization layer and type-c block uses dilation after the activation layer. While dilation is used in our network, dilation parameter is increased by a factor of two in each residual block starting from one.

Interpolated convolution is proposed by Odena *et al.* [24] to overcome the well-known checkerboard artifacts during the up-sampling operation caused by the transposed convolution (also known as deconvolution). Instead of learning a direct mapping from a low resolution feature map to high resolution, the input is resized to the desired size and then the convolution operation is applied. Figure 5 shows how the interpolated convolution affects the image synthesis elegantly.

3.2. Discriminator network

Discriminator network D takes the generated and real images and aims to distinguish them while the generator network G makes an effort to fool it. As long as D successfully classifies its input, G benefits from the gradient provided by the D network via its adversarial loss.

We achieve our goal of obtaining an objective value that measures the quality of the image as a whole as well as the consistency in local details through our PGGAN approach depicted in Figure 2. Rather than training two separate networks simultaneously, we design a weight sharing architecture at the first few layers so that they learn common low level visual features. After a certain layer, they are split into two pathways. The first path ends up with a binary output which decides whether the whole image is real or not. The second path evaluates the local texture details similar to the PatchGAN. Fully connected layers are added at the end of the second path of our discriminator network to reveal full dependency across the local patches. The overall architecture hence provides an objective evaluation of the naturalness of the whole image as well as the coherence of the local texture.

3.3. Objective function

At the training stage, we use a combination of three loss functions. They are optimized jointly via backpropagation using Adam optimizer [16]. We describe each loss function briefly as follows.

Reconstruction loss computes the pixel-wise L1 distance between the synthesized image and the ground truth. Even though it forces the network to produce a blurry output, it guides the network to roughly predict texture colors and low frequency details. It is defined as:

$$\mathcal{L}_{rec} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{WHC}||y - x||_1 \tag{1}$$

where N is the number of samples, x is the ground truth, y is the generated output image, W, H, C are width, height, and channel size of the images, respectively.

Adversarial loss is computed by the both paths of PGGAN discriminator network D that is introduced in the training phase. Generator G and D are trained simultaneously by solving $\arg\min_G \max_D L_{GAN}(G, D)$:

$$\begin{aligned}\mathcal{L}_{GAN}(G, D) &= \mathbb{E}_{x\sim p(x)}[\log D(x)] \\ &+ \mathbb{E}_{y\sim p_G(\tilde{x})}[\log(1 - D(G(\tilde{x})))]\end{aligned} \tag{2}$$

where $\tilde{x}$ is the corrupted image.

Joint loss function defines the objective used in the training phase. Each component of the loss function is governed by a coefficient λ:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{g\_adv} + \lambda_3 \mathcal{L}_{p\_adv} \quad (3)$$

where $\mathcal{L}_{g\_adv}$ and $\mathcal{L}_{p\_adv}$ refer to $\mathcal{L}_{GAN}$ in Equation 2 corresponding to two output paths of the PGGAN (see Figure 3). We update the generator parameters by joint loss $\mathcal{L}$, unshared G-GAN layers by $\mathcal{L}_{g\_adv}$, unshared P-GAN layers by $\mathcal{L}_{p\_adv}$ and shared layers by $\mathcal{L}_{g\_adv} + \mathcal{L}_{p\_adv}$.

4. Results

In this section, we evaluate the performance of our method and compare PGGAN with the recent inpainting methods through ablation studies, quantitative measurements, perceptual scores and visual evaluations.

4.1. Datasets

Paris Street View [5] has 14900 training images and 100 test images which is collected from Paris. Comparisons and our ablation study are mostly performed on this dataset.

Google Street View [38] consist of 62058 high quality images. It is divided into 10 parts. We use the first and tenth parts as the testing set, the ninth part for validation, and the rest of the parts are included in the training set. In this way, 46200 images are used for training.

Places [41] is one of the largest dataset for visual tasks that has nearly 8 million training images. Since there is considerable amount of data in the set, it is helpful for testing generalizability of out networks.

4.2. Training details and implementation

All of the experimental setup is implemented using Pytorch [1] with GPU support. Our networks are trained separately on four NVIDIA™ Tesla P100 and a K40 graphic cards.

In order to obtain comparable results from our generative ResNet implementation, we use 3 subsampling blocks when type-a blocks are used. If dilated convolution is used in the residual blocks, subsampling is set to two since dilation parameter makes it possible to reach wider regions without subsampling.

While training our networks with PGGAN discriminator, we set $\lambda_1 = 0.995$, $\lambda_2 = 0.0025$ and $\lambda_3 = 0.0025$ in Equation 3.

4.3. Ablation study

In order to analyze effects of different components introduced, we perform several experiments by changing parameters one at a time. First, we compare the different discriminator architectures on the same generator network ResNet. All the networks are trained until no significant change occurs. Figure 4 shows sample results. It can be observed for instance in the last column, the window details are reconstructed differently across the methods. As expected, the G-GAN discriminator aids in completing only the coarse image structures. PatchGAN demonstrates significant improvement compared to G-GAN but reconstructed images still have a sign of global misconception. PGGAN blends both local and global structure and provides visually more plausible results.

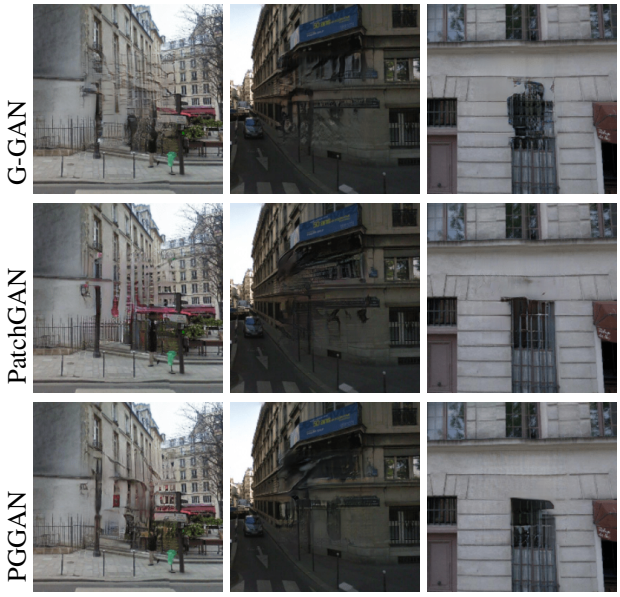


Figure 4: Results are obtained by training the same generator network with different discriminator architectures.

Along with the discriminator design, another important factor for image synthesis is the layers used in generator network models. In this study, we prefer interpolated convolution rather than transposed convolution because it provides smooth outputs. To illustrate the impact of the interpolated convolution, we tested the same PGGAN except the upsampling layer as demonstrated in Figure 5.

Impact of the interpolated convolution can be clearly observed by zooming to the results of Figure 5. It clears the noise also known as checkerboard artifacts caused by the transposed convolution. However, there are examples that have more consistent structures obtained by the transposed convolution (e.g. see the first column of the figure). These layers have distinct characteristics that each direct the generator to a different point in the solution space. Both layers should be analyzed further which is not in the scope of this study.

[1] http://pytorch.org/

Figure 5: Sample outputs; top: transposed convolution (tconv) and bottom: interpolated convolution (iconv) [24].

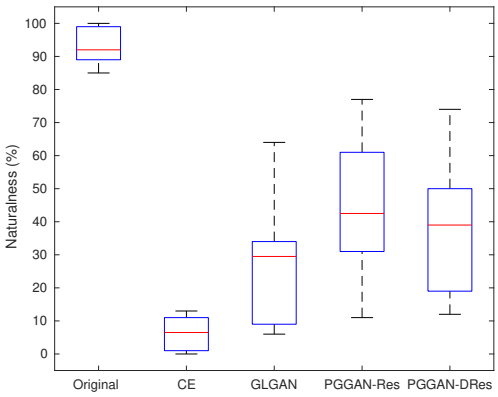


Figure 6: Perceptual comparison of Paris [5] images inpainted by different approaches.

4.4. Comparative evaluation

We compare our PGGAN with ResNet (PGGAN-Res) and PGGAN with ResNet-Dilated convolution (PGGAN-DRes) to three current inpainting methods: (i) CE-Context-Encoder is adapted from [25] to work with 256x256 images where full images are reconstructed; (ii) GLGAN [11] over 256x256 images; (iii) Neural Patch Synthesis (NPS) [34] over 512x512 images.

Speed: As PGGAN and GLGAN are both end-to-end texture generators, their computation times are similar on the order of miliseconds. On the other hand, NPS approach takes several seconds due to their local texture constraint.

PSNR and **SSIM** [32] are the two mostly used evaluation criteria among the image generation community although it is known that they are not sufficient for quality assessment. Nonetheless, in order to quantitatively compare our method with the current works, we report PSNR, SSIM, mean L1, and mean L2 loss in Table 1 and Table 2 for 256x256 and 512x512 images, respectively.

Method	L1 Loss	L2 Loss	psnr(dB)	ssim
CE [25]	6.21	1.34	18.12	0.838
GLGAN[11]	5.82	2.33	18.28	0.863
PGGAN-DRes	**5.54**	**1.19**	**19.03**	**0.866**
PGGAN-Res	5.46	1.2	18.92	0.865

Table 1: Performance comparison on 256x256 images from Paris Street View evaluation set.

Method	L1 Loss	L2 Loss	psnr(dB)	ssim
NPS[34]	10.01	2.21	18.0	-
PGGAN-DRes	**5.42**	**1.16**	**18.9**	0.884

Table 2: Comparison between NPS and our DRes-PGGAN with 512x512 Paris Street View images.

PGGAN achieves an improvement in all measures for both 512x512 and 256x256 images. These results are also supported by perceptual and visual evaluations as presented next.

4.5. Perceptual evaluation

We perform perceptual evaluation among PGGAN-Res, PGGAN-DRes, CE and GLGAN. 12 voters from our laboratory scored naturalness (as natural/not natural) of the original images and inpainting results of the methods. Overall each tester evaluated randomly sorted and blinded 500 images (5 x 100 images of the Paris Street View validation set). Figure 6 shows the boxplot of the percent naturalness score accumulated over users for each method.

Results indicate that CE presented for 128x128 images has low performance on the 256x256 test images as also reported in [25]. Rest of the methods performed similarly however, slightly better scores for PGGAN were obtained. This suggests that further emphasis of local coherence along with global structure can help to generate more plausible textures.

4.6. Visual results

We compare visual performance of PGGAN, NPS, and GLGAN on the common Paris Street View dataset. Figures 7 and 8 show the results for images of size 256x256 and 512x512 respectively. Some fail case results can be seen in Figure 9. Results from Places and Google Street View datasets[2] are shown in Figures 10 and 11.

[2]See supplementary materials for extensive results.

Figure 7: Visual comparison on 256x256 Paris Street View Dataset [5].

5. Conclusion

The image inpainting results in this paper suggest that low-level merging then high-level splitting a patch-based technique such as PatchGAN with a traditional GAN network can aid in acquiring local continuity of image texture while conforming to the holistic nature of the images. This merger produces visually and quantitatively better results than the current inpainting methods. However, the inpainting problem which is tightly coupled to the generative modeling problem is still open to further progress.

Figure 8: Visual comparison between PGGAN-RES and NPS [34] on 512x512 Paris Street View Dataset [5].

Figure 9: Non-cherry picked results from PGGAN-DRes.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24:1–24:11, July 2009. 3

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 3

[3] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13/9:1200–1212, September 2004. 3

[4] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1486–1494, Cambridge, MA, USA, 2015. MIT Press. 2

[5] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM Trans. Graph.*, 31(4):101:1–101:9, July 2012. 5, 6, 7, 8

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. 1

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 2, 3

[8] Y. Han, J. J. Yoo, and J. C. Ye. Deep residual learning for

Figure 10: Sample outputs of PGGAN-DRes on Places dataset [41].

Figure 11: Sample outputs of PGGAN-DRes on Google Street View dataset [38].

compressed sensing ct reconstruction via persistent homology analysis. *CoRR*, abs/1611.06391, 2016. 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. 2

[10] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 630–645, 2016. 2

[11] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4), 2017. 1, 2, 3, 4, 6, 7

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. R. Bach and D. M. Blei, editors, *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456, 2015. 2

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016. 2

[14] J. Johnson, A. Alahi, and L. Fei-Fei. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*, pages 694–711. Springer International Publishing, Cham, 2016. 2, 3

[15] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1646–1654, 2016. 1, 2

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 2

[18] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. 1, 2

[19] L. Lettry, K. Vanhoey, and L. V. Gool. DARN: a deep adversial residual network for intrinsic image decomposition. *CoRR*, abs/1612.07899, 2016. 2

[20] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2479–2486, 2016. 2, 3

[21] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 3

[22] Y. Liu and V. Caselles. Exemplar-based image inpainting using multiscale graph cuts. *Trans. Img. Proc.*, 22(5):1699–1711, May 2013. 3

[23] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. 2017. 2

[24] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 2, 4, 6

[25] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros. Context encoders:feature learning by inpainting. In *CVPR*, 2016. 1, 2, 3, 6, 7

[26] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 2

[27] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *International Conference on Learning Representations (ICLR)*. 2016. 2

[28] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring for hand-held cameras. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[29] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016. 2

[30] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 2

[31] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, Dec. 2010. 2

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, Apr. 2004. 6

[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 2

[34] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. *arXiv preprint arXiv:1611.09969*, 2016. 1, 2, 3, 6, 8

[35] R. A. Yeh*, C. Chen*, T. Y. Lim, S. A. G., M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. * equal contribution. 3

[36] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015. 2, 4

[37] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016. 2

[38] A. Zamir and M. Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2014. 5, 9

[39] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *CoRR*, abs/1608.03981, 2016. 1, 2

[40] J. J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016. 2

[41] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 5, 9

Supplementary Materials: Patch-Based Image Inpainting with Generative Adversarial Networks

1. Additional visual results

Following figures show the visual results obtained by the proposed PGGAN algorithm. Input images are taken from ImageNet[1], Google Street View[2] and Places2[3] datasets.

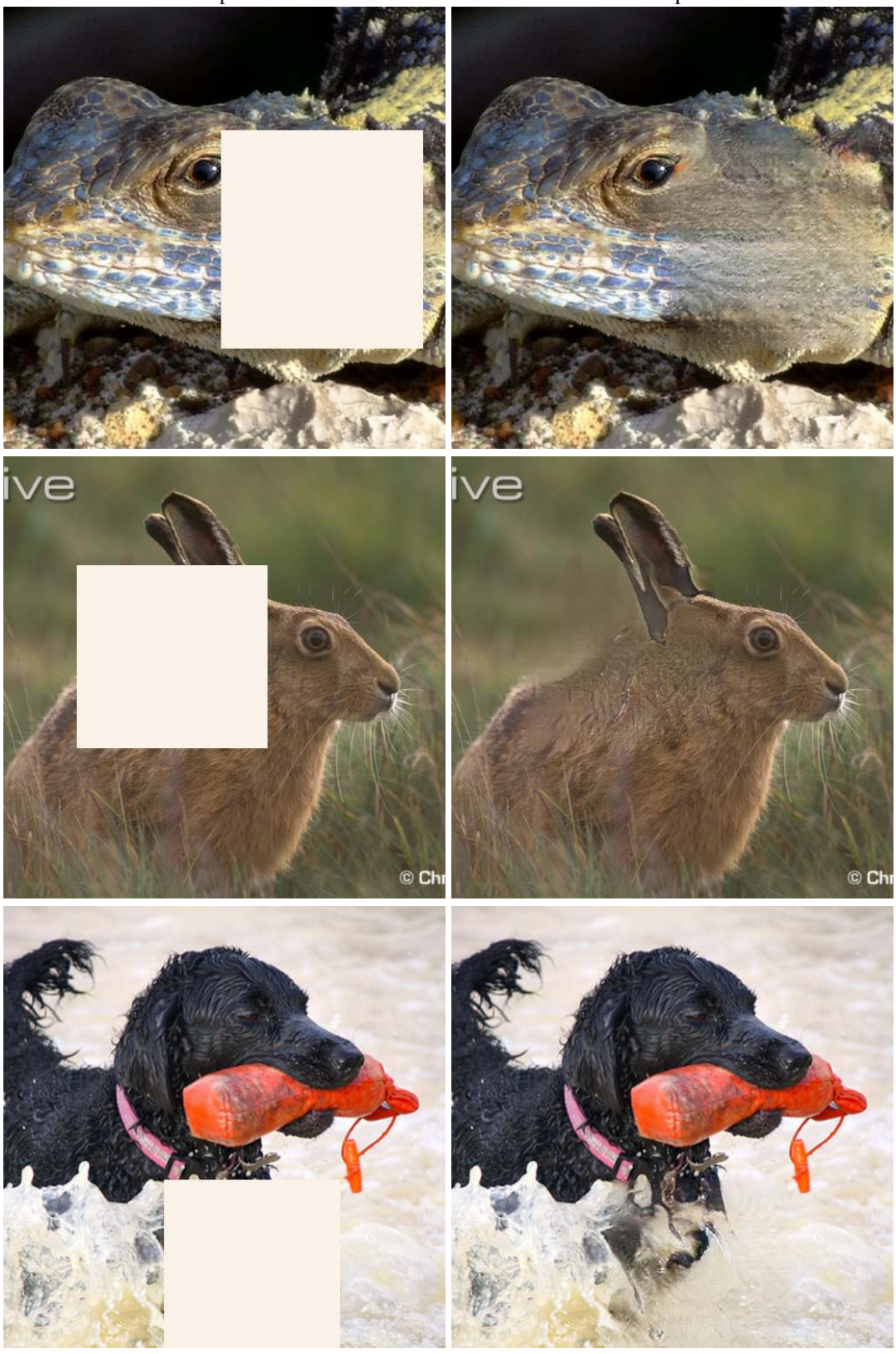
1.1. ImageNet

We perform high resolution inpainting experiments on ImageNet dataset. Input images are scaled to 512x512 and randomly located regions are cropped. Our model can successfully fill the blank areas as demonstrated in following figures.

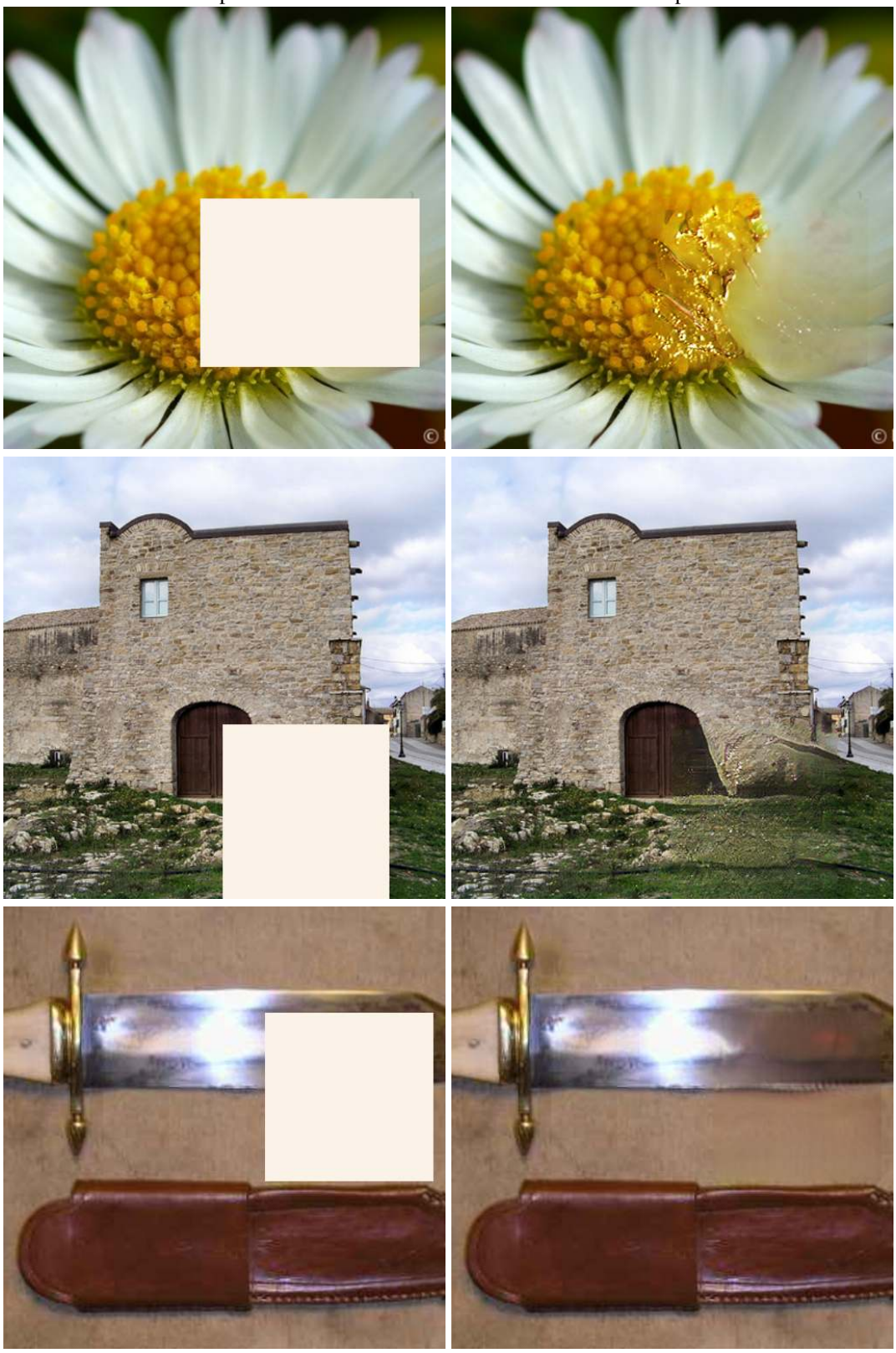
Input | Output

[1] http://image-net.org
[2] http://crcv.ucf.edu/data/GMCP_Geolocalization
[3] http://places2.csail.mit.edu

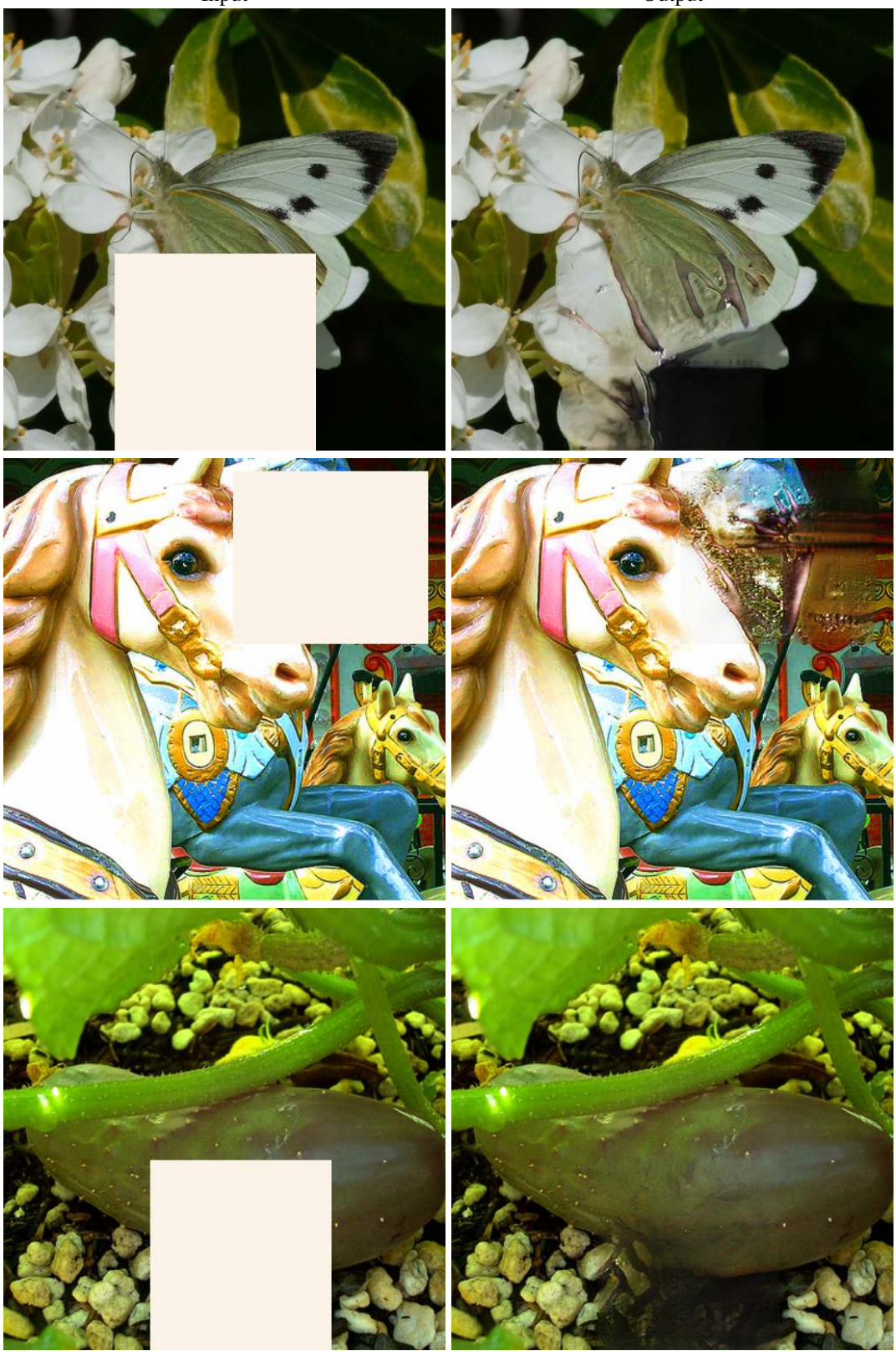
Input Output



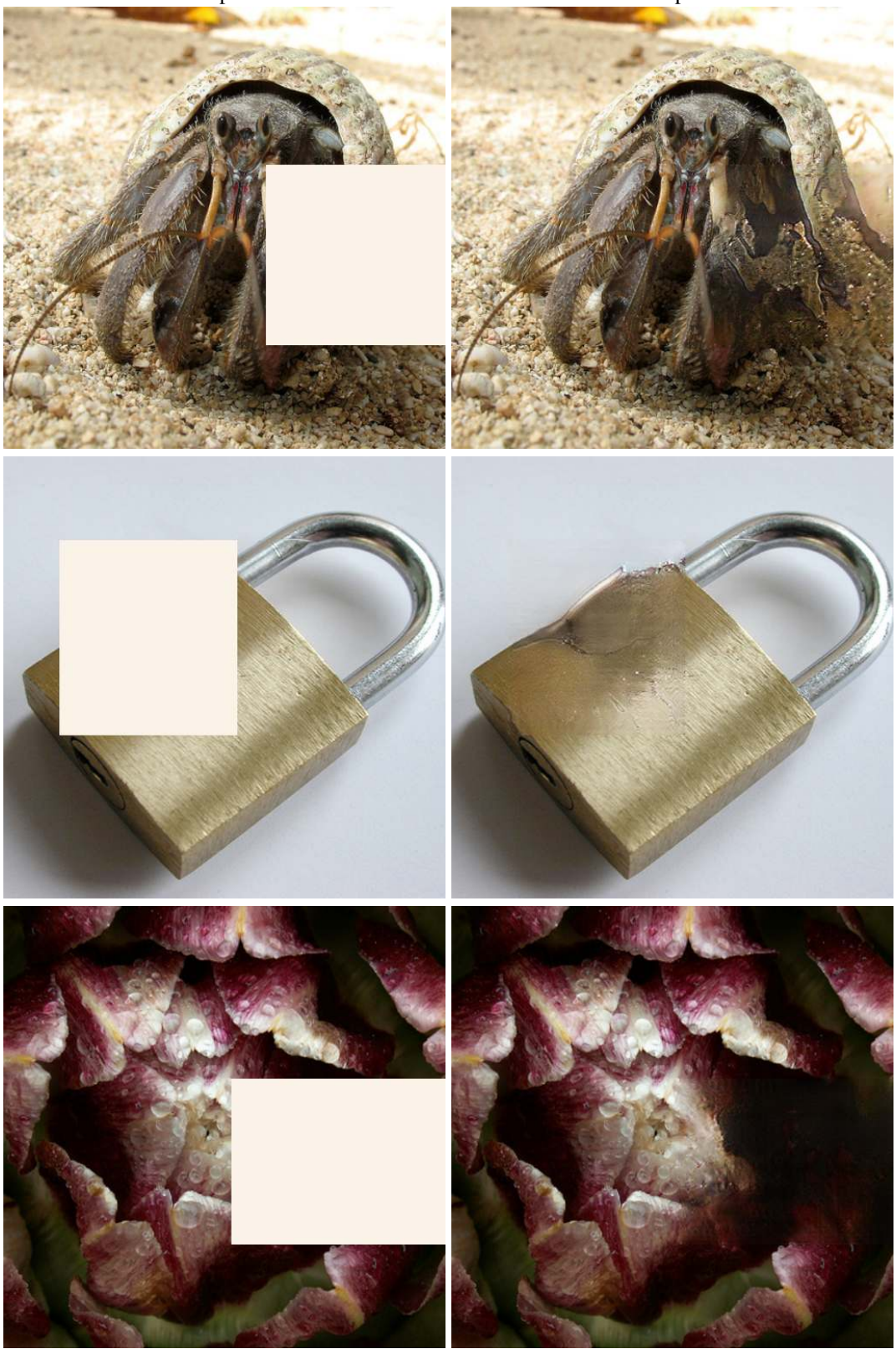
Input Output



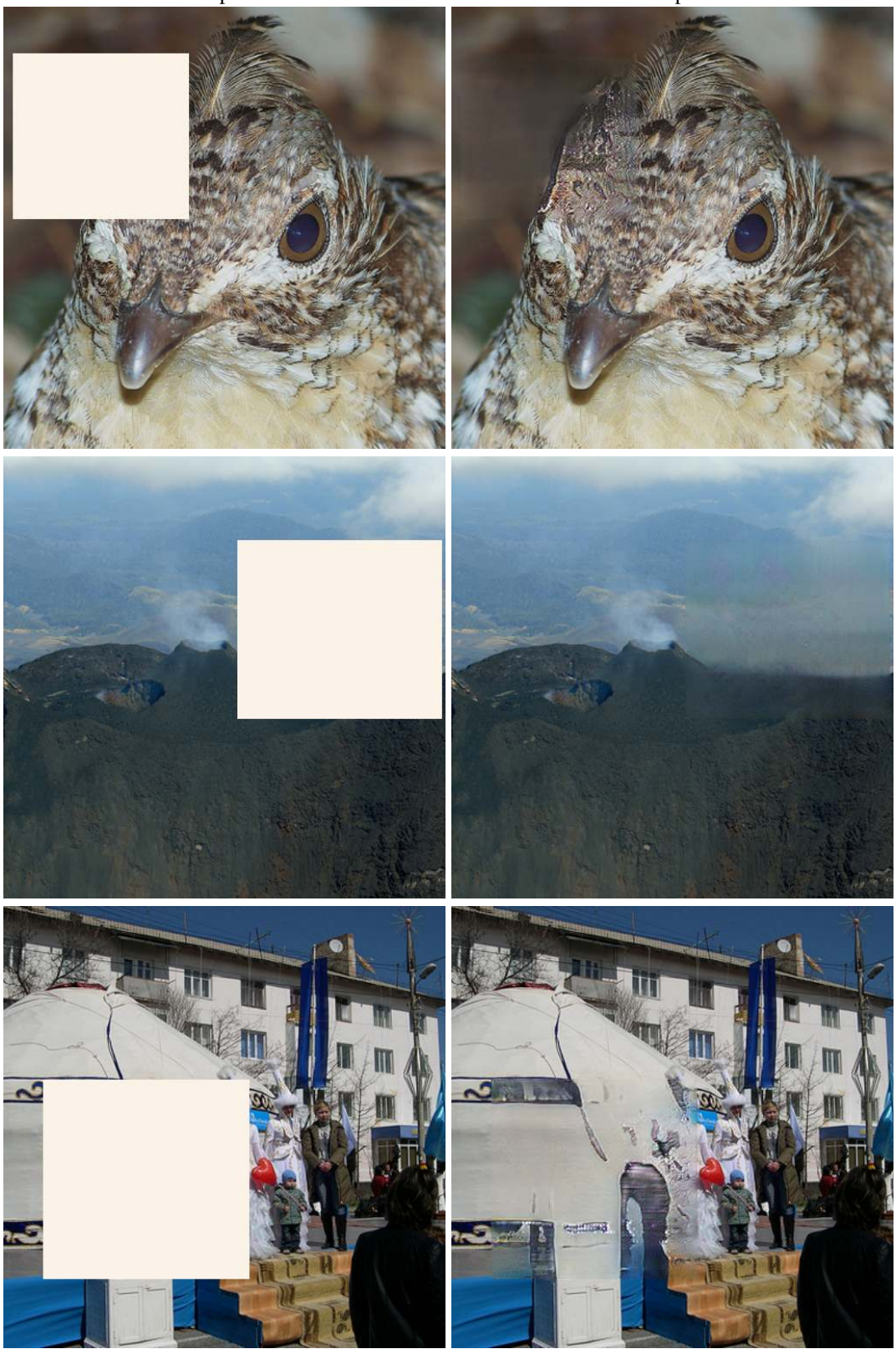
Input Output



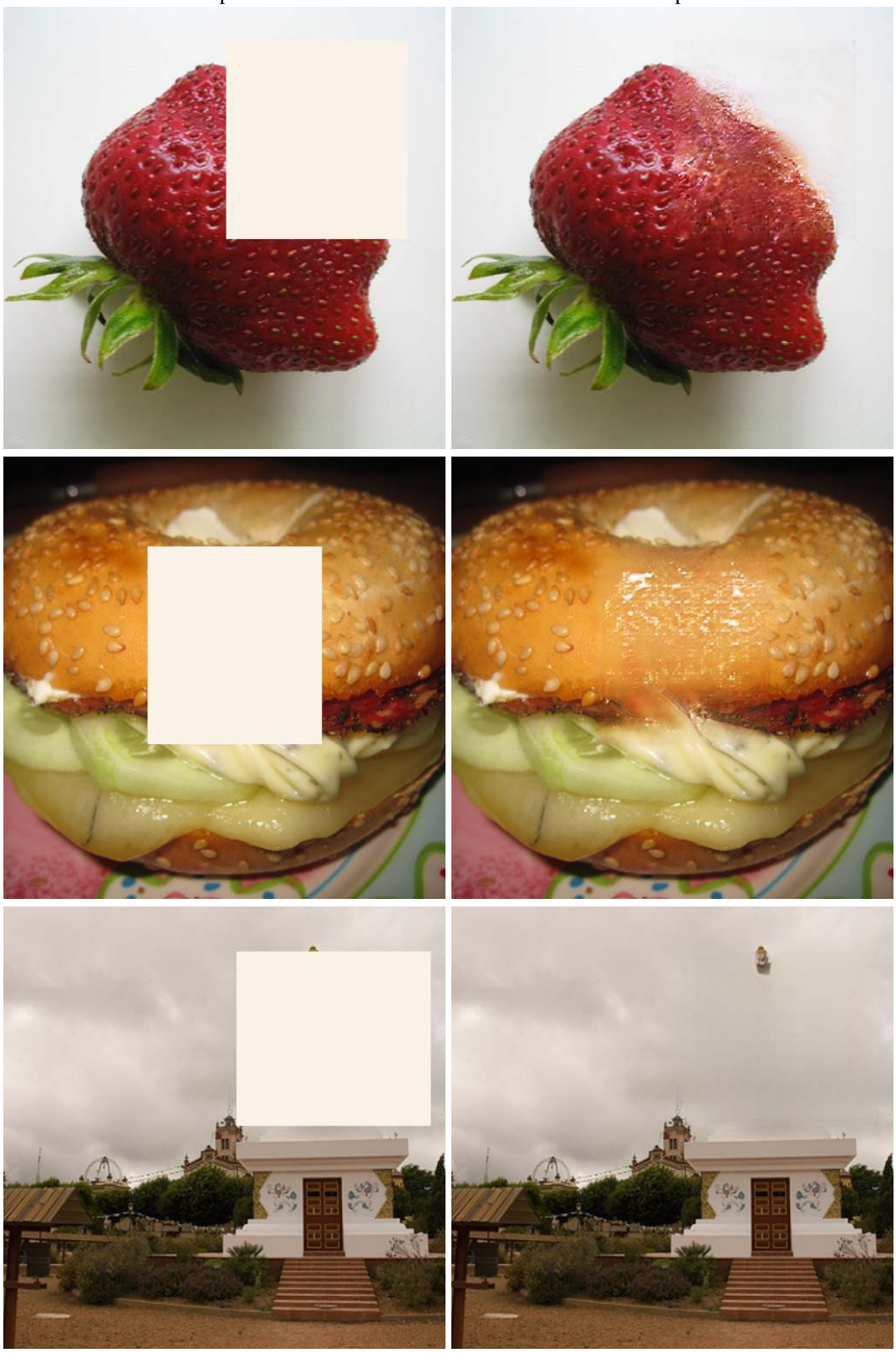
Input Output

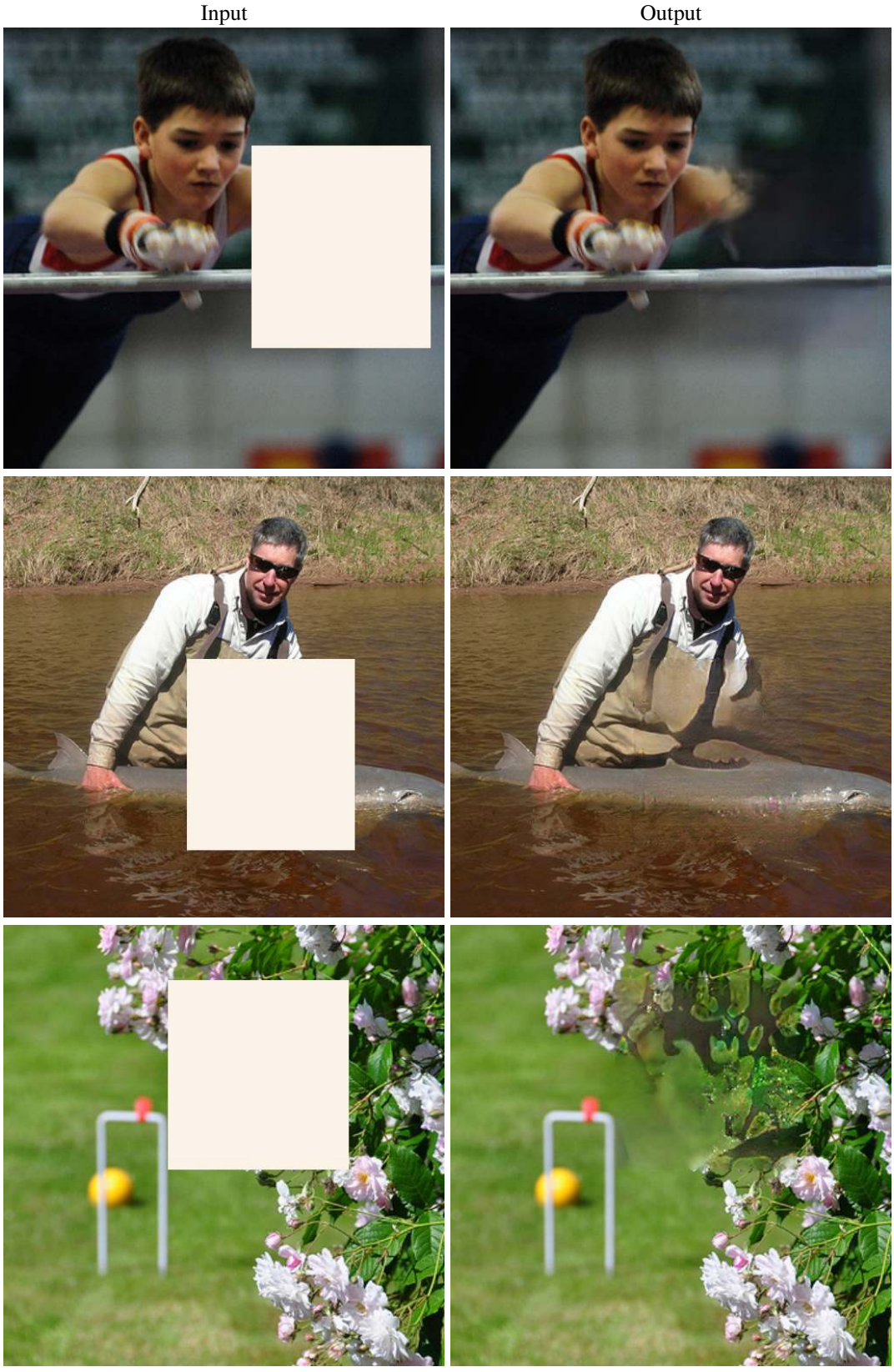


Input Output

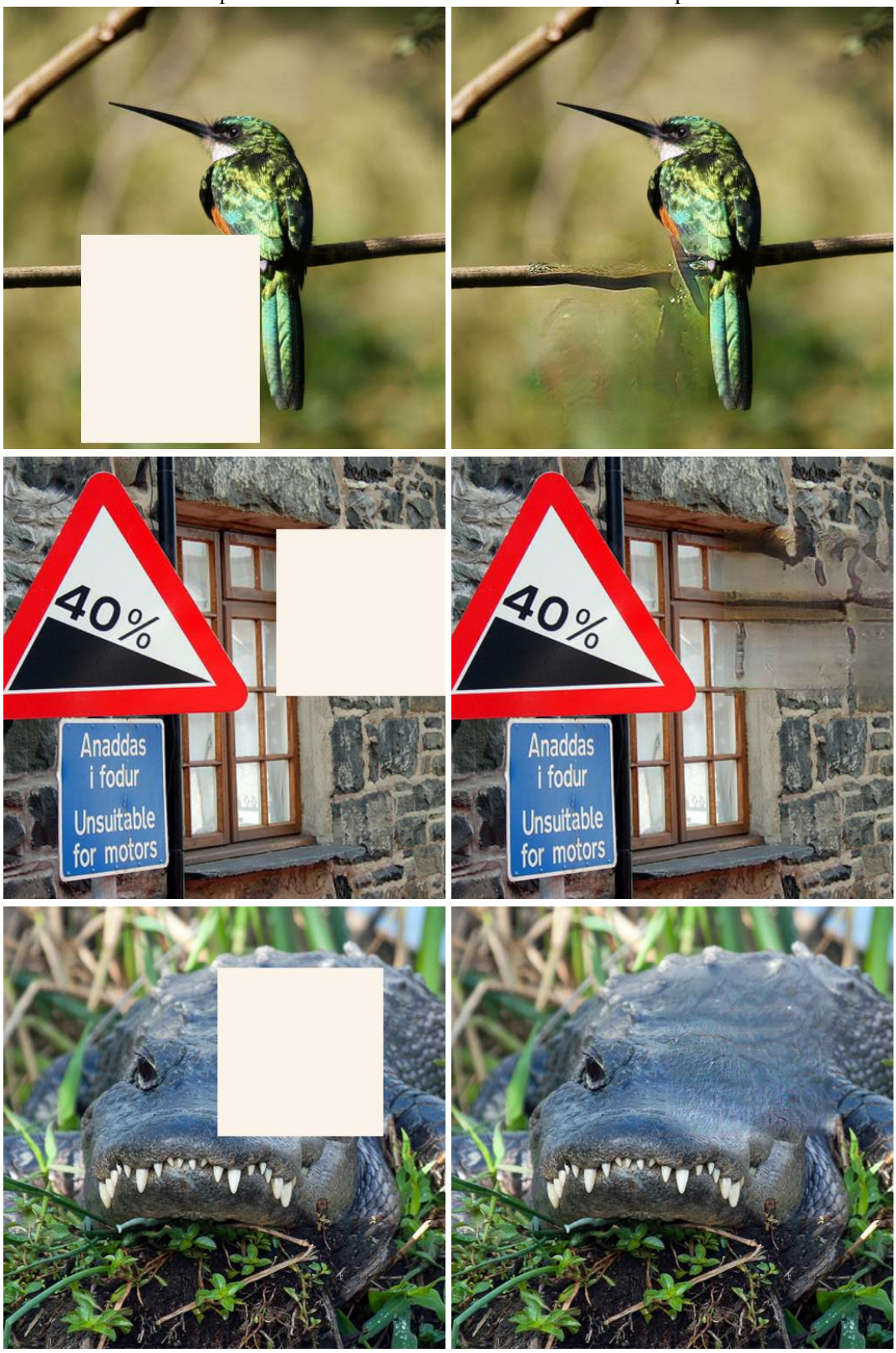


Input Output




Input
Output

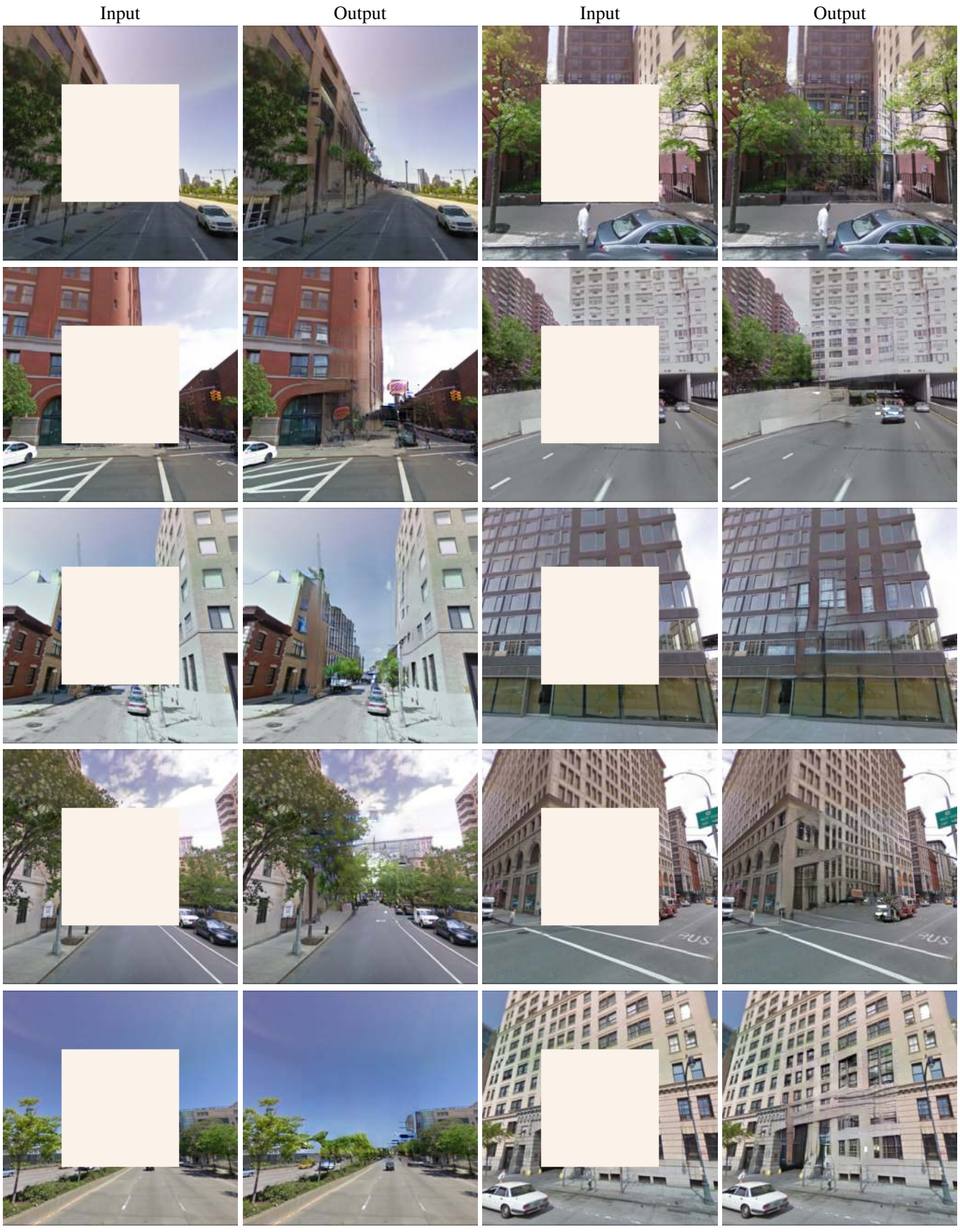
Input Output



1.2. Google Street View

Images from the Google Street View dataset are scaled to 256x256. 128x128 sized center patches are extracted from inputs. Our network reconstructs whole images without using the mask location.


Input
Output
Input
Output

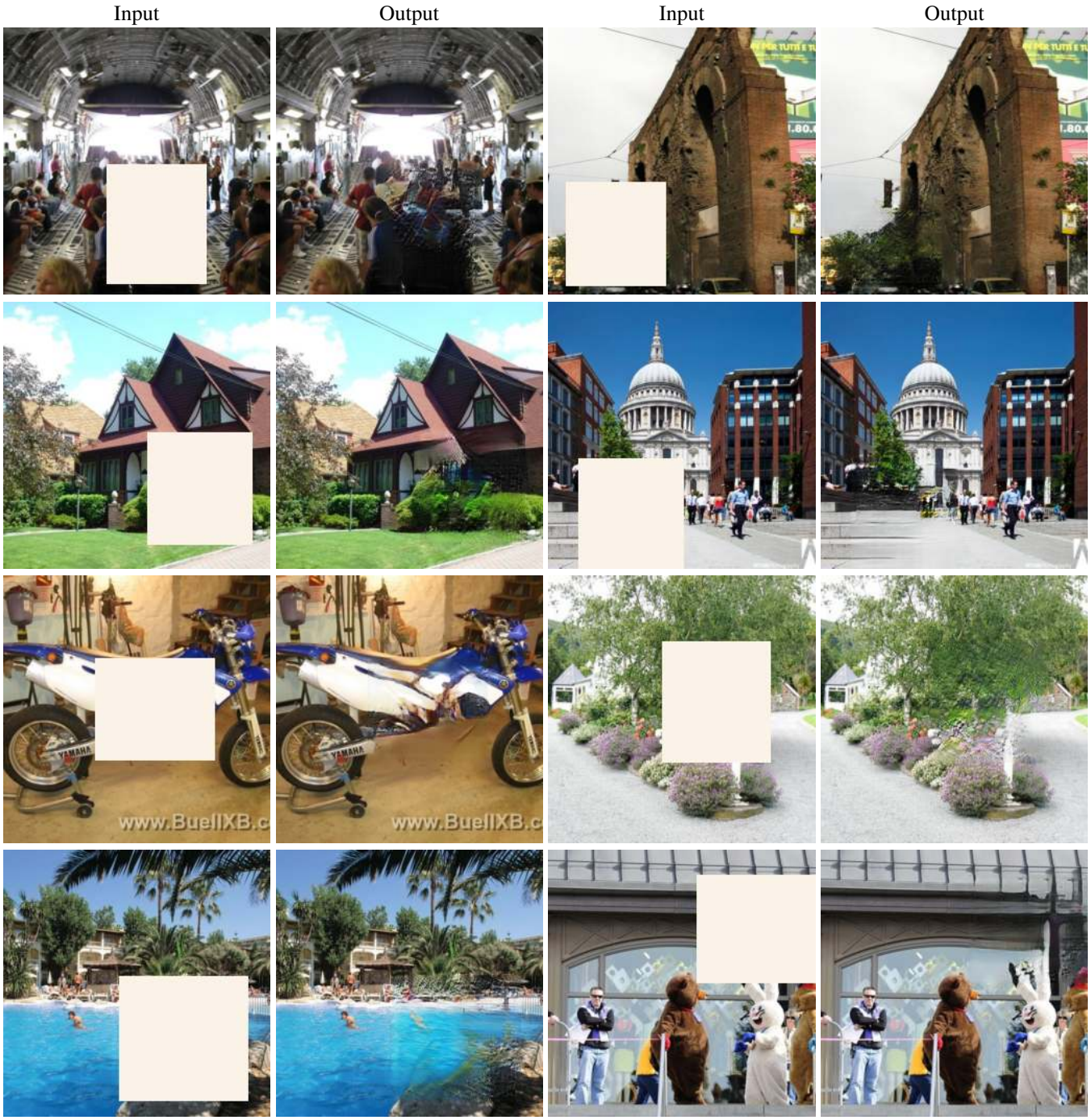

Input
Output
Input
Output

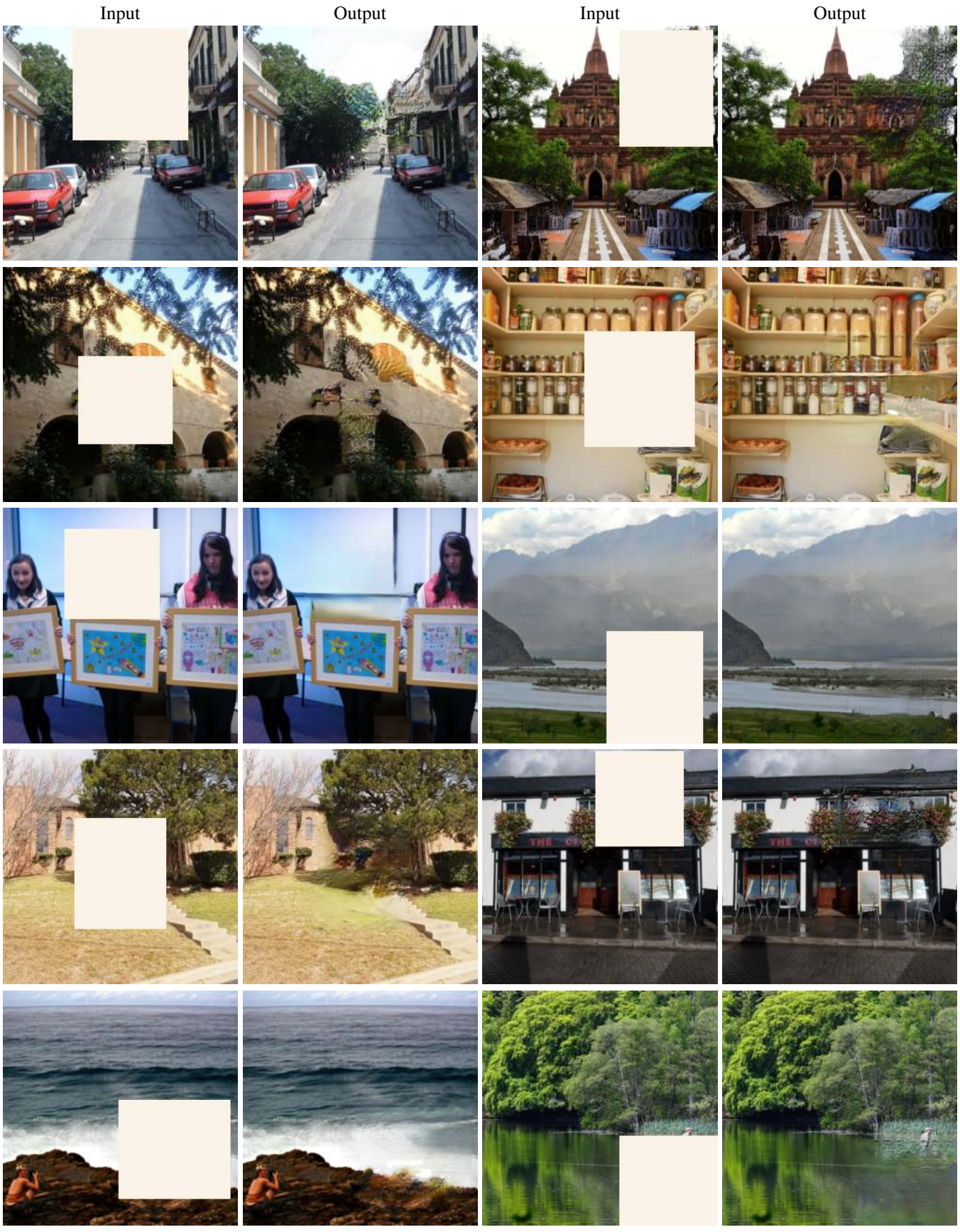

Input
Output
Input
Output

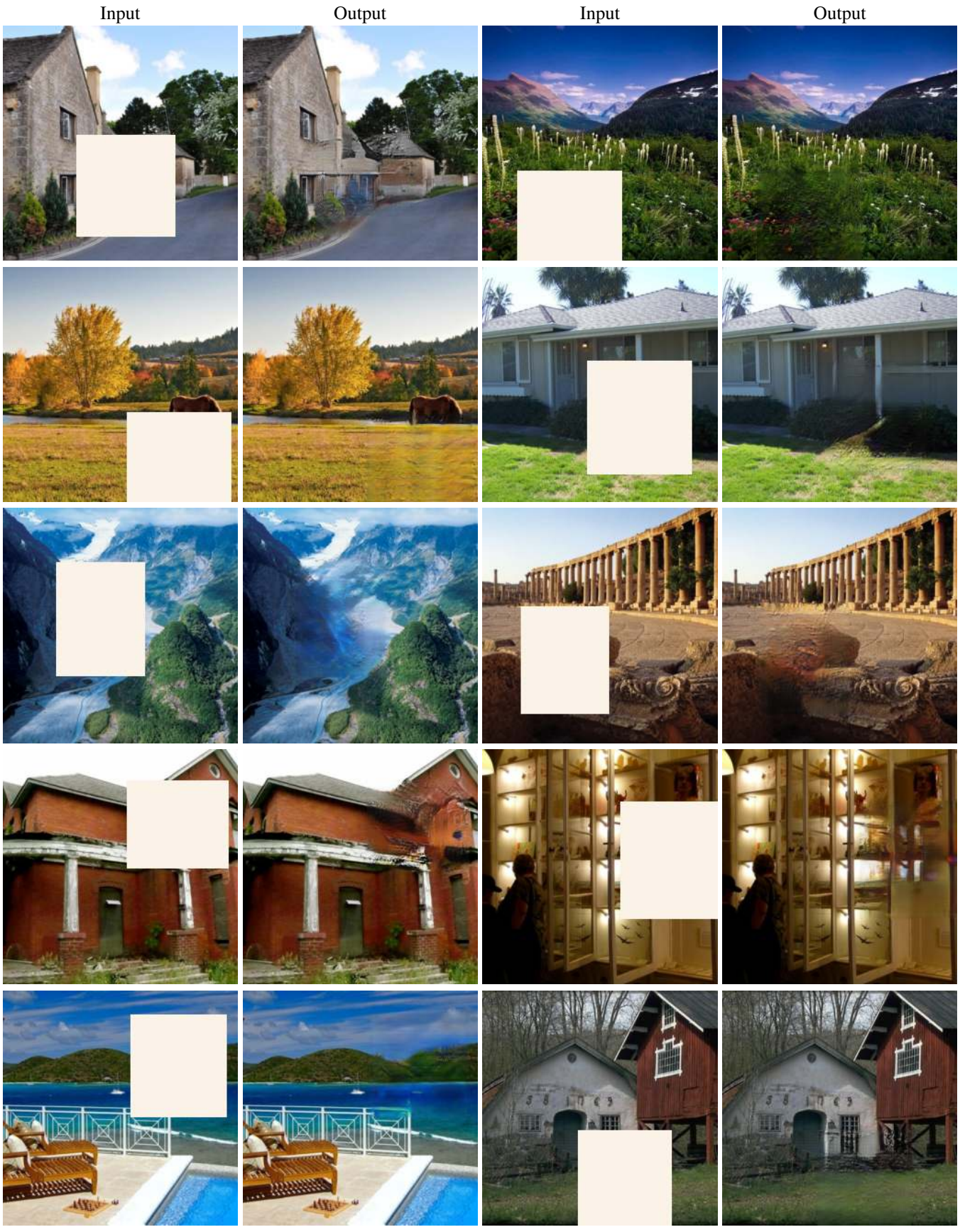

Input
Output
Input
Output

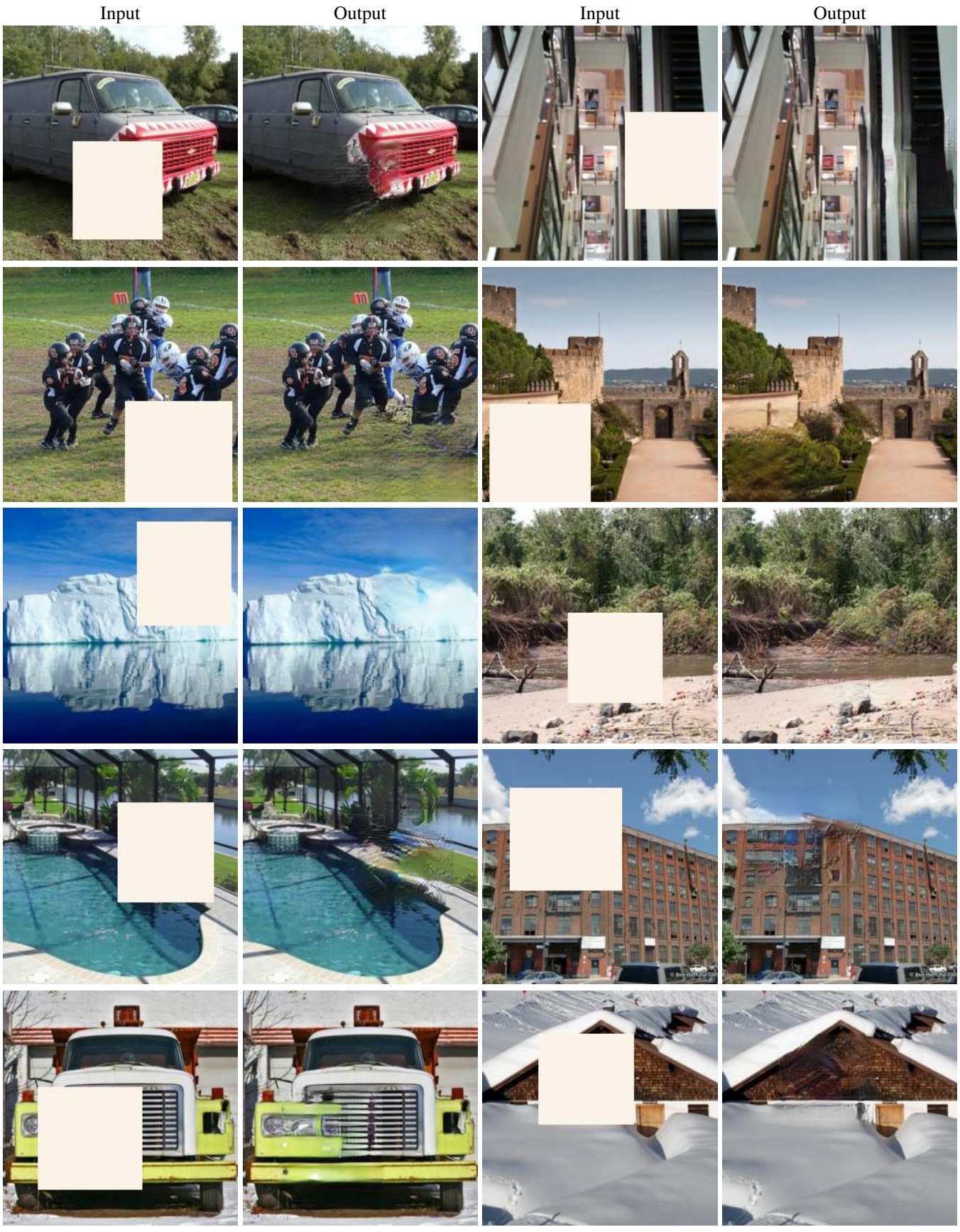
1.3. Places2

We train PGGAN with 8 millions images from Places2 dataset. During the training, inputs are scaled to size of 256x256 and random sized mask is applied to them. Results are presented below.




Input
Output
Input
Output


Input
Output
Input
Output


Input
Output
Input
Output