Free-Form Image Inpainting with Gated Convolution

Jiahui Yu[1]   Zhe Lin[2]   Jimei Yang[2]   Xiaohui Shen[3]   Xin Lu[2]   Thomas Huang[1]

[1]University of Illinois at Urbana-Champaign
[2]Adobe Research
[3]ByteDance AI Lab

Figure 1: Free-form image inpainting results by our system built on gated convolution. Each triad shows original image, free-form input and our result from left to right. The system supports free-form mask and guidance like user sketch. It helps user remove distracting objects, modify image layouts and edit faces in images.

Abstract

We present a generative image inpainting system to complete images with free-form mask and guidance. The system is based on gated convolutions learned from millions of images without additional labelling efforts. The proposed gated convolution solves the issue of vanilla convolution that treats all input pixels as valid ones, generalizes partial convolution by providing a learnable dynamic feature selection mechanism for each channel at each spatial location across all layers. Moreover, as free-form masks may appear anywhere in images with any shape, global and local GANs designed for a single rectangular mask are not applicable. Thus, we also present a patch-based GAN loss, named SN-PatchGAN, by applying spectral-normalized discriminator on dense image patches. SN-PatchGAN is simple in formulation, fast and stable in training. Results on automatic image inpainting and user-guided extension demonstrate that our system generates higher-quality and more flexible results than previous methods. Our system helps user quickly remove distracting objects, modify image layouts, clear watermarks and edit faces. Code, demo and models are available at: https://github.com/JiahuiYu/generative_inpainting.

1. Introduction

Image inpainting (a.k.a. image completion or image hole-filling) is a task of synthesizing alternative contents in missing regions such that the modification is visually realistic and semantically correct. It allows to remove distracting objects or retouch undesired regions in photos. It can also be extended to tasks including image/video un-cropping, rotation, stitching, re-targeting, re-composition, compression, super-resolution, harmonization and many others.

In computer vision, two broad approaches to image inpainting exist: patch matching using low-level image features and feed-forward generative models with deep convolutional networks. The former approach [3, 8, 9] can synthesize plausible stationary textures, but usually makes critical failures in non-stationary cases like complicated scenes, faces and objects. The latter approach [15, 49, 45, 46, 38, 37, 48, 26, 52, 33, 35, 19] can exploit semantics learned from large scale datasets to synthesize contents in non-stationary images in an end-to-end fashion.

However, deep generative models based on vanilla con-

volutions are naturally ill-fitted for image hole-filling because the spatially shared convolutional filters treat all input pixels or features as same valid ones. For hole-filling, the input to each layer are composed of valid pixels/features outside holes and invalid ones in masked regions. Vanilla convolutions apply same filters on all valid, invalid and mixed (for example, the ones on hole boundary) pixels/features, leading to visual artifacts such as color discrepancy, blurriness and obvious edge responses surrounding holes when tested on free-form masks [15, 49].

To address this limitation, recently partial convolution [23] is proposed where the convolution is masked and normalized to be conditioned only on valid pixels. It is then followed by a rule-based mask-update step to update valid locations for next layer. Partial convolution categorizes all input locations to be either invalid or valid, and multiplies a zero-or-one mask to inputs throughout all layers. The mask can also be viewed as a single un-learnable feature gating channel[1]. However this assumption has several limitations. First, considering the input spatial locations across different layers of a network, they may include (1) valid pixels in input image, (2) masked pixels in input image, (3) neurons with receptive field covering no valid pixel of input image, (4) neurons with receptive field covering different number of valid pixels of input image (these valid image pixels may also have different relative locations), and (5) synthesized pixels in deep layers. Heuristically categorizing all locations to be either invalid or valid ignores these important information. Second, if we extend to user-guided image inpainting where users provide sparse sketch inside the mask, should these pixel locations be considered as valid or invalid? How to properly update the mask for next layer? Third, for partial convolution the "invalid" pixels will progressively disappear layer by layer and the rule-based mask will be all ones in deep layers. However, to synthesize pixels in hole these deep layers may also need the information of whether current locations are inside or outside the hole. The partial convolution with all-ones mask cannot provide such information. We will show that if we allow the network to learn the mask automatically, the mask may have different values based on whether current locations are masked or not in input image, even in deep layers.

We propose gated convolution for free-form image inpainting. It learns a dynamic feature gating mechanism for each channel and each spatial location (for example, inside or outside masks, RGB channels or user-guidance channels). Specifically we consider the formulation where the input feature is firstly used to compute gating values $g = \sigma(w_g x)$ (σ is sigmoid function, w_g is learnable param-

---

[1]Applying mask before convolution or after is equivalent when convolutions are stacked layer-by-layer in neural networks. Because the output of current layer is the input to next layer and the masked region of input image is already filled with zeros.

eter). The final output is a multiplication of learned feature and gating values $y = \phi(wx) \odot g$ where ϕ can be any activation function. Gated convolution is easy to implement and performs significantly better when (1) the masks have arbitrary shapes and (2) the inputs are no longer simply RGB channels with a mask but also have conditional inputs like sparse sketch. For network architectures, we stack gated convolution to form an encoder-decoder network following [49]. Our inpainting network also integrates contextual attention module within same refinement network [49] to better capture long-range dependencies.

Without compromise of performance, we also significantly simplify training objectives as two terms: a pixel-wise reconstruction loss and an adversarial loss. The modification is mainly designed for free-form image inpainting. As the holes may appear anywhere in images with any shape, global and local GANs [15] designed for a single rectangular mask are not applicable. Instead, we propose a variant of generative adversarial networks, named SN-PatchGAN, motivated by global and local GANs [15], MarkovianGANs [21], perceptual loss [17] and recent work on spectral-normalized GANs [24]. The discriminator of SN-PatchGAN directly computes hinge loss on each point of the output map with format $\mathbb{R}^{h \times w \times c}$, formulating $h \times w \times c$ number of GANs focusing on different locations and different semantics (represented in different channels). SN-PatchGAN is simple in formulation, fast and stable in training and produces high-quality inpainting results.

Table 1: Comparison of different approaches including PatchMatch [3], Global&Local [15], ContextAttention [49], PartialConv [23] and our approach. The comparison of image inpainting is based on four dimensions: Semantic Understanding, Non-Local Algorithm, Free-Form Masks and User-Guided Option.

	PM [3]	GL [15]	CA [49]	PC [23]	Ours
Semantics		✓	✓	✓	✓
Non-Local	✓		✓		✓
Free-Form	✓			✓	✓
User-guided	✓				✓

For practical image inpainting tools, enabling user interactivity is crucial because there could exist many plausible solutions for filling a hole in an image. To this end, we present an extension to allow user sketch as guided input. Comparison to other methods is summarized in Table 1. Our main contributions are as follows: (1) We introduce gated convolution to learn a dynamic feature selection mechanism for each channel at each spatial location across all layers, significantly improving the color consistency and inpainting quality of free-form masks and inputs. (2) We present a more practical patch-based GAN discriminator,

SN-PatchGAN, for free-form image inpainting. It is simple, fast and produces high-quality inpainting results. (3) We extend our inpainting model to an interactive one, enabling user sketch as guidance to obtain more user-desired inpainting results. (4) Our proposed inpainting system achieves higher-quality free-form inpainting than previous state of the arts on benchmark datasets including Places2 natural scenes and CelebA-HQ faces. We show that the proposed system helps user quickly remove distracting objects, modify image layouts, clear watermarks and edit faces in images.

2. Related Work

2.1. Automatic Image Inpainting

A variety of approaches have been proposed for image inpainting. Traditionally, patch-based [8, 9] algorithms progressively extend pixels close to the hole boundaries based on low-level features (for example, features of mean square difference on RGB space), to search and paste the most similar image patch. These algorithms work well on stationary textural regions but often fail on non-stationary images. Further, Simakov et al. propose bidirectional similarity synthesis approach [36] to better capture and summarize non-stationary visual data. To reduce the high cost of memory and computation during search, tree-based acceleration structures of memory [25] and randomized algorithms [3] are proposed. Moreover, inpainting results are improved by matching local features like image gradients [2, 5] and offset statistics of similar patches [11]. Recently, image inpainting systems based on deep learning are proposed to directly predict pixel values inside masks. A significant advantage of these models is the ability to learn adaptive image features for different semantics. Thus they can synthesize more visually plausible contents especially for images like faces [22, 47], objects [29] and natural scenes [15, 49]. Among all these methods, Iizuka et al. [15] propose a fully convolutional image inpainting network with both global and local consistency to handle high-resolution images on a variety of datasets [18, 32, 53]. This approach, however, still heavily relies on Poisson image blending with traditional patch-based inpainting results [11]. Yu et al. [49] propose an end-to-end image inpainting model by adopting stacked generative networks to further ensure the color and texture consistence of generated regions with surroundings. Moreover, for capturing long-range spatial dependencies, contextual attention module [49] is proposed and integrated into networks to explicitly borrow information from distant spatial locations. However, this approach is mainly trained on large rectangular masks and does not generalize well on free-form masks. To better handle irregular masks, partial convolution [23] is proposed where the convolution is masked and re-normalized to utilize valid pixels only. It

is then followed by a rule-based mask-update step to re-compute new masks layer by layer.

2.2. Guided Image Inpainting and Synthesis

To improve image inpainting, user guidance is explored including dots or lines [1, 3, 7, 40], structures [13], transformation or distortion information [14, 30] and image exemplars [4, 10, 20, 43, 51]. Notably, Hays and Efros [10] first utilize millions of photographs as a database to search for an example image which is most similar to the input, and then complete the image by cutting and pasting the corresponding regions from the matched image.

Recent advances in conditional generative networks empower user-guided image processing, synthesis and manipulation learned from large-scale datasets. Here we selectively review several related work. Zhang et al. [50] propose colorization networks which can take user guidance as additional inputs. Wang et al. [42] propose to synthesize high-resolution photo-realistic images from semantic label maps using conditional generative adversarial networks. The Scribbler [34] explore a deep generative network conditioned on sketched boundaries and sparse color strokes to synthesize cars, bedrooms, or faces.

2.3. Feature-wise Gating

Feature-wise gating has been explored widely in vision [12, 28, 39, 41], language [6], speech [27] and many other tasks. For examples, Highway Networks [39] utilize feature gating to ease gradient-based training of very deep networks. Squeeze-and-Excitation Networks re-calibrate feature responses by explicitly multiplying each channel with learned sigmoidal gating values. WaveNets [27] achieve better results by employing a special feature gating $y = \tanh(w_1 x) \cdot \text{sigmoid}(w_2 x)$ for modeling audio signals.

3. Approach

In this section, we describe our approach from bottom to top. We first introduce the details of the Gated Convolution, SN-PatchGAN, and then present the overview of inpainting network in Figure 3 and our extension to allow optional user guidance.

3.1. Gated Convolution

We first explain why vanilla convolutions used in [15, 49] are ill-fitted for the task of free-form image inpainting. We consider a convolutional layer in which a bank of filters are applied to the input feature map as output. Assume input is $C-channel$, each pixel located at (y, x) in $C'-channel$ output map is computed as

$$O_{y,x} = \sum_{i=-k'_h}^{k'_h} \sum_{j=-k'_w}^{k'_w} W_{k'_h+i,k'_w+j} \cdot I_{y+i,x+j},$$

where x, y represents x-axis, y-axis of output map, k_h and k_w is the kernel size (*e.g.* 3×3), $k'_h = \frac{k_h-1}{2}$, $k'_w = \frac{k_w-1}{2}$, $W \in \mathbb{R}^{k_h \times k_w \times C' \times C}$ represents convolutional filters, $I_{y+i,x+j} \in \mathbb{R}^C$ and $O_{y,x} \in \mathbb{R}^{C'}$ are inputs and outputs. For simplicity, the bias in convolution is ignored.

The equation shows that for all spatial locations (y, x), the same filters are applied to produce the output in vanilla convolutional layers. This makes sense for tasks such as image classification and object detection, where all pixels of input image are valid, to extract local features in a sliding-window fashion. However, for image inpainting, the input are composed of both regions with valid pixels/features outside holes and invalid pixels/features (in shallow layers) or synthesized pixels/features (in deep layers) in masked regions. This causes ambiguity during training and leads to visual artifacts such as color discrepancy, blurriness and obvious edge responses during testing, as reported in [23].

Recently partial convolution is proposed [23] which adapts a masking and re-normalization step to make the convolution dependent only on valid pixels as

$$O_{y,x} = \begin{cases} \sum \sum W \cdot (I \odot \frac{M}{sum(M)}), & \text{if sum(M)} > 0 \\ 0, & \text{otherwise} \end{cases}$$

in which M is the corresponding binary mask, 1 represents pixel in the location (y, x) is valid, 0 represents the pixel is invalid, $\odot$ denotes element-wise multiplication. After each partial convolution operation, the mask-update step is required to propagate new M with the following rule: $m'_{y,x} = 1$, iff sum(M) > 0.

Partial convolution [23] improves the quality of inpainting on irregular mask, but it still has remaining issues: (1) It heuristically classifies all spatial locations to be either valid or invalid. The mask in next layer will be set to ones no matter how many pixels are covered by the filter range in previous layer (for example, 1 valid pixel and 9 valid pixels are treated as same to update current mask). (2) It is incompatible with additional user inputs. We aim at a user-guided image inpainting system where users can optionally provide sparse sketch inside the mask as conditional channels. In this situation, should these pixel locations be considered as valid or invalid? How to properly update the mask for next layer? (3) For partial convolution the invalid pixels will progressively disappear in deep layers, gradually converting all mask values to ones. However, our study shows that if we allow the network to learn optimal mask automatically, the network assigns soft mask values to every spatial locations even in deep layers. (4) All channels in each layer share the same mask, which limits the flexibility. Essentially, partial convolution can be viewed as un-learnable single-channel feature hard-gating.

We propose gated convolution for image inpainting network, as shown in Figure 2. Instead of hard-gating mask
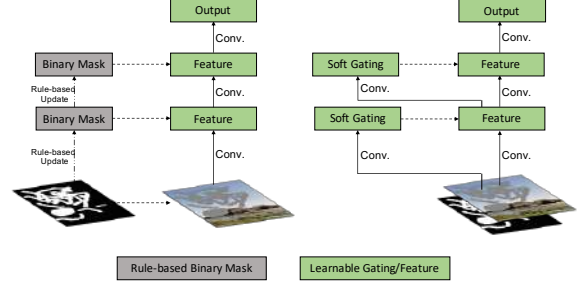

Figure 2: Illustration of partial convolution (left) and gated convolution (right).

updated with rules, gated convolutions learn soft mask automatically from data. It is formulated as:

$$Gating_{y,x} = \sum \sum W_g \cdot I$$
$$Feature_{y,x} = \sum \sum W_f \cdot I$$
$$O_{y,x} = \phi(Feature_{y,x}) \odot \sigma(Gating_{y,x})$$

where σ is sigmoid function thus the output gating values are between zeros and ones. ϕ can be any activation functions (for examples, ReLU, ELU and LeakyReLU). W_g and W_f are two different convolutional filters.

The proposed gated convolution learns a dynamic feature selection mechanism for each channel and each spatial location. Interestingly, visualization of intermediate gating values show that it learns to select the feature not only according to background, mask, sketch, but also considering semantic segmentation in some channels. Even in deep layers, gated convolution learns to highlight the masked regions and sketch information in separate channels to better generate inpainting results.

3.2. Spectral-Normalized Markovian Discriminator (SN-PatchGAN)

For previous inpainting networks which try to fill a single rectangular hole, an additional local GAN is used on the masked rectangular region to improve results [15, 49]. However, we consider the task of free-form image inpainting where there may be multiple holes with any shape at any location. Motivated by global and local GANs [15], MarkovianGANs [16, 21], perceptual loss [17] and recent work on spectral-normalized GANs [24], we present a simple and effective GAN loss, SN-PatchGAN, for training free-form image inpainting networks.

A convolutional network is used as the discriminator where the input consists of image, mask and guidance channels, and the output is a 3-D feature of shape $\mathbb{R}^{h \times w \times c}$ (h, w, c representing the height, width and number of channels respectively). As shown in Figure 3, six strided convolutions with kernel size 5 and stride 2 is stacked to captures
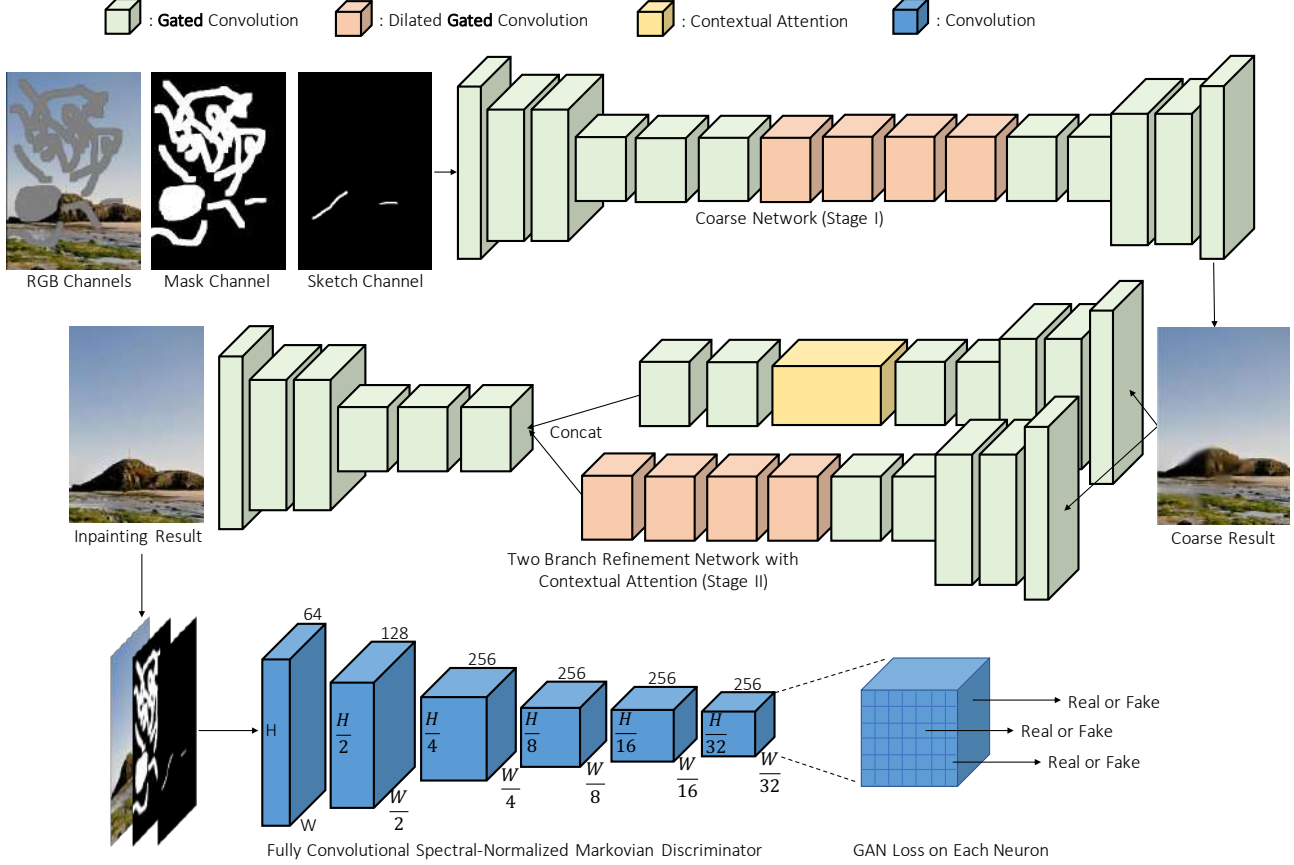
Figure 3: Overview of our framework with gated convolution and SN-PatchGAN for free-form image inpainting.

the feature statistics of Markovian patches [21]. We then directly apply GANs for each feature element in this feature map, formulating $h \times w \times c$ number of GANs focusing on different locations and different semantics (represented in different channels) of input image. It is noteworthy that the receptive field of each neuron in output map can cover entire input image in our training setting, thus a global discriminator is not necessary.

We also adapt the recently proposed spectral normalization [24] to further stabilize the training of GANs. We use the default fast approximation algorithm of spectral normalization described in SN-GANs [24]. To discriminate if the input is real or fake, we also use the hinge loss as objective function for generator $\mathcal{L}_G = -\mathbb{E}_{z \sim \mathbb{P}_z(z)}[D^{sn}(G(z))]$ and discriminator $\mathcal{L}_{D^{sn}} = \mathbb{E}_{x \sim \mathbb{P}_{data}(x)}[ReLU(\mathbb{1} - D^{sn}(x))] + \mathbb{E}_{z \sim \mathbb{P}_z(z)}[ReLU(\mathbb{1} + D^{sn}(G(z)))]$ where D^{sn} represents spectral-normalized discriminator, G is image inpainting network that takes incomplete image z.

With SN-PatchGAN, our inpainting network trains faster and more stable than baseline model [49]. Perceptual loss is not used since similar patch-level information is already encoded in SN-PatchGAN. Compared with Partial-Conv [23] in which 6 different loss terms and balancing

hyper-parameters are used, our final objective function for inpainting network is only composed of pixel-wise ℓ_1 reconstruction loss and SN-PatchGAN loss, with default loss balancing hyper-parameter as $1:1$.

3.3. Inpainting Network Architecture

We customize a generative inpainting network [49] with the proposed gated convolution and SN-PatchGAN loss. Specifically, we adapt the full model architecture in [49] with both coarse and refinement networks. The full framework is summarized in Figure 3.

For coarse and refinement networks, we use a simple encoder-decoder network [49] instead of U-Net used in PartialConv [23]. We found that skip connections in a U-Net [31] have no significant effect for non-narrow mask. This is mainly because for center of a masked region, the inputs of these skip connections are almost zeros thus cannot propagate detailed color or texture information to the decoder of that region. For hole boundaries, our encoder-decoder architecture equipped with gated convolution is sufficient to generate seamless results.

We replace all vanilla convolutions with gated convolutions [49]. One potential problem is that gated convolutions

introduce additional parameters. To maintain the same efficiency with our baseline model [49], we slim the model width by 25% and have not found obvious performance drop both quantitatively and qualitatively. The inpainting network is trained end-to-end and can be tested on free-form holes at arbitrary locations. Our network is fully convolutional and supports different input resolutions in inference.

3.4. Free-Form Mask Generation

The algorithm to automatically generate free-form masks is important and non-trivial. The sampled masks, in essence, should be (1) similar to masks drawn in real use-cases, (2) diverse to avoid over-fitting, (3) efficient in computation and storage, (4) controllable and flexible. Previous method [23] collects a fixed set of irregular masks from an occlusion estimation method between two consecutive frames of videos. Although random dilation, rotation and cropping are added to increase its diversity, the method does not meet other requirements listed above.

We introduce a simple algorithm to automatically generate random free-form masks on-the-fly during training. For the task of hole filling, users behave like using an eraser to brush back and forth to mask out undesired regions. This behavior can be simply simulated with a randomized algorithm by drawing lines and rotating angles repeatedly. To ensure smoothness of two lines, we also draw a circle in joints between the two lines. More details are included in the supplementary materials due to space limit.

3.5. Extension to User-Guided Image Inpainting

We use sketch as an example user guidance to extend our image inpainting network as a user-guided system. Sketch (or edge) is simple and intuitive for users to draw. We show both cases with faces and natural scenes. For faces, we extract landmarks and connect related landmarks. For natural scene images, we directly extract edge maps using the HED edge detector [44] and set all values above a certain threshold (*i.e.* 0.6) to ones. Sketch examples are shown in the supplementary materials due to space limit.

For training the user-guided image inpainting system, intuitively we will need additional constraint loss to enforce the network generating results conditioned on the user guidance. However with the same combination of pixel-wise reconstruction loss and GAN loss (with conditional channels as input to the discriminator), we are able to learn conditional generative network in which the generated results respect user guidance faithfully. We also tried to use additional pixel-wise loss on HED [44] output features with the raw image or the generated result as input to enforce constraints, but the inpainting quality is similar. The user-guided inpainting model is separately trained with a 5-channel input (R,G,B color channels, mask channel and sketch channel).

4. Results

We evaluate the proposed free-form image inpainting system on Places2 [53] and CelebA-HQ faces [18]. Our model has totally 4.1M parameters, and is trained with TensorFlow v1.8, CUDNN v7.0, CUDA v9.0. For testing, it runs at 0.21 seconds per image on single NVIDIA(R) Tesla(R) V100 GPU and 1.9 seconds on Intel(R) Xeon(R) CPU @ 2.00GHz for images of resolution 512×512 on average, regardless of hole size.

4.1. Quantitative Results

As mentioned in [49], image inpainting lacks good quantitative evaluation metrics. Nevertheless, we report in Table 2 our evaluation results in terms of mean ℓ_1 error and mean ℓ_2 error on validation images of Places2, with both center rectangle mask and free-form mask. As shown in the table, learning-based methods perform better than PatchMatch [3] in terms of mean ℓ_1 and ℓ_2 errors. Moreover, partial convolution implemented within the same framework obtains worse performance, which may due to un-learnable rule-based gating.

Table 2: Results of mean ℓ_1 error and mean ℓ_2 error on validation images of Places2 with both rectangle masks and free-form masks. Both PartialConv* and ours are trained on same random combination of rectangle and free-form masks. No edge guidance is utilized in training/inference to ensure fair comparison. * denotes our implementation within the same framework due to unavailability of official implementation and models.

Method	rectangular mask		free-form mask	
	ℓ_1 err.	ℓ_2 err.	ℓ_1 err.	ℓ_2 err.
PatchMatch [3]	16.1%	3.9%	11.3%	2.4%
Global&Local [15]	9.3%	2.2%	21.6%	7.1%
ContextAttention [49]	**8.6%**	2.1%	17.2%	4.7%
PartialConv* [23]	9.8%	2.3%	10.4%	1.9%
Ours	**8.6%**	**2.0%**	**9.1%**	**1.6%**

4.2. Qualitative Comparisons

Next, we compare our model with previous state-of-the-art methods [15, 23, 49]. Figure 4 and Figure 5 shows automatic and user-guided inpainting results with several representative images. For automatic image inpainting, the result of PartialConv is obtained from its online demo[2]. For user-guided image inpainting, we train PartialConv* with the exact same setting of GatedConv, expect the convolution types (sketch regions are treated as valid pixels for rule-based mask updating). For all learning-based methods, no post-processing step is performed to ensure fairness.

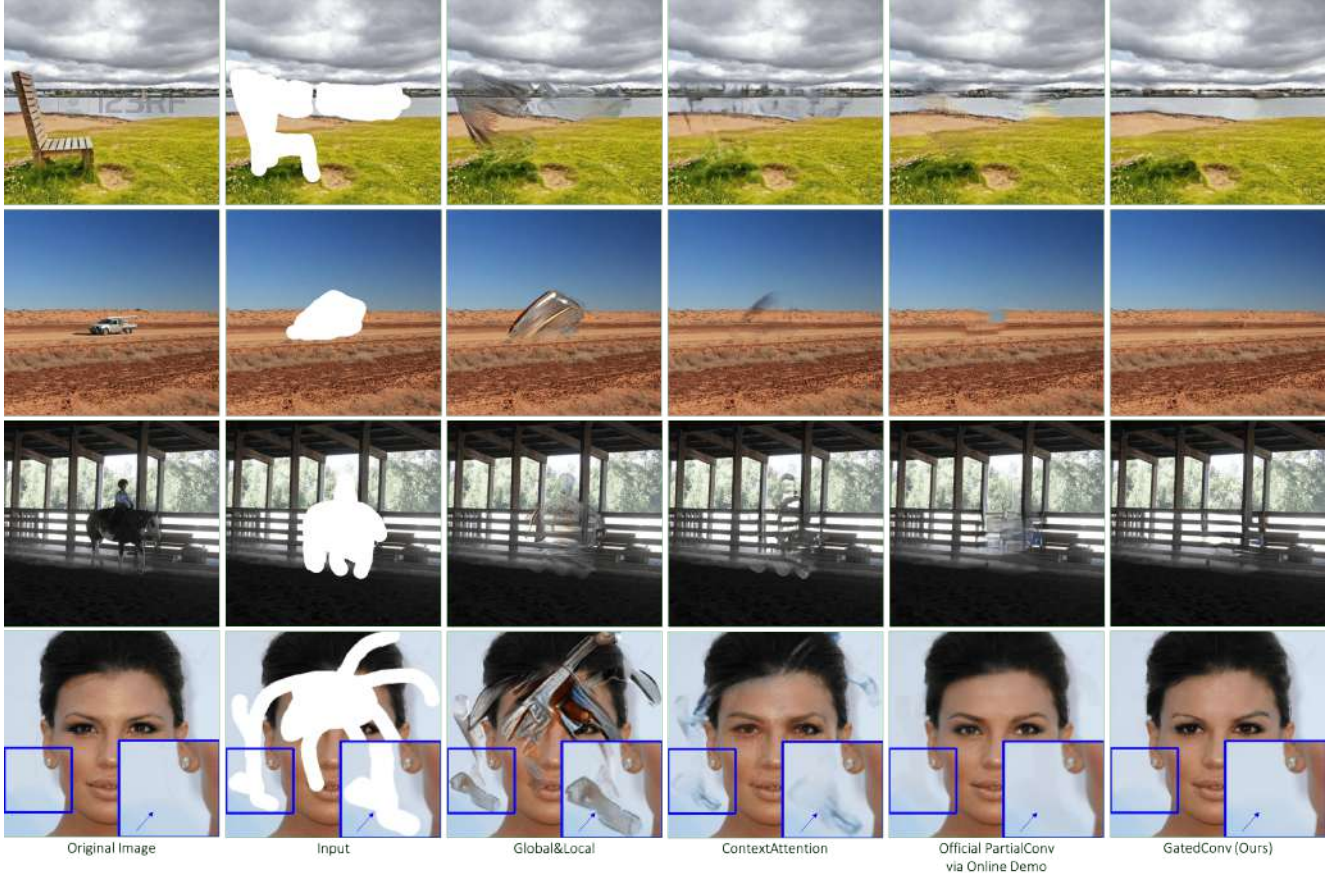
[2]https://www.nvidia.com/research/inpainting

Figure 4: Example cases of qualitative comparison on the Places2 and CelebA-HQ validation sets. More comparisons are included in supplementary materials due to space limit. Best viewed (*e.g.*, shadows in uniform region) with zoom-in.
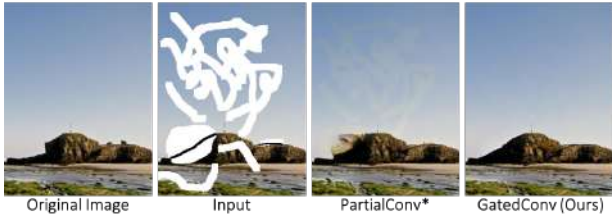

Figure 5: Comparison of user-guided image inpainting.

As reported in [15], simple uniform region (last row of Figure 4 and Figure 5) are hard cases for learning-based image inpainting networks. Previous methods with vanilla convolution have obvious visual artifacts and edge responses in/surrounding holes. PartialConv produces better results but still exhibits observable color discrepancy. Our method based on gated convolution obtains more visually pleasing results without noticeable color inconsistency. In Figure 5, given sparse sketch, our method produces realistic results with seamless boundary transitions.

4.3. Object Removal and Creative Editing

Moreover, we study two important real use cases of image inpainting: object removal and creative editing.

Object Removal. In the first example, we try to remove the distracting person in Figure 6. We compare our method with commercial product Photoshop (based on PatchMatch [3]) and the previous state-of-the-art generative inpainting network (official released model trained on Places2) [49]. The results show that *Content-Aware Fill* function from Photoshop incorrectly copies half of face from left. This example reflects the fact that traditional methods without learning from large-scale data ignore the semantics of an image, which leads to critical failures in non-stationary/complicated scenes. For learning-based methods with vanilla convolution [49], artifacts exist near hole boundaries.

Creative Editing. Next we study the case where user interacts with the inpainting system to produce more desired results. The examples on both faces and natural scenes are shown in Figure 7. Our inpainting results nicely follow the user sketch, which is useful for creatively editing image lay-

Figure 6: Object removal case study with comparison.

Figure 7: Examples of user-guided inpainting/editing of faces and natural scenes.

outs, faces and many others.

4.4. User Study

We performed a user study by first collecting 30 test images (with holes but no sketches) from Places2 validation dataset without knowing their inpainting results on each model. We then computed results of the following four methods for comparison: (1) ground truth, (2) our model, (3) re-implemented PartialConv [23] within same framework, and (4) official PartialConv [23]. We did two types of user study. (A) We evaluate each method individually to rate the naturalness/inpainting quality of results (from 1 to 10, the higher the better), and (B) we compare our model and the official PartialConv model to evaluate which method produces better results. 104 users finished the user study with the results shown as follows.

(A) Naturalness: (1) 9.89, (2) 7.72, (3) 7.07, (4) 6.54

(B) Pairwise comparison of (2) our model vs. (4) official PartialConv model: 79.4% vs. 20.6% (the higher the better).

4.5. Ablation Study of SN-PatchGAN

Figure 8: Ablation study of SN-PatchGAN. From left to right, we show original image, masked input, results with one global GAN and our results with SN-PatchGAN.

SN-PatchGAN is proposed for the reason that free-form masks may appear anywhere in images with any shape. Previously introduced global and local GANs [15] designed for a single rectangular mask are not applicable. We provide ablation experiments of SN-PatchGAN in the context of image inpainting in Figure 8. SN-PatchGAN leads to significantly better results, which verifies that (1) one vanilla global discriminator has worse performance [15], and (2) GAN with spectral normalization has better stability and performance [24]. Although introducing more loss functions may help in training free-form image inpainting networks [23], we demonstrate that a simple combination of SN-PatchGAN loss and pixel-wise ℓ_1 loss, with default loss balancing hyper-parameter as 1:1, produces photo-realistic inpainting results. More comparison examples are shown in the supplementary materials.

5. Conclusions

We presented a novel free-form image inpainting system based on an end-to-end generative network with gated convolution, trained with pixel-wise ℓ_1 loss and SN-PatchGAN. We demonstrated that gated convolutions significantly improve inpainting results with free-form masks and user guidance input. We showed user sketch as an exemplar guidance to help users quickly remove distracting objects, modify image layouts, clear watermarks, edit faces and interactively create novel objects in photos. Quantitative results, qualitative comparisons and user studies demonstrated the superiority of our proposed free-form image inpainting system.

References

[1] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226. ACM, 2001. 3

[2] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001. 3

[3] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2009)*, 2009. 1, 2, 3, 6, 7

[4] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 3

[5] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2012)*, 2012. 3

[6] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org, 2017. 3

[7] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *ACM Transactions on graphics (TOG)*. ACM, 2003. 3

[8] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001. 1, 3

[9] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999. 1, 3

[10] James Hays and Alexei A Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*. ACM, 2007. 3

[11] Kaiming He and Jian Sun. Image completion approaches using the statistics of similar patches. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2423–2435, 2014. 3

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[13] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image completion using planar structure guidance. *ACM Transactions on Graphics (TOG)*, 33(4):129, 2014. 3

[14] Jia-Bin Huang, Johannes Kopf, Narendra Ahuja, and Sing Bing Kang. Transformation guided image completion. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, pages 1–9. IEEE, 2013. 3

[15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017. 1, 2, 3, 4, 6, 7, 8, 11, 12, 13

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 4

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 2, 4

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 3, 6

[19] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019. 1

[20] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics (ToG)*, 24(3):795–802, 2005. 3

[21] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016. 2, 4, 5

[22] Yijun Li, Sifei Liu, Jimei Yang, and Ming-Hsuan Yang. Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919, 2017. 3

[23] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 2, 3, 4, 5, 6, 8, 11, 12

[24] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 2, 4, 5, 8

[25] David M Mount and Sunil Arya. Ann: library for approximate nearest neighbour searching, 1998. 3

[26] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019. 1

[27] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 3

[28] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4797–4805. Curran Associates Inc., 2016. 3

[29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 3

[30] Darko Pavić, Volker Schönefeld, and Leif Kobbelt. Interactive image completion with perspective correction. *The Visual Computer*, 22(9):671–681, 2006. 3

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3

[33] Min-cheol Sagong, Yong-goo Shin, Seung-wook Kim, Seung Park, and Sung-jea Ko. Pepsi: Fast image inpainting with parallel decoding network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11360–11368, 2019. 1

[34] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2017. 3

[35] Yong-Goo Shin, Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Wook Kim, and Sung-Jea Ko. Pepsi++: Fast and lightweight network for image inpainting. *arXiv preprint arXiv:1905.09010*, 2019. 1

[36] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 3

[37] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1

[38] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018. 1

[39] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015. 3

[40] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *ACM Transactions on Graphics (ToG)*, 24(3):861–868, 2005. 3

[41] Hongzhen Wang, Ying Wang, Qian Zhang, Shiming Xiang, and Chunhong Pan. Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sensing*, 9(5):446, 2017. 3

[42] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 3

[43] Oliver Whyte, Josef Sivic, and Andrew Zisserman. Get out of my picture! internet-based inpainting. In *Proceedings of the 20th British Machine Vision Conference, London*, 2009. 3

[44] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 6, 11, 12

[45] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5840–5848, 2019. 1

[46] Chao Yang, Yuhang Song, Xiaofeng Liu, Qingming Tang, and C-C Jay Kuo. Image inpainting using block-wise procedural training with annealed adversarial counterpart. *arXiv preprint arXiv:1803.08943*, 2018. 1

[47] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017. 3

[48] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018. 1

[49] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 1, 2, 3, 4, 5, 6, 7, 11, 12, 13

[50] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 36(4):119, 2017. 3

[51] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Guided image inpainting: Replacing an image region by pulling content from another image. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1514–1523. IEEE, 2019. 3

[52] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019. 1

[53] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3, 6

In this supplementary material, we first provide details of our free-form mask generation algorithm in Section A and sketch generation algorithm in Section B. We then study the effects of sketch input in Section C with an example where the input image uses the same mask but different sketches. Next we provide visualization and interpretation of learned gating values in Section D. We show additional ablation study of our proposed SN-PatchGAN in Section E. We show more comparison results of Global&Local [15], ContextAttention [49], PartialConv [23] (both our implementation within same framework and official model via online demo[3]) and our GatedConv in Section F. We finally show more inpainting results of our system with support of free-form masks and user guidance on both natural scenes and faces in Section G. Moreover, a recorded *real-time* video demo is available at: https://youtu.be/uZkEi9Y2dj4.
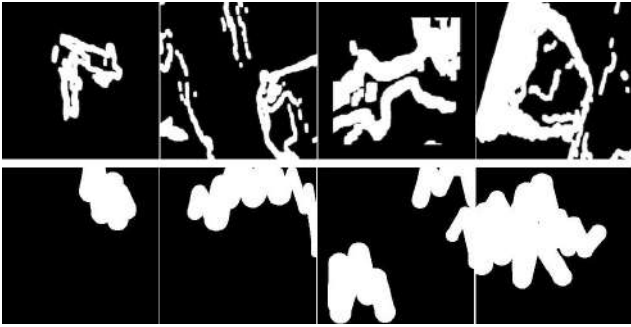
A. Free-Form Mask Generation

Figure 9: Sampled free-form masks with previous work [23] (1st row) and our automatic algorithm (2nd row).

The algorithm to automatically generate free-form masks is important and non-trivial. The sampled masks, in essence, should be (1) similar in shape to holes drawn in real use-cases, (2) diverse to avoid over-fitting, (3) efficient in computation and storage, (4) controllable and flexible. Previous method [23] collects a fixed set of irregular masks from an occlusion estimation method between two consecutive frames of videos. Although random dilation, rotation and cropping are added to increase its diversity, the method does not meet other requirements listed above.

We introduce a simple algorithm to automatically generate random free-form masks on-the-fly during training. For the task of hole filling, users behave like using an eraser to brush back and forth to mask out undesired regions. This behavior can be simply simulated with a randomized algorithm by drawing lines and rotating angles repeatedly. To ensure smoothness of two lines, we also draw a circle in joints between the two lines.

---

[3]https://www.nvidia.com/research/inpainting/

Algorithm 1 Algorithm for sampling free-form training masks. *maxVertex*, *maxLength*, *maxBrushWidth*, *maxAngle* are four hyper-parameters to control the mask generation.

mask = zeros(imageHeight, imageWidth)
numVertex = random.uniform(maxVertex)
startX = random.uniform(imageWidth)
startY = random.uniform(imageHeight)
brushWidth = random.uniform(maxBrushWidth)
for $i = 0$ to numVertex **do**
    angle = random.uniform(maxAngle)
    if (i % 2 == 0) **then**
        angle = 2 * pi - angle // comment: reverse mode
    end if
    length = random.uniform(maxLength)
    Draw line from point (startX, startY) with angle, length and brushWidth as line width.
    startX = startX + length * sin(angle)
    startY = startY + length * cos(angle)
    Draw a circle at point (startX, startY) with radius as half of brushWidth. // comment: ensure smoothness of strokes.
end for
mask = random.flipLeftRight(mask)
mask = random.flipTopBottom(mask)

We use *maxVertex*, *maxLength*, *maxWidth* and *maxAngle* as four hyper-parameters to provide large varieties of sampled masks. Moreover, our algorithm generates masks on-the-fly with little computational overhead and no storage is required. In practice, the computation of free-form masks on CPU can be easily hid behind training networks on GPU in modern deep learning frameworks. The overall mask generation algorithm is illustrated in Algorithm 1. Additionally we can sample multiple strokes in single image to mask multiple regions, and add regular masks (*e.g.* rectangular) on top of sampled free-form masks. Example masks compared with previous method [23] is shown in Figure 9.

B. Sketch Generation

Figure 10: For face dataset (on the left), we directly detect landmarks of faces and connect related nearby landmarks as training sketch, which is extremely robust and useful for editing faces. We use HED [44] model with threshold 0.6 to extract binary sketch for natural scenes (on the right).
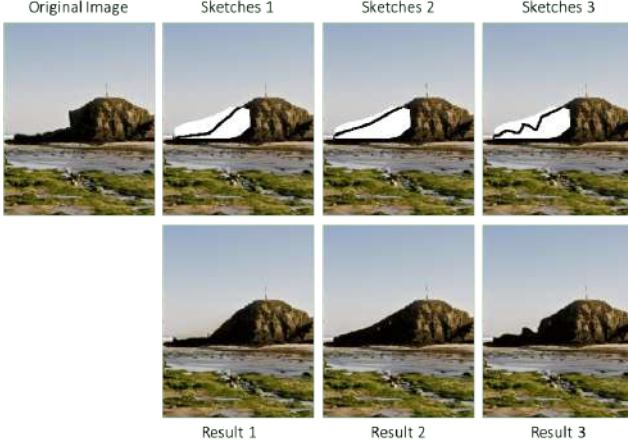
Figure 11: Image inpainting examples where the input image uses same mask but different sketches.

We use sketch as an example user guidance to extend our image inpainting network as a user guided system. We show both cases on faces and natural scenes. For faces, we extract landmarks and connect related landmarks. For natural scene images, we directly extract edge maps using the HED [44] edge detector and set all values above a certain threshold (*i.e.* 0.6) to ones. Sketch examples are shown in Figure 10. Alternative methods to generative better sketch or other user guidance should also work well with our user-guided image inpainting system.

C. The Effects of Sketch Input

As shown in Section 4.3, our inpainting network can nicely follow the user sketch, which is useful for creative editing of images. We show in Figure 11 an additional comparison case where the input image uses the same mask but different sketches.

D. Visualization and Interpretation

In Figure 12, we provide the visualization and interpretation of learned gating values in our inpainting network, and compare them with that of PartialConv [23].

E. Ablation Study of SN-PatchGAN

In this section, we present ablation study to demonstrate the effectiveness of SN-PatchGAN. It is noteworthy that SN-PatchGAN is proposed because free-form masks may appear anywhere in images with any shape. Global and local GANs [15] designed for a single rectangular mask are not applicable. Previous work have already shown that (1) one vanilla global discriminator has much worse performance than two local and global discriminators [15], and (2) GAN with spectral normalization has better stability and performance. We also provide experiments of

SN-PatchGAN in the context of image inpainting in Figure 13. Our image inpainting network trained on a global GAN without spectral normalization has significantly worse performance on all examples.

F. More Comparison Results

In this section, we show more comparison results of learning-based image inpainting systems including Global&Local [15], ContextAttention [49], PartialConv [23] (both our implementation within same framework and official model via online demo) and our proposed method based on gated convolution. Note that the models of scenes and faces are trained in separate following all other methods [15, 23, 49]. All testing images are not in the training set. Results are shown in Figure 14 and Figure 15. Compared with our baseline PartialConv, our inpainting system generates higher-quality inpainting results. Although PartialConv significantly improves over previous baselines like Global&Local [15] and ContextAttention [49], it still produces observable color inconsistency or shadows in both official online demo and our reproduced version (best-viewed with zoom-in on PDF to see color shadows and artifacts). Moreover, PartialConv fails especially on cases (1) when holes are large and involving transitions of two segments (*e.g.*, a mask covering both sky and ground), and (2) when the image has strong structure/contour/edge prior. The reasons are discussed in the introduction of main paper that unlearnable rule-based hard-gating heuristically categorizes all input locations to be either invalid or valid, ignoring many other important information. Gated convolution is able to leverage these information by learning a soft-gating end-to-end.

G. More Inpainting Results of Our System

In this section, we present more examples towards real use cases based on our proposed image inpainting system. We show inpainting results on both natural scenes and faces in Figure 16, Figure 17 and Figure 18. We show our inpainting system helps user quickly remove distracting objects, modify image layouts, edit faces and interactively create novel objects in images.
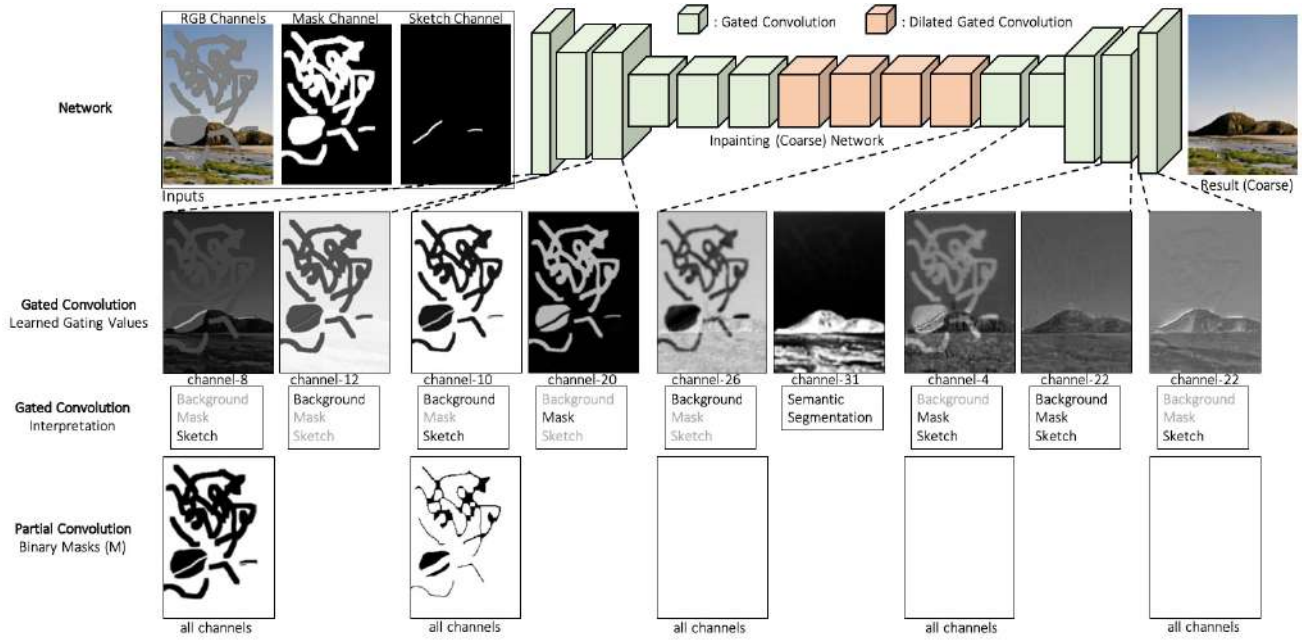
Figure 12: Comparisons of gated convolution and partial convolution with visualization and interpretation of learned gating values. We first show our inpainting network architecture based on [49] by replacing all convolutions with gated convolutions in the 1st row. Note that for simplicity, the following refinement network in [49] is ignored in the figure. With same settings, we train two models based on gated convolution and partial convolution separately. We then directly visualize intermediate un-normalized gating values in the 2nd row. The values differ mainly based on three parts: **background**, **mask** and **sketch**. In the 3rd row, we provide an interpretation based on which part(s) have higher gating values. Interestingly we also find that for some channels (*e.g.* channel-31 of the layer after dilated convolution), the learned gating values are based on foreground/background semantic segmentation. For comparison, we also visualize the un-learnable fixed binary mask M of partial convolution in the 4th row.

Figure 13: Ablation Study of SN-PatchGAN. From left to right, we show original image, masked input, results with one global GAN and our results with SN-PatchGAN. SN-PatchGAN is proposed because free-form masks may appear anywhere in images with any shape. Global and local GANs [15] designed for a single rectangular mask are not applicable.
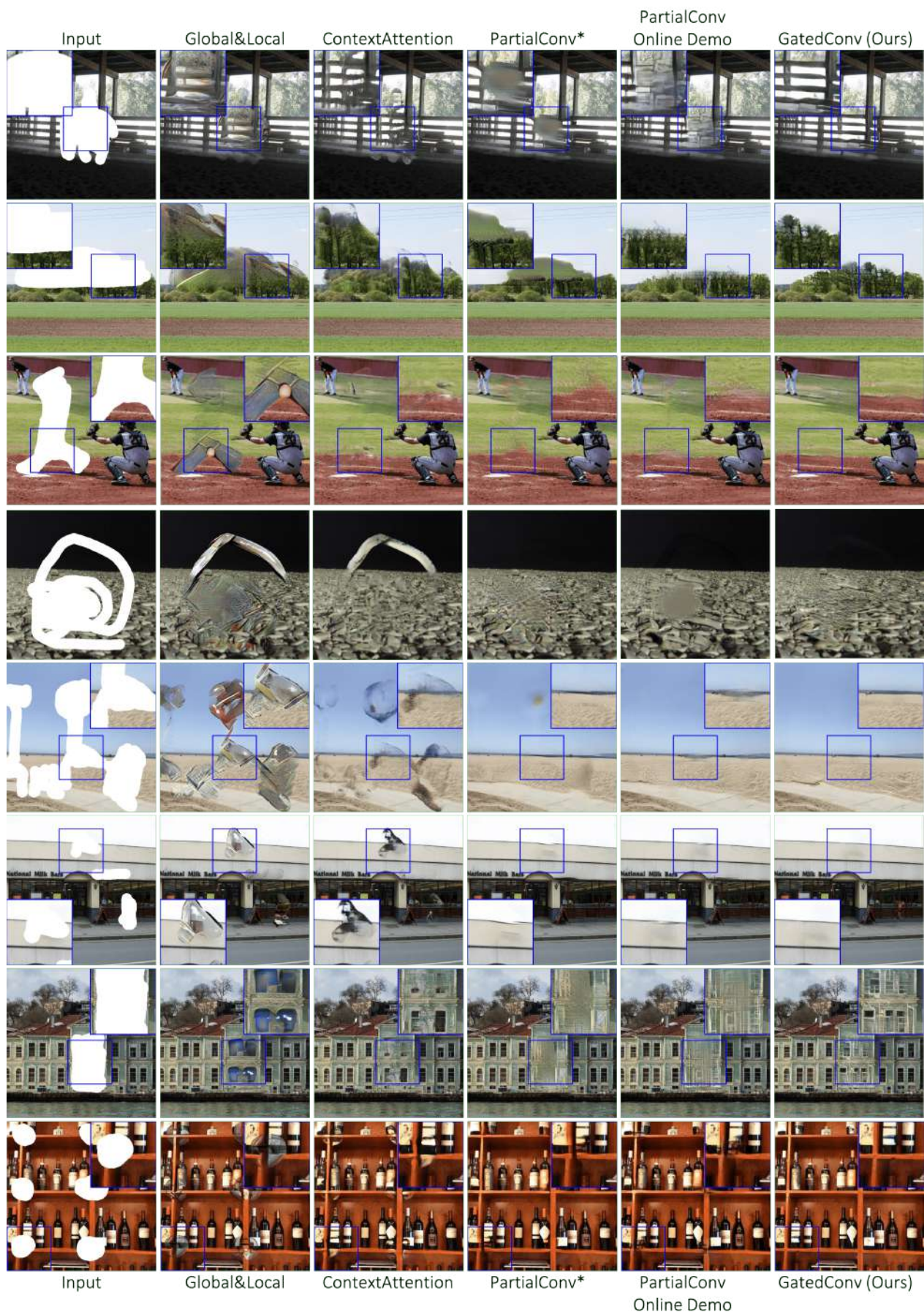
Figure 14: More comparison results on natural scenes. Best-viewed with zoom-in on PDF to see color shadows and artifacts.
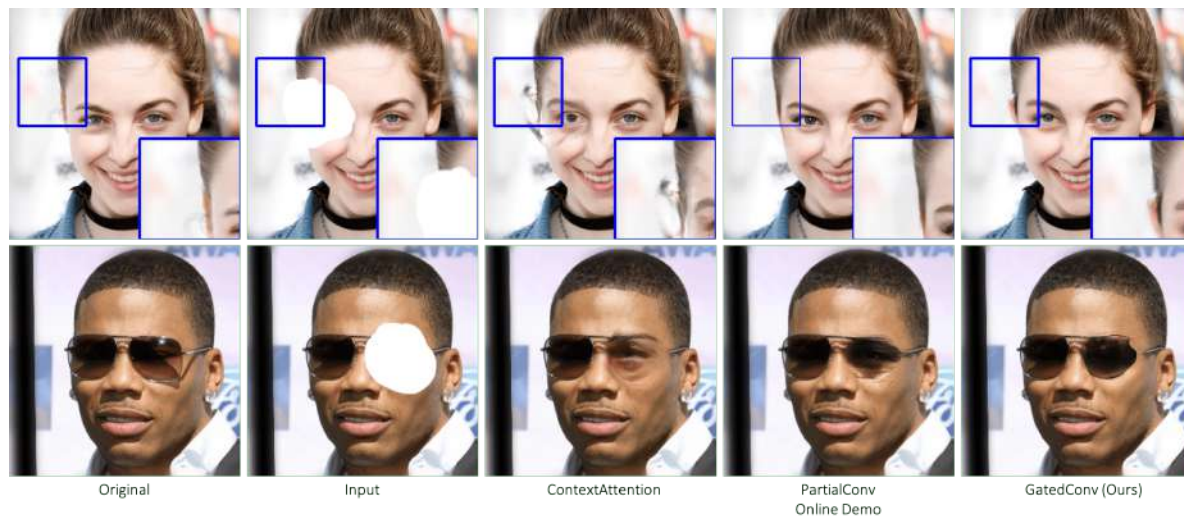
Figure 15: More comparison results on faces. Best-viewed with zoom-in on PDF to see color shadows and artifacts.
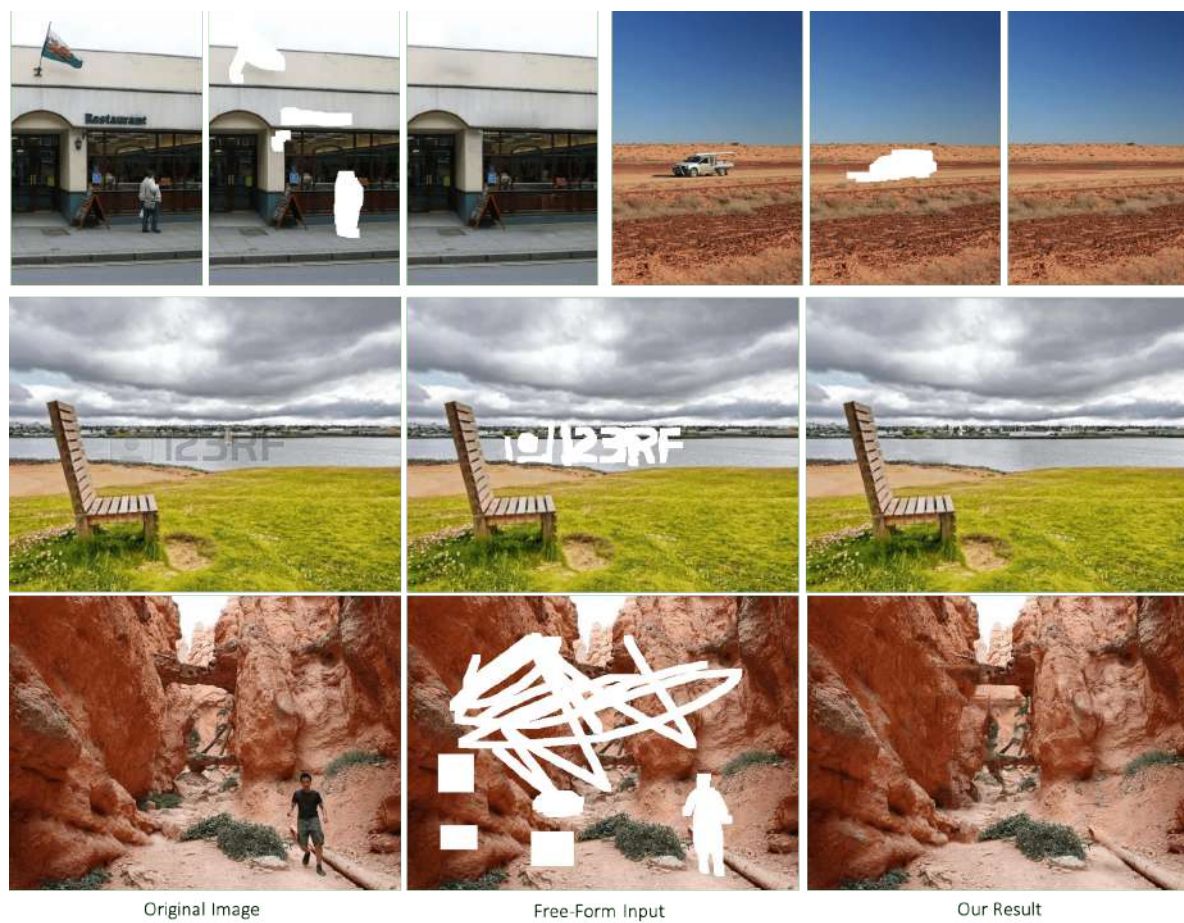

Figure 16: More results from our free-form inpainting system on natural images (1).
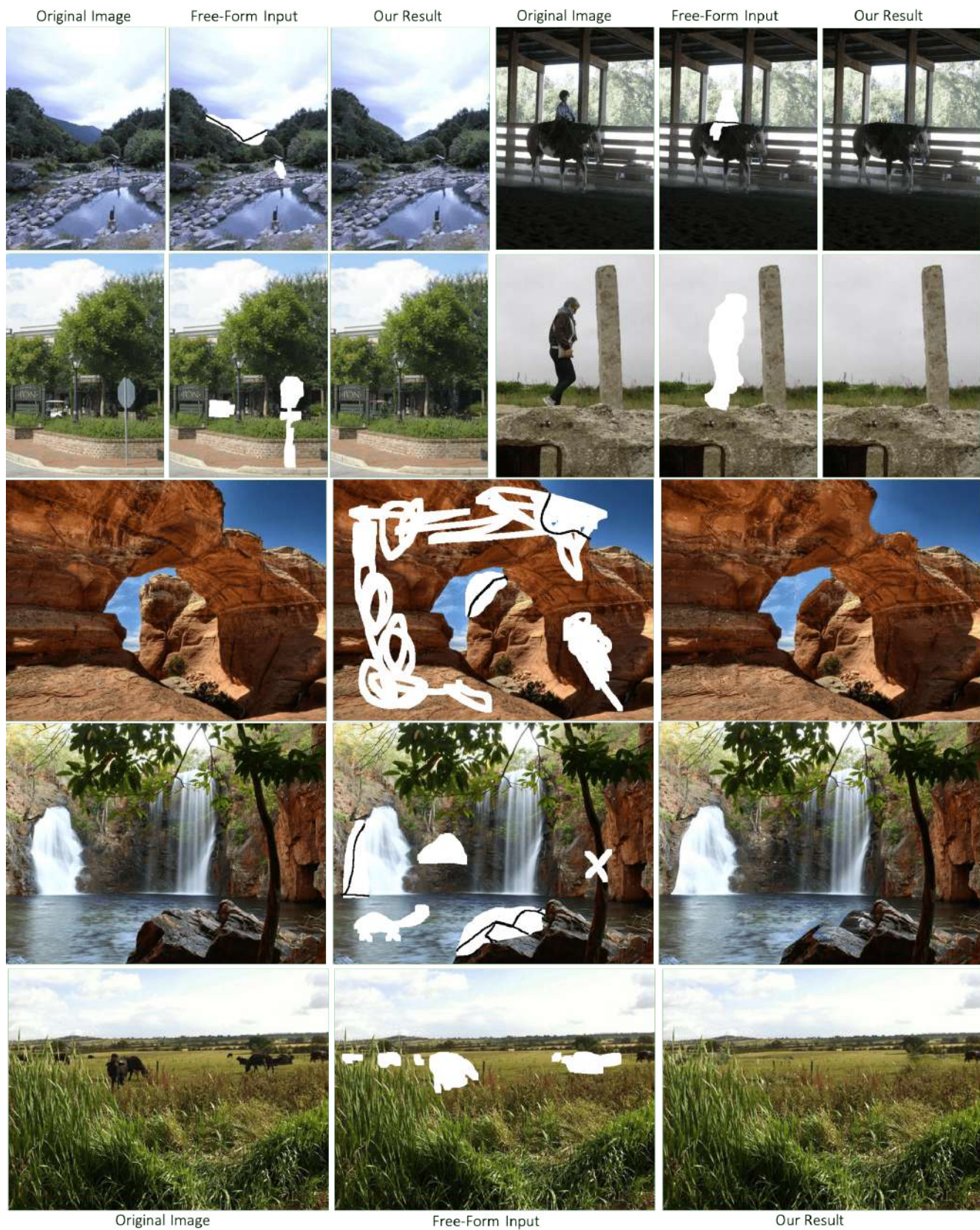
Figure 17: More results from our free-form inpainting system on natural images (2).

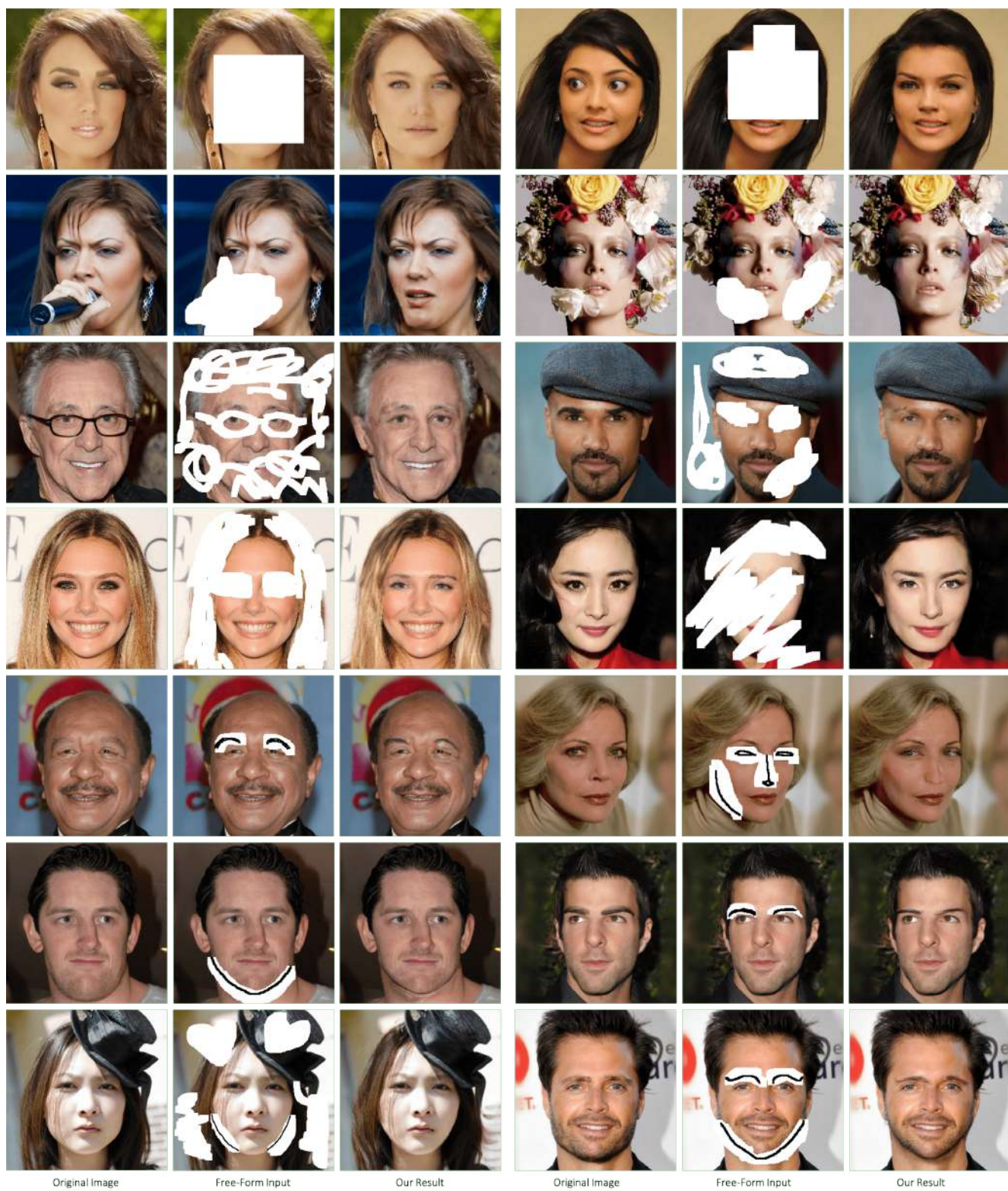| Original Image | Free-Form Input | Our Result | Original Image | Free-Form Input | Our Result |

Figure 18: More results from our free-form inpainting system on faces.