Free-Form Image Inpainting with Gated Convolution (Supplementary)

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

In this supplementary material, we first provide details of our free-form mask generation algorithm in Section 1 and sketch generation algorithm in Section 2. We then study the effects of sketch input in Section 3 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 4. We show additional ablation study of our proposed SN-PatchGAN in Section 5. We show more comparison results of Global&Local [1], ContextAttention [4], PartialConv [2] (both our implementation within same framework and official model via online demo[1]) and our GatedConv in Section 6. 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 7. Moreover, a recorded *real-time* video demo is available at: https://youtu.be/uZkEi9Y2dj4.

1. Free-Form Mask Generation

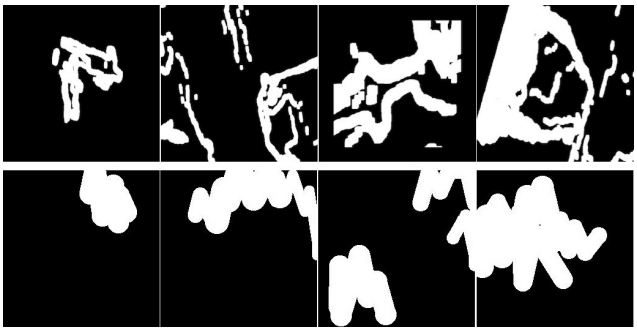

Figure 1: Sampled free-form masks with previous work [2] (1st row) and our automatic algorithm (2nd row).

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 [2] 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.

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

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

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 [2] is shown in Figure 1.

2. Sketch Generation

Figure 2: 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 [3] model with threshold 0.6 to extract binary sketch for natural scenes (on the right).

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 [3] edge detector and set all values above a certain threshold (*i.e.* 0.6) to ones. Sketch examples are shown in Figure 2. Alternative methods to generative better sketch or other user guidance should also work well with our user-guided image inpainting system.

3. 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 3 an additional comparison case where the input image uses the same mask but different sketches.

4. Visualization and Interpretation

In Figure 4, we provide the visualization and interpretation of learned gating values in our inpainting network, and compare them with that of PartialConv [2].

5. 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 [1] designed for a single rectangular mask are not applicable. Previous work have already shown that (1) one vanilla global discriminator has much worse performance

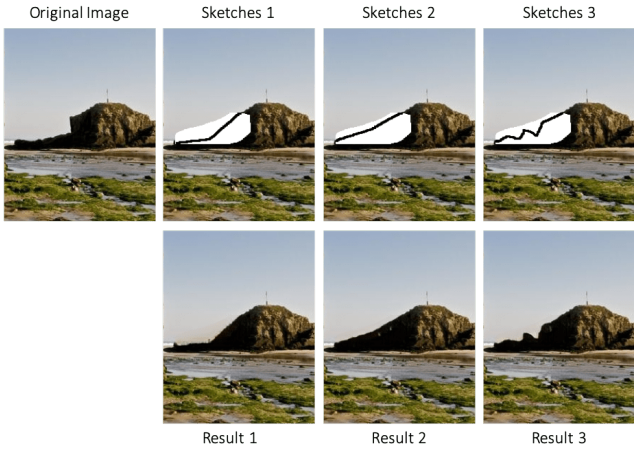

Figure 3: Image inpainting examples where the input image uses same mask but different sketches.

than two local and global discriminators [1], 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 5. Our image inpainting network trained on a global GAN without spectral normalization has significantly worse performance on all examples.

6. More Comparison Results

In this section, we show more comparison results of learning-based image inpainting systems including Global&Local [1], ContextAttention [4], PartialConv [2] (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 [1, 2, 4]. All testing images are not in the training set. Results are shown in Figure 6 and Figure 7. Compared with our baseline PartialConv, our inpainting system generates higher-quality inpainting results. Although PartialConv significantly improves over previous baselines like Global&Local [1] and ContextAttention [4], 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.

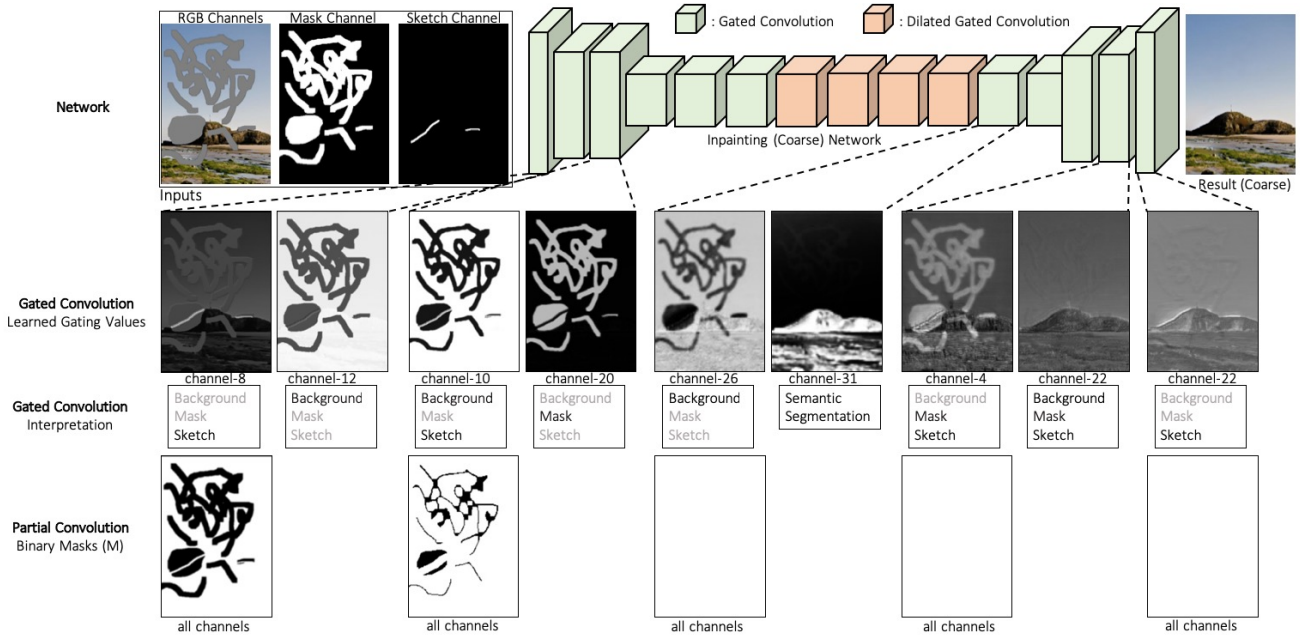

Figure 4: Comparisons of gated convolution and partial convolution with visualization and interpretation of learned gating values. We first show our inpainting network architecture based on [4] by replacing all convolutions with gated convolutions in the 1st row. Note that for simplicity, the following refinement network in [4] 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 5: 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 [1] designed for a single rectangular mask are not applicable.

7. 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 8, Figure 9 and Figure 10. We show our inpainting system helps user quickly remove distracting objects, modify image layouts, edit faces and interactively create novel objects in images.

References

[1] 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

[2] 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. 1, 2

[3] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 2

[4] 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

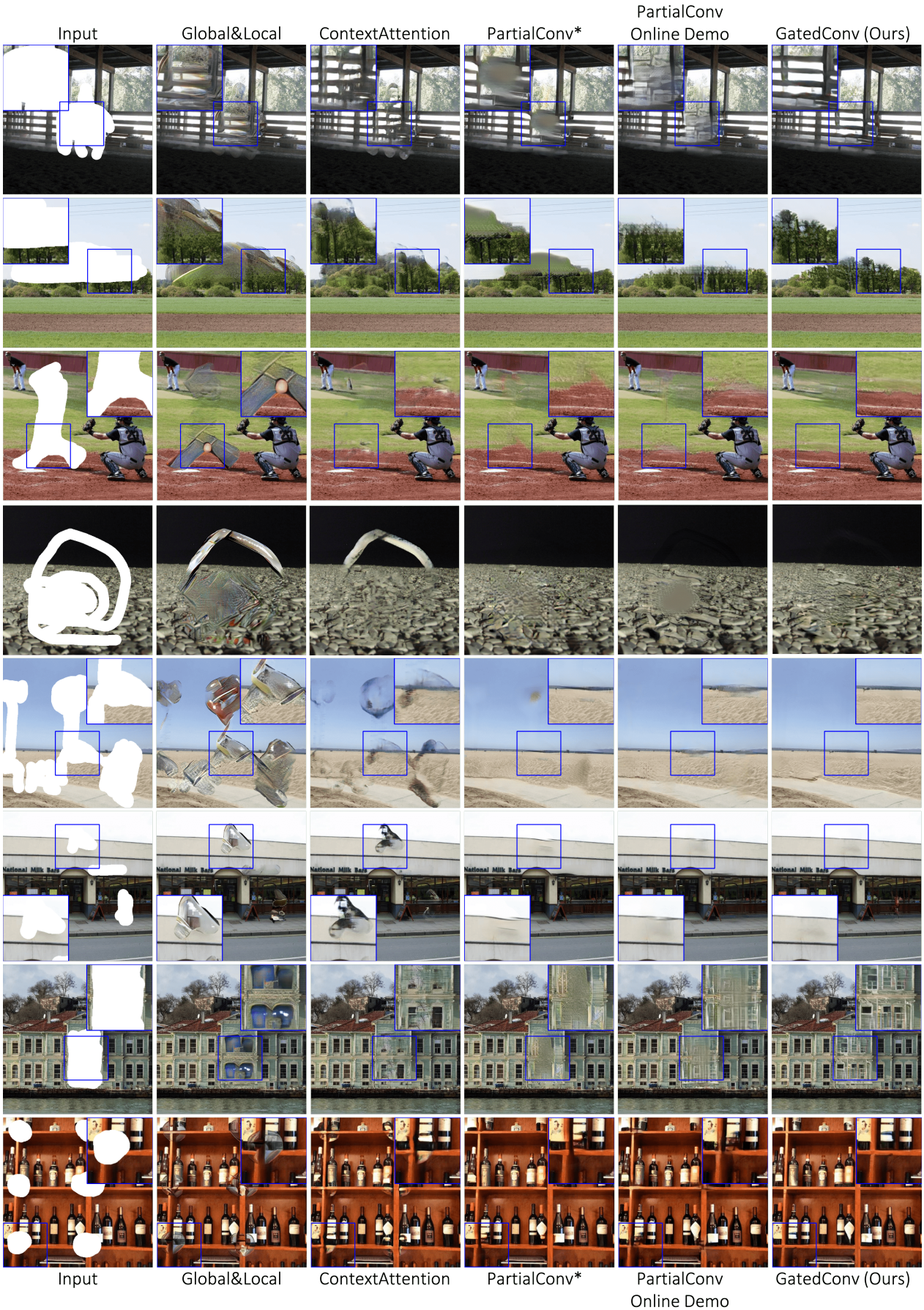

Figure 6: More comparison results on natural scenes. Best-viewed with zoom-in on PDF to see color shadows and artifacts.

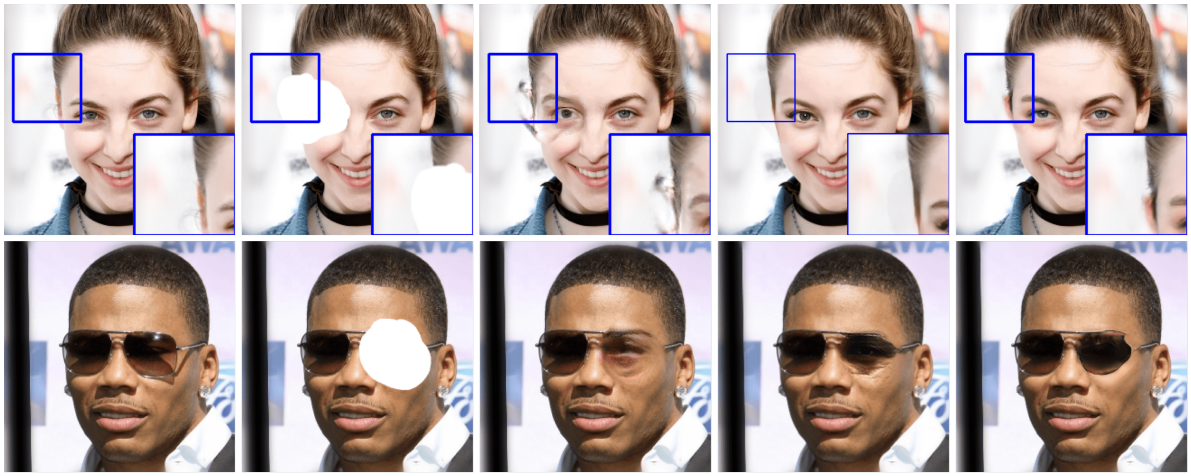

Figure 7: More comparison results on faces. Best-viewed with zoom-in on PDF to see color shadows and artifacts.

Figure 8: More results from our free-form inpainting system on natural images (1).

Figure 9: More results from our free-form inpainting system on natural images (2).

Figure 10: More results from our free-form inpainting system on faces.