DRB-GAN: A Dynamic ResBlock Generative Adversarial Network for Artistic Style Transfer
– Supplementary Material –

Wenju Xu[1], Chengjiang Long[2*], Ruisheng Wang[3], Guanghui Wang[4*]
[1]OPPO US Research Center, InnoPeak Technology Inc, Palo Alto, CA, USA
[2]JD Finance America Corporation, Mountain View, CA, USA
[3]Department of Geomatics Engineering, University of Calgary, Alberta, Canada
[4]Department of Computer Science, Ryerson University, Toronto, ON, Canada
xuwenju123@gmail.com, cjfykx@gmail.com, ruiswang@ucalgary.ca, wangcs@ryerson.ca

Abstract

The supplementary material provides the configuration of our DRB-GAN networks. We also present more details on the effectiveness of dynamic blocks, consistency on collection style transfer, as well as some additional visualization results. Note that we could not include the material in the main part of the paper due to the space limit.

1. Configuration of Our DRB-GAN Networks

The Dynamic Convolution layer is a 3×3 convolution layer where the weights are produced by the hyper-network. A basic Conv layer contains a 3×3 convolution operation, a batch-normalization, and a ReLU activation sequentially. The learnable encoder is a CNN consisting of 5 Conv layers with stride size 2. The Content Encoder is a CNN having 2 Conv layers with stride size 2. The SW-LIN decoder is a CNN decoder using the SW-LIN to modulate the activations in normalization layers. And the spatial window is to be a center region of size $(H - h, W - w, C)$ given its corresponding feature map of size (H, W, C), where H, W are the spatial resolution and C is the channel dimension. The style code is a concatenation of outputs from fixed VGG and learnable encoder. In our collection discriminator, the feature extractor contains 2 Conv layers with stride size 2 and D network consists of 3 Conv layers with stride size 2.

2. Effectiveness of Dynamic Blocks

The proposed dynamic module is an important network design. We further conduct two experiments. One is to show the advantage of the progressive generation strategy by varying the number of the proposed dynamic blocks, and the other is to explore the advantage of the dynamic block by replacing it with the AdaIN residual block. Quantitative and qualitative results are shown in Table 1 and Figure 1. We observe that the proposed model with 4 blocks works the best. However, only using one dynamic block could outperform the model using AdaIN blocks. This could be attributed to the proposed dynamic block. Therefore, we adopt 4 dynamic blocks as default in our experiments.

Table 1. Quantitative results of using different numbers of dynamic blocks.

Setting	1 DBk	2 DBlks	3 DBlks	4 Dblks	4 AdaIN Blks
Score	0.567	0.570	0.571	0.573	0.558
GPU Memory (MB)	1084	1164	1244	1324	1149

3. Consistency on Collection Style Transfer

Our DRB-GAN model achieves both arbitrary style transfer and collection style transfer. Note that we adopt the DRB-GAN@20 for collection style transfer. The stylizations are listed in Figure 3. We observe that the results of collection style transfer are consistent in stroke size and color, which further demonstrates that our DRB-GAN model has a good capability to transfer domain-level style with very few blocks.

4. More Visualization Results

Style Transfer Involving Multiple Domains. To further understand the advantage of our proposed DRB-GAN, we present additional visualization results in Figure 5. As we can see, our model can efficiently transfer the content image into 10 different styles by only using one unified model.

Performance on Arbitrary New Styles. To clarify, we have evaluated our DRB-GAN model on images in arbi-

*This work was supervised by Chengjiang Long and Guanghui Wang.

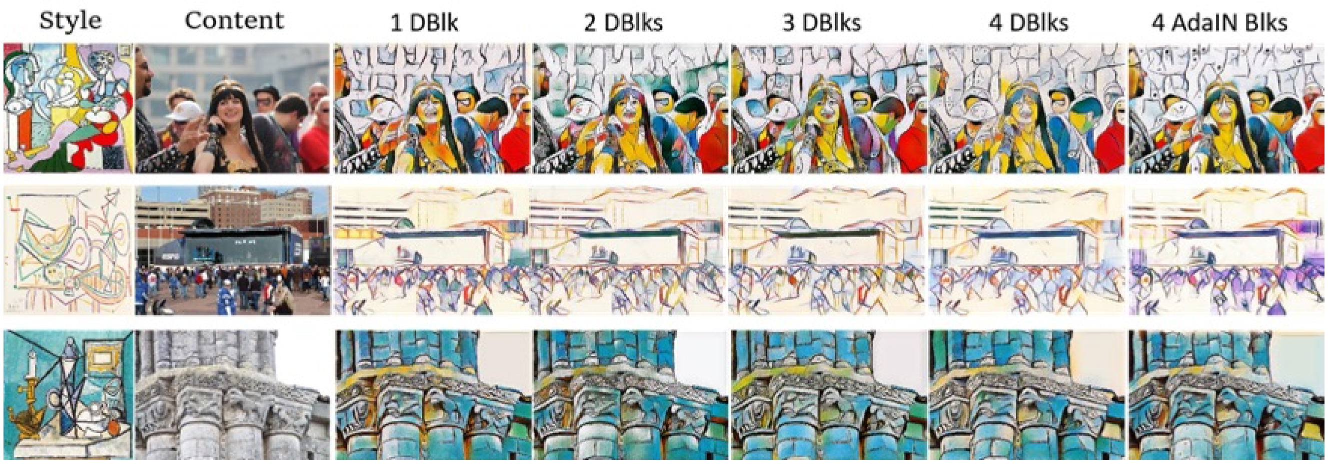

Figure 1. The effect of the Dynamic Blocks

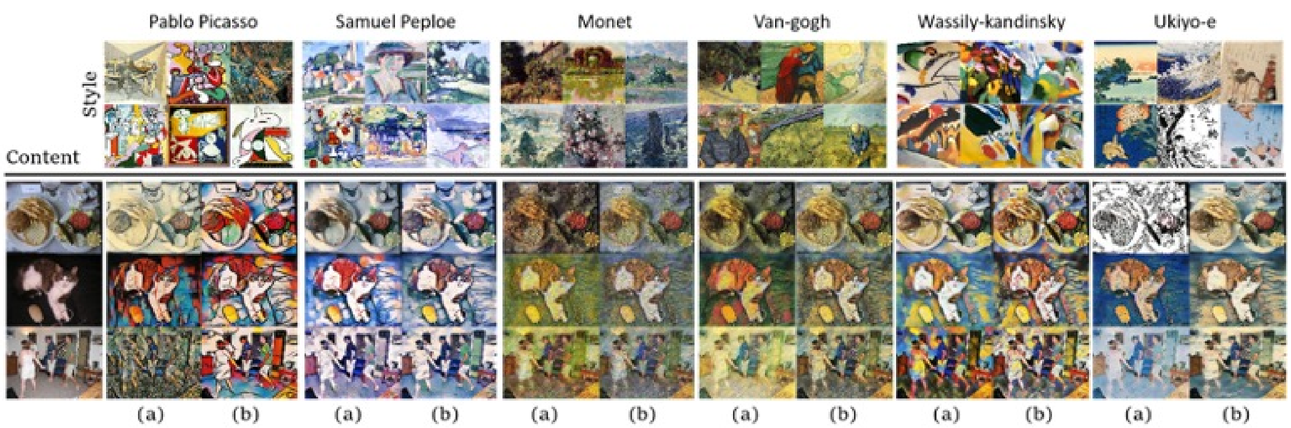

Figure 2. Examples of two types of artistic style transfer: (a) arbitrary style transfer and (b) collection style transfer. Our proposed is able to conduct arbitrary style transfer well and even can extend to handle collection style transfer.

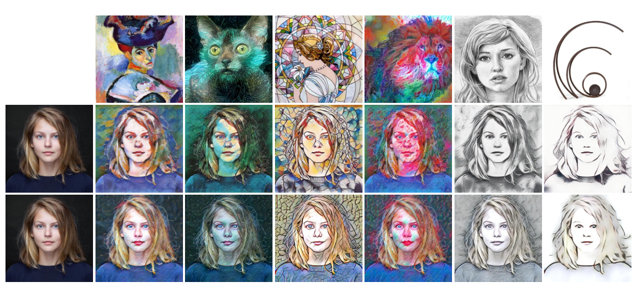

Figure 3. Results on arbitrary new styles. The images in the first column are the content images. Images in the first row are the style images. Images in the second row are the outputs of our DRB-GAN. Images in the third row are the outputs of AdaIN.

trary new styles. As shown in Figure 3, our DRB-GAN model generalizes well for these new styles. It achieves better qualitative performance than AdaIN.

Model Weighted Averaging. As shown in Figure 4, our DRB-GAN model obtains obvious benefits when adding more individual style within one artistic domain, while other arbitrary style transfer method (AdaIN) focus on the single image style. Notably, averaging weights of AdaIN leads to worse performance as the number of used style images increases.

Four-way Style Interpolation. To further illustrate the manifold structure, a four-way style interpolation is shown in Figure 6. The style interpolation is continuous and smooth, and therefore linearly combine the generated parameters is an efficient way to achieve the style interpolation.

High-resolution Image Generation. The proposed DRB-GAN allows us to produce high-quality stylized images in high-resolution. To obtain these synthesized images, the smaller edge of the content images (Figure 8) is resized to 1280 and the aspect ratio is kept. The results are presented in Figure 9 - Figure 13. As we can observe, the stylizations exhibit a lot of fine details such as homogeneous regions and

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 4. Qualitative comparison on the effect of different K values. (a) MetaNet@2, (b) MetaNet@10, (c) MetaNet@20, (d) AdaIN@2, (e) AdaIN@10, (f) AdaIN@20, (g) DRB-GAN@2, (h) DRB-GAN@10, (i) DRB-GAN@20.

Figure 5. Qualitative performance of one unified model on 10 different styles unified model .

smooth color transitions. More importantly, the structure similarity is well preserved.

Figure 6. Four-way style interpolation across multiple domains. By interpolate the generated weights, we can interpolate between arbitrary styles. The style and content images are listed in the last row for reference.

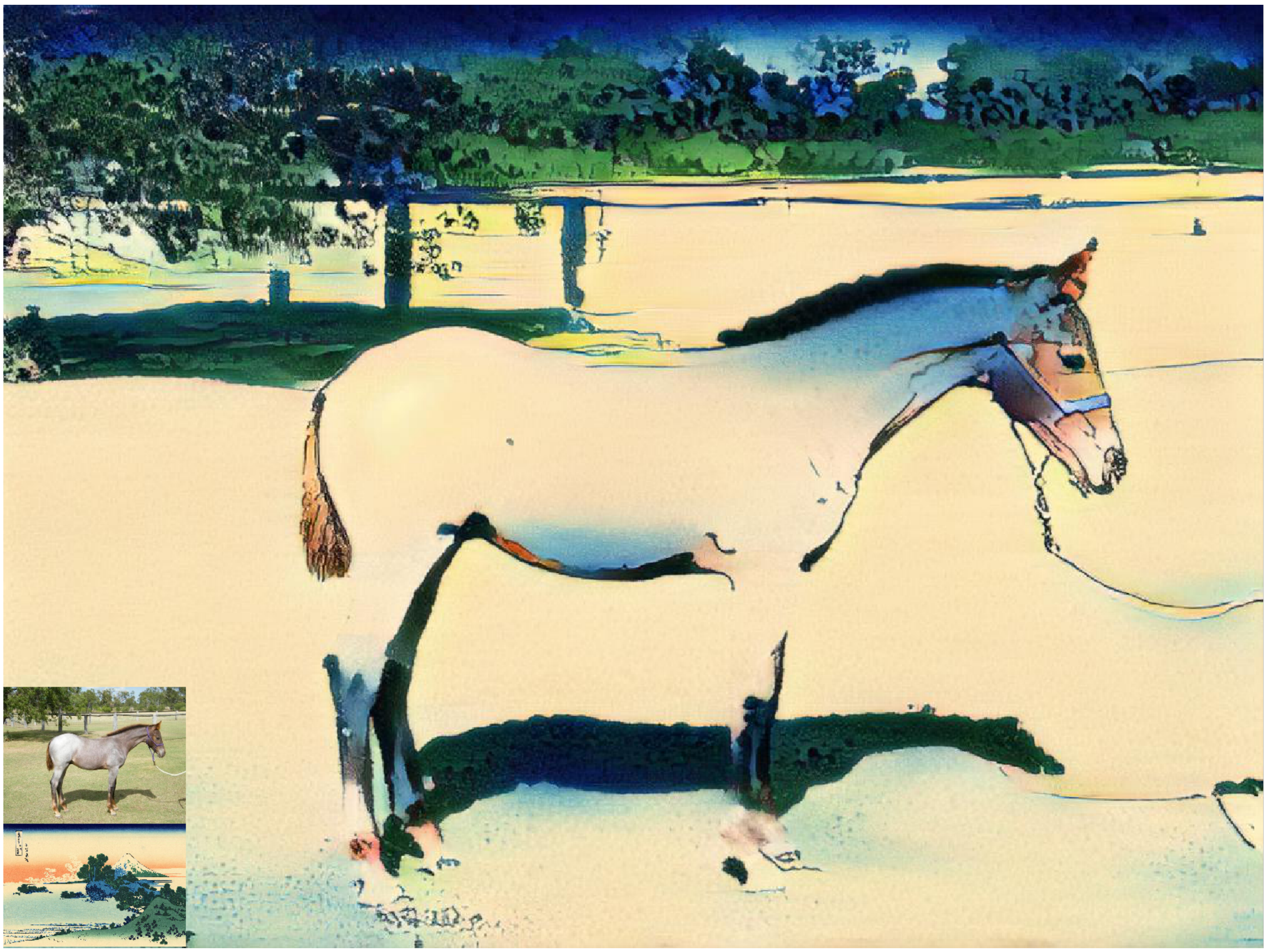

Figure 7. An example of stylization in Ukiyo-e style

Figure 8. The content images used for image style transfer. The text at the bottom indicates the target style. The smaller edge of the images will be resized to 1280 pixel for high-resolution image generation.

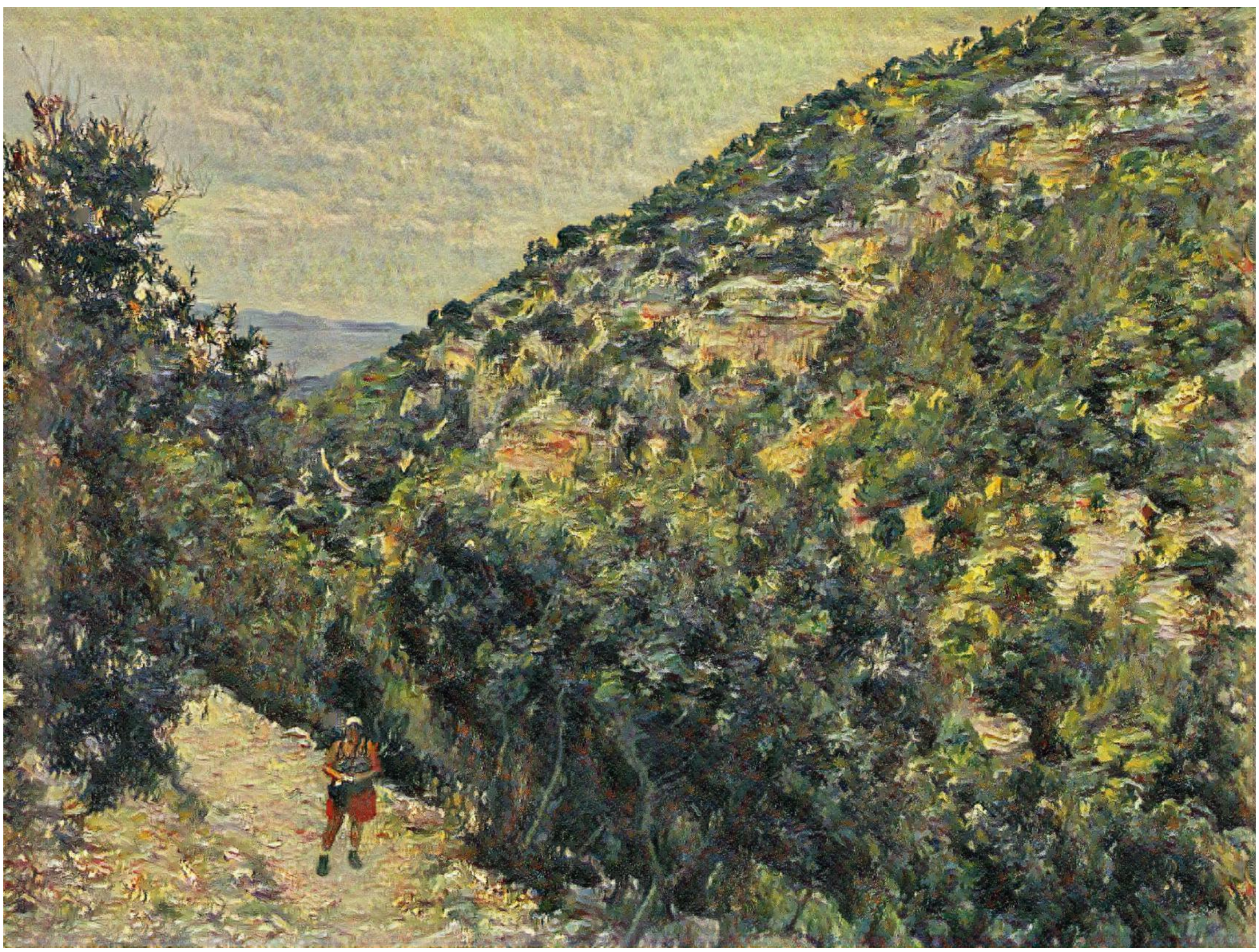

Figure 9. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (Monet).

Figure 10. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (Morisot)

Figure 11. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (Picasso)

Figure 12. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (Kirchner)

Figure 13. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (Kandinsky)

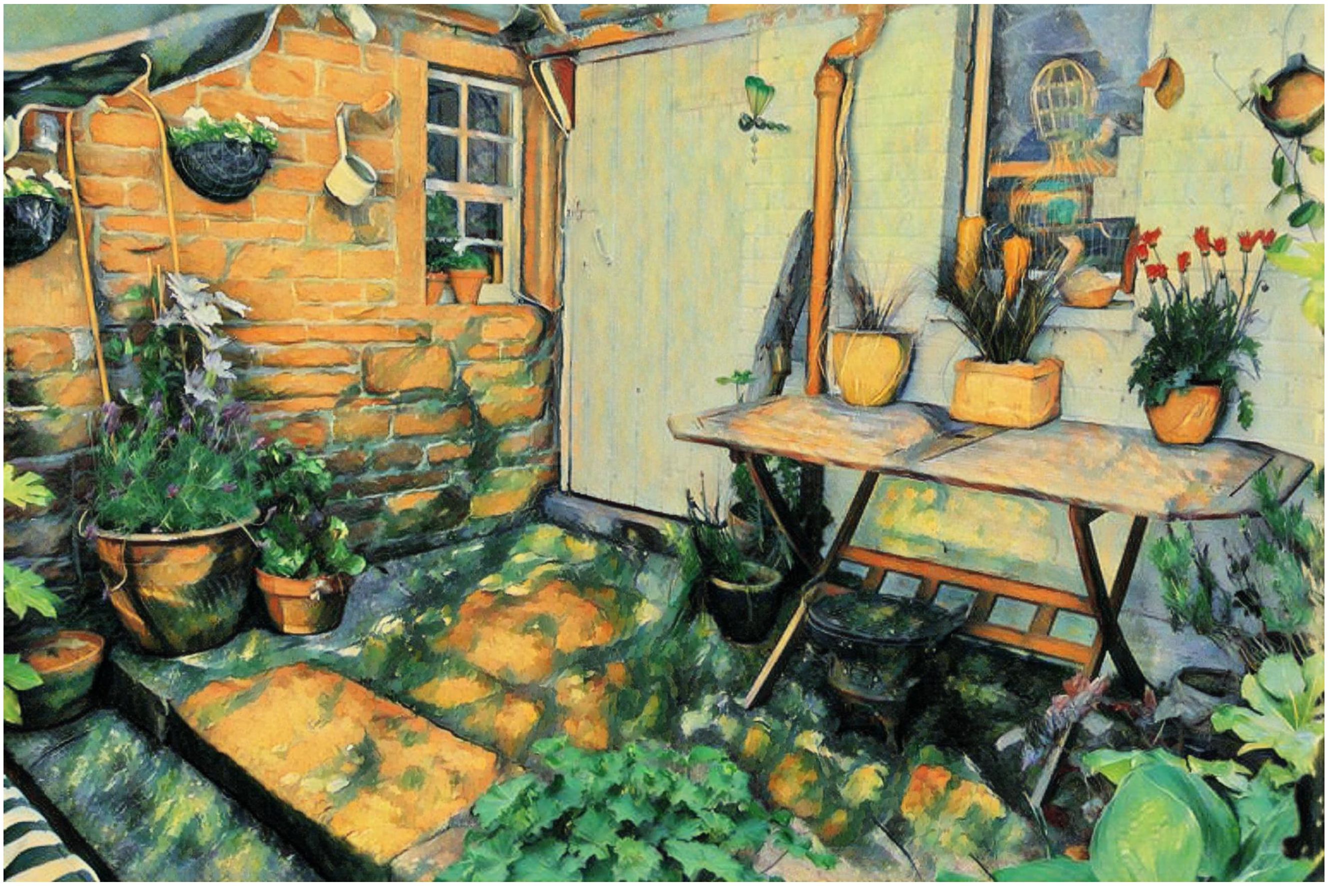

Figure 14. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (Cezanne)

Figure 15. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (Gauguin)

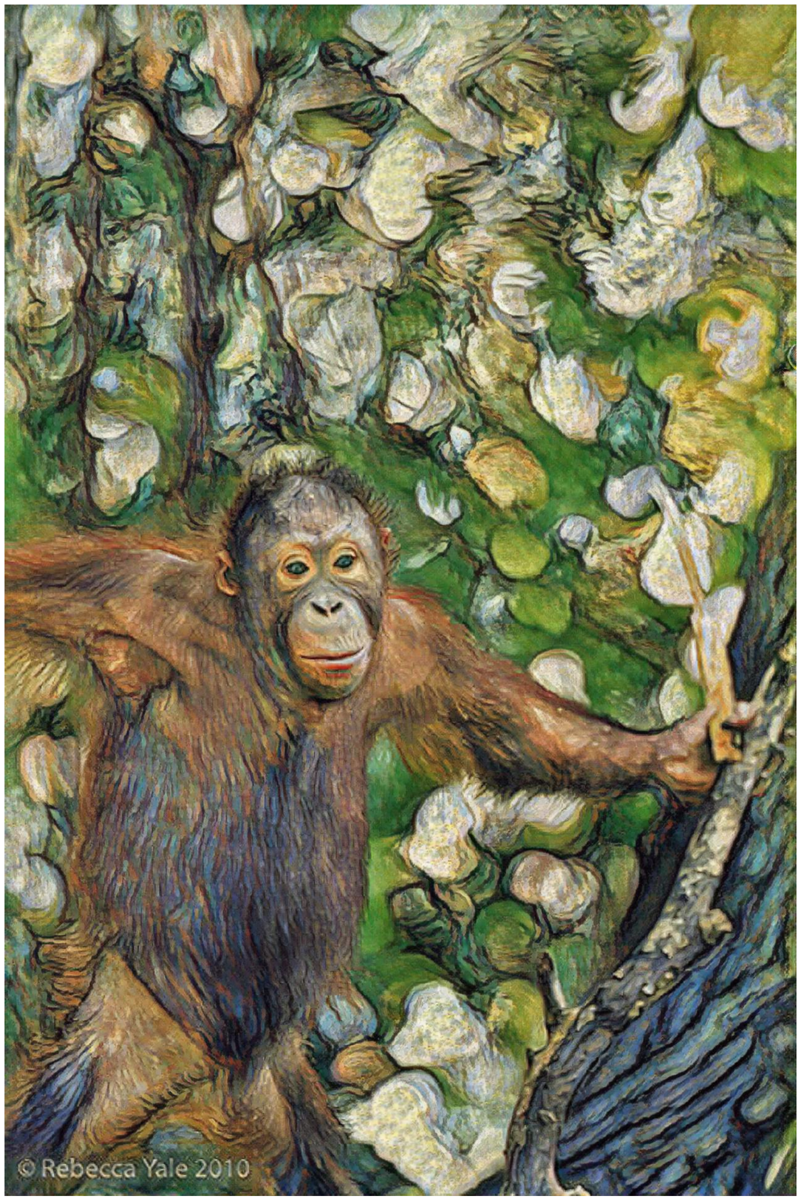

Figure 16. High-resolution image generated by our model. The color variations, scale and stroke size are visible. (van Gogh)