A Content Transformation Block For Image Style Transfer
- Supplementary Material -

Dmytro Kotovenko Artsiom Sanakoyeu Pingchuan Ma Sabine Lang Björn Ommer
Heidelberg Collaboratory for Image Processing, IWR, Heidelberg University

1. Additional Visual Comparison

In this supplementary material, we present additional comparisons with existing style transfer methods for the following 8 artists: Berthe Morisot, Claude Monet, Ernst Ludwig Kirchner, Pablo Picasso, Paul Cezanne, Paul Gauguin, Vincent van Gogh, and Wassily Kandinsky. Comparisons are presented in Fig. S1 and Fig. S2. We observe that while providing better stylization than the state-of-the-art AST [3] method, we also retain the content of images better and produce no artifacts; please zoom in for details. All results are generated in resolution with 1280 pixels as the minimal side of the image.

We also stylized two random videos from the internet to show that our method is able to produce real-time high definition stylization of videos, also free of flickering. As input we took two fragments, each 3 minutes long, from the video Provence: Legendary Light, Wind, and Wine at timepoints 7:02 and 10:10 and one entire video Chaplin Modern Times-Factory Scene (late afternoon). For best viewing experience, please watch all videos in 4K resolution, since the quality drops significantly due to YouTube's compression algorithms otherwise. For video stylization, we provide a comparison between our method and the AST [3]. In addition, to visualize the necessity of the Content Transformation Block $\mathcal{T}$, we run a side-by-side stylization of our model with and without extra training of $\mathcal{T}$ block. We notice that there is a difference in the way how content is retained and also how parts of the image are highlighted. Our model with block $\mathcal{T}$ achieves better preservation of human figures, especially at smaller scale. The links to the playlists: 1, fragment 2 and fragment 3.

2. Implementation Details

2.1. Network Architecture Notation

Our generator network consists of three consequent blocks: encoder E, content transformation block $\mathcal{T}$ and decoder D. Besides that, we have two discriminators: $\mathcal{D}_c$ and $\mathcal{D}_s$. For brevity we use the following naming conventions:

* conv-$k \times k$-stride-s denotes a convolutional layer with kernel size $k \times k$ and stride s;
* LFN-G-g-W-w denotes a Local Feature Normalization Layer group size g and window size w;
* upscale-3×3 denotes an upscaling layer that consists of nearest neighbor; upscaling is done by a factor of 2, followed by a convolutional layer with kernel size 3×3 and stride 1.
* ResBlock-3×3 denotes a residual block that consists of two convolutional layers with kernel size 3×3 and stride 1 followed by LFN-G-32-W-32;
* cfss-k-LFN-g-w-LReLU denote a $f \times f$ convolution with stride s and k filters followed by LFN-G-g-W-w layer and LReLU with slope 0.2 ;

All convolutional layers use reflection padding.

We describe the architecture of the encoder and the decoder in Tab. S1.

2.1.1 Content Transformation Block $\mathcal{T}$

Content transformation block $\mathcal{T}$ consists of 9 consequent residual blocks ResBlock-3×3 with each convolution having 256 kernels.

2.1.2 Architecture of the Discriminators $\mathcal{D}_s$ and $\mathcal{D}_c$

Both discriminators described in Tab. S2 have a double purpose: predicting the class of the input image and predicting domain of the image (real painting or not). On the one hand, the discriminator $\mathcal{D}_s$ takes images as an input and predicts scene class and domain. The discriminator $\mathcal{D}_c$, on the other hand, takes a feature vector as an input and predicts class (person/non-person) and domain. Both predictions are given as two values in the last line of $\mathcal{D}_c$ architecture in Tab. S2.

For the discriminator $\mathcal{D}_s$ conditions are more complicated: a class of the image scene is predicted in the final layer, see Tab. S2. To obtain domain predictions, we attach a convolutional layer of one single kernel to the outputs of

Encoder	Decoder
Input ($256 \times 256 \times 3$ image)	-
`conv-`3×3`-stride-1`	`conv-`3×3`-stride-1`
`LFN-G-32-W-128`	(`ResBlock-`3×3) $\times 9$
`conv-`3×3`-stride-2`	`upscale-`3×3
`LFN-G-32-W-64`	`LFN-G-32-W-16`
`conv-`3×3`-stride-2`	`upscale-`3×3
`LFN-G-32-W-32`	`LFN-G-32-W-32`
`conv-`3×3`-stride-2`	`upscale-`3×3
`LFN-G-32-W-32`	`LFN-G-32-W-64`
`conv-`3×3`-stride-2`	`upscale-`3×3
`LFN-G-32-W-16`	`LFN-G-32-W-128`
	`conv-`7×7`-stride-1`
	sigmoid

Table S1. Description of the encoder and the decoder architecture. ReLU layers are omitted for brevity.

the 4th and the 5th convolutional layers of the discriminator and compute average mean on its outputs. This value is used as domain prediction.

2.2. Training Details

Training process consists of two stages: a randomly initialized network is trained at first on patches of size 256×256 pix cropped from the real paintings and patches cropped from the photographs with scene class label for 400000 iterations with batch size 8. Afterwards, we continue training procedure on patches of size 768×768 pix for another 400000 iterations with batch size 1 on the two aforementioned datasets. At this stage, we also train on patches of person and non-person class extracted from both paintings and photographs dataset. At each training stage we use two different Adam [2] optimizers both with learning rate 2×10^{-4}: one for discriminators $\mathcal{D}_s$, $\mathcal{D}_c$ and another for encoder E, transformation block $\mathcal{T}$ and decoder D. To avoid that the generator is incapable of fooling the discriminator, we impose a constraint that the discriminator $\mathcal{D}_s$ wins in 80% of the cases [1, 3]. To achieve this, we compute a running average of the discriminator's $\mathcal{D}_s$ accuracy; if the accuracy is < 0.8 we update the discriminator $\mathcal{D}_s$, otherwise we update the generator.

D_C	D_S
Input ($16 \times 16 \times 256$ tensor)	Input ($256 \times 256 \times 3$ image)
global avg pooling	c5s2-128-LFN-32-128-LReLU
fc-512-ReLU	c5s2-128-LFN-32-64-LReLU
fc-512-ReLU	c5s2-256-LFN-32-32-LReLU
fc-2, fc-2	c5s2-512-LFN-32-16-LReLU
	c5s2-512-LFN-32-8-LReLU
	c5s2-1024-LFN-32-4-LReLU
	c5s2-1024-LFN-32-4-LReLU
	conv-6×6-stride-1-LReLU
	max_pool
	fc-num_classes

Table S2. Description of the architecture of discriminators D_S and D_C.

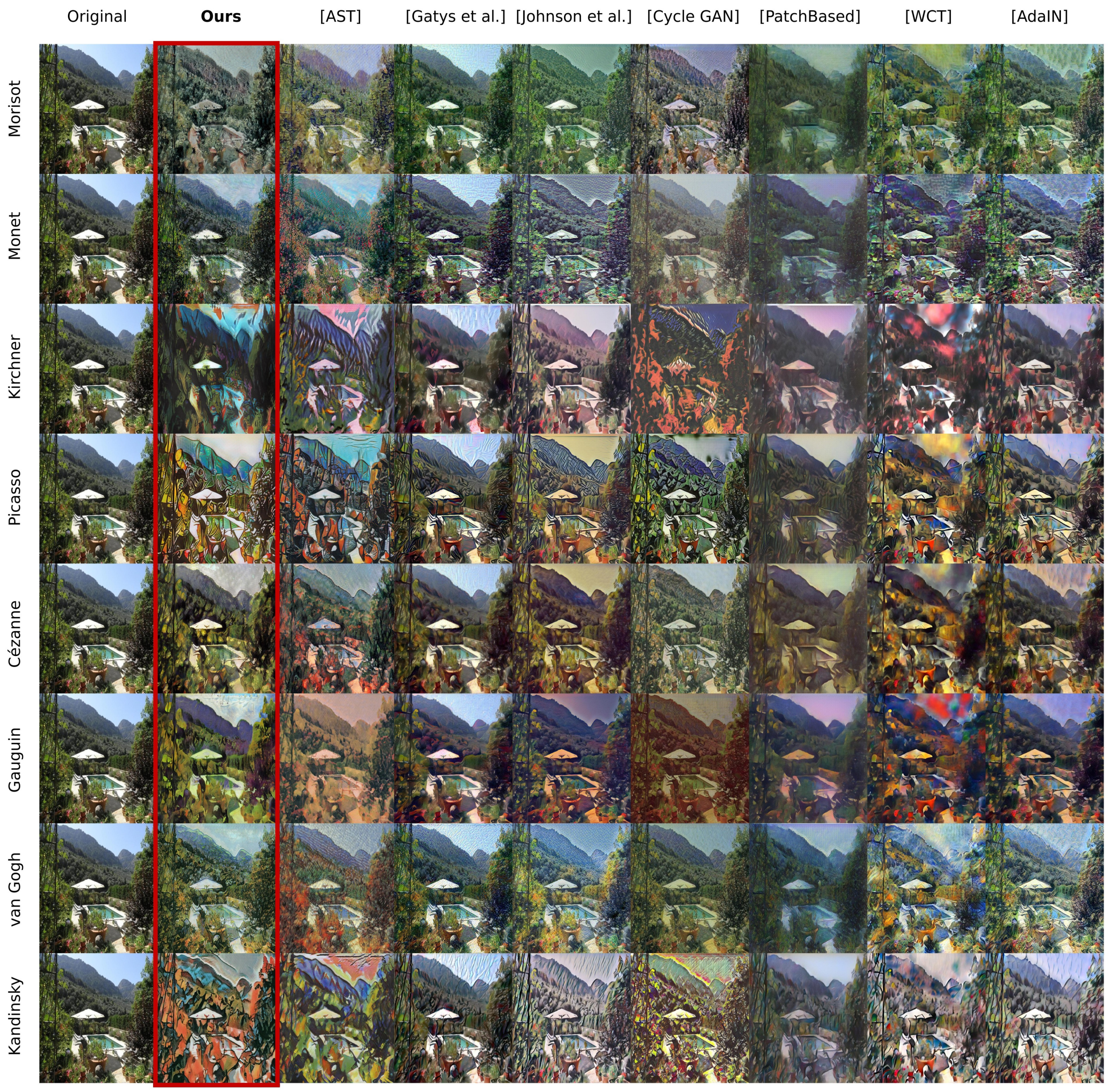

Figure S1. Comparison between our and other methods on full images with the same content for different artists.

0

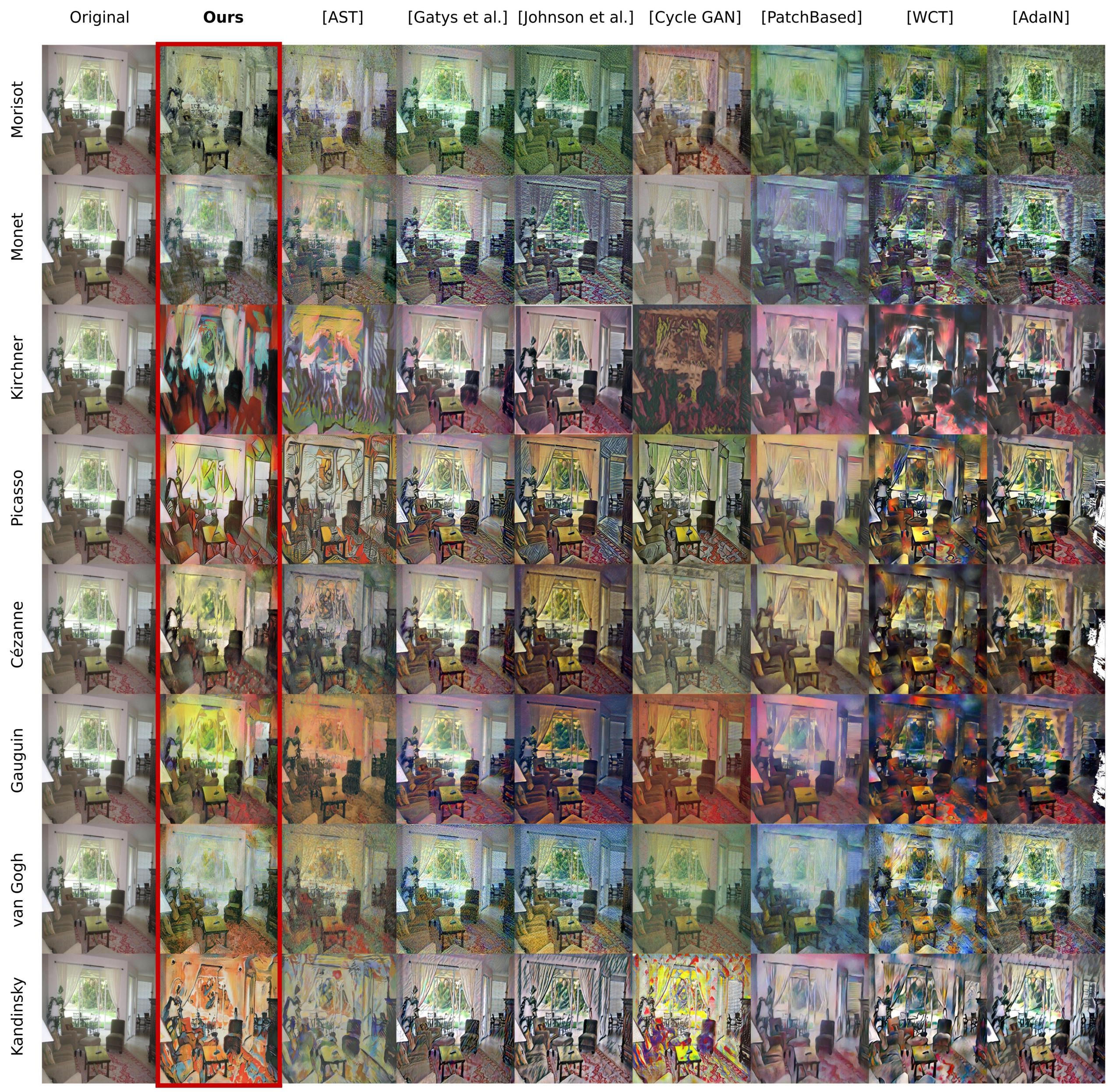

Figure S2. Comparison between our and other methods on full images with the same content for different artists.

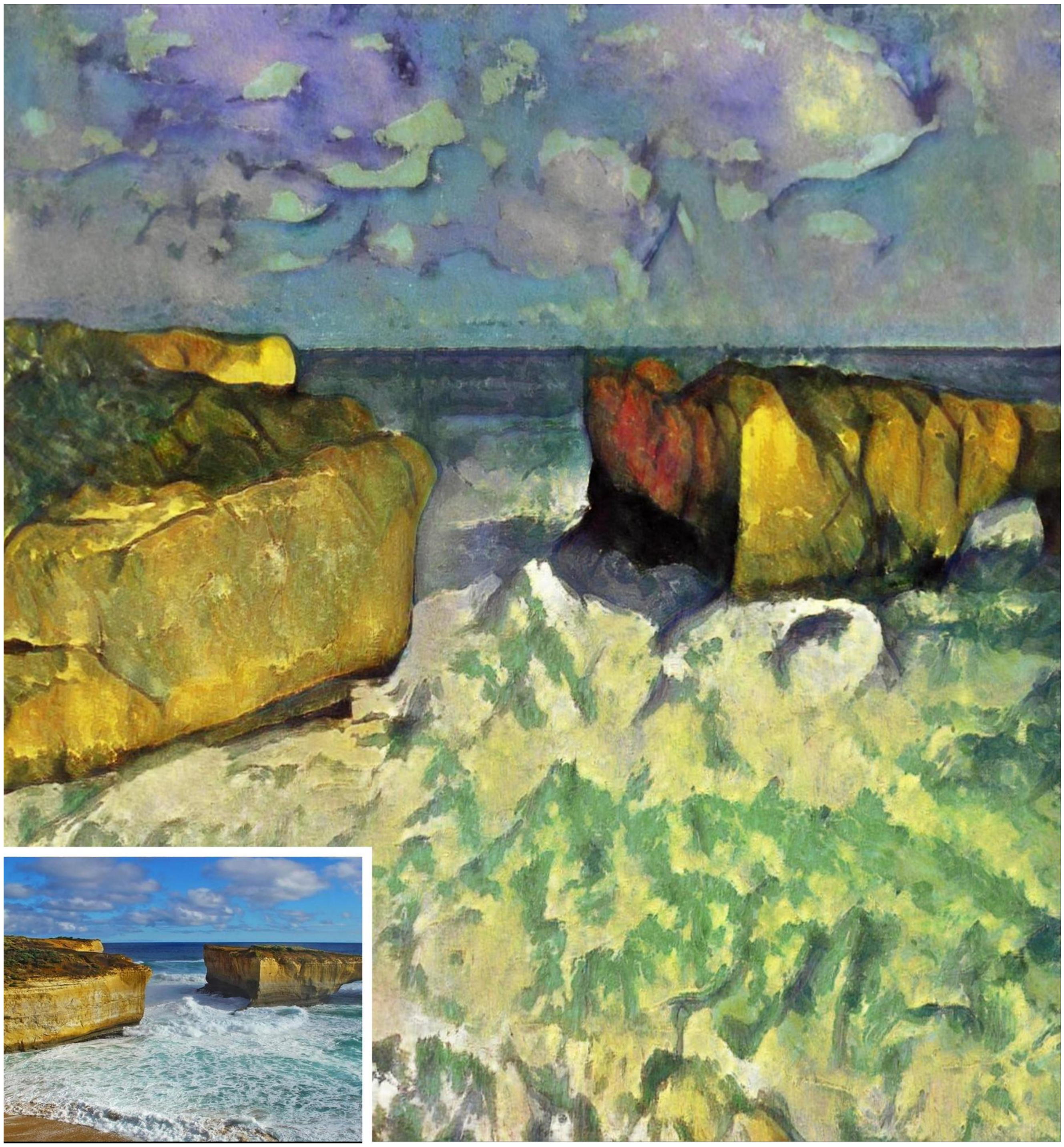

Figure S3. Image stylization for Paul Cezanne using our approach.

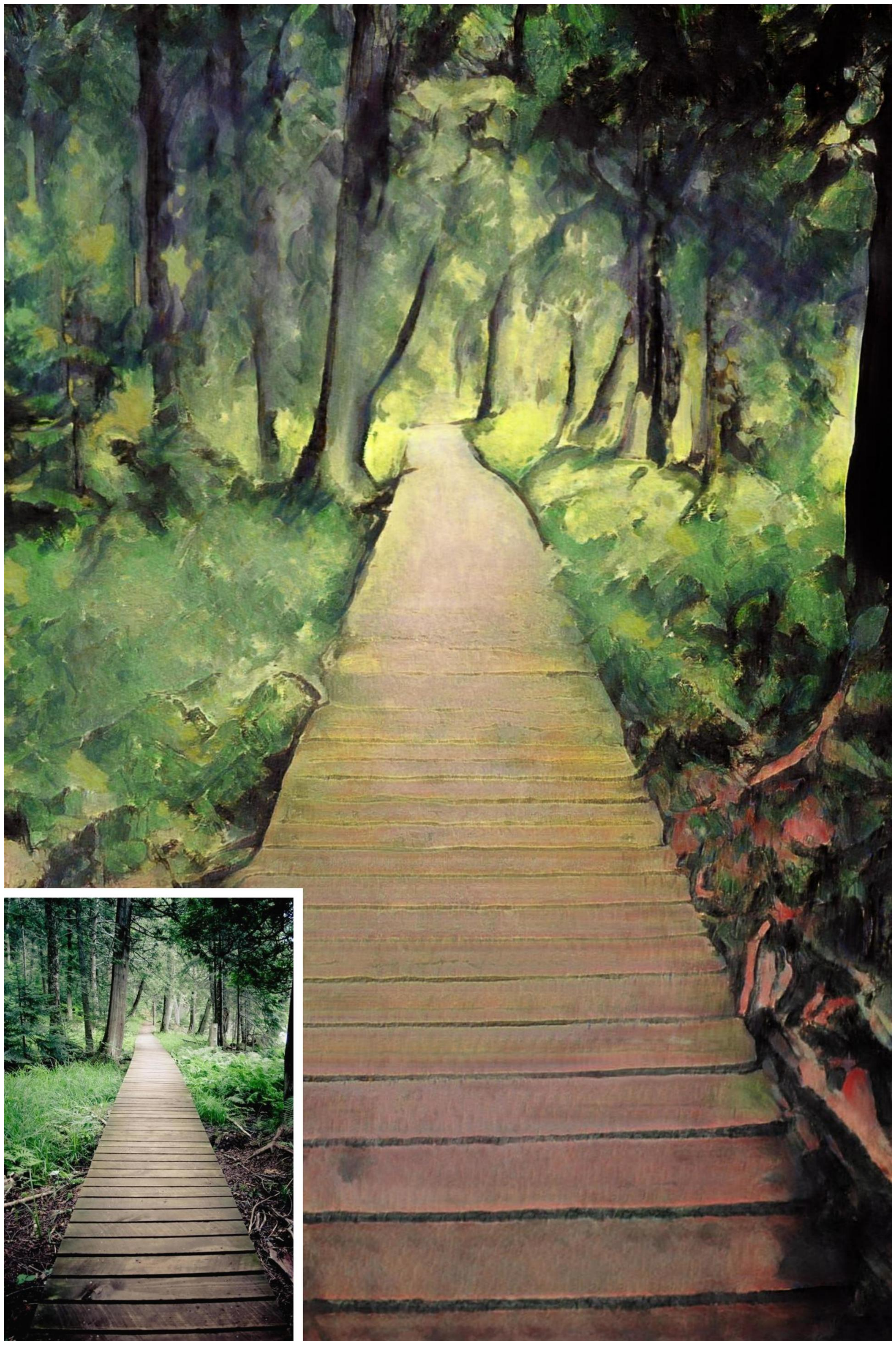

Figure S4. Image stylization for Paul Cezanne using our approach.

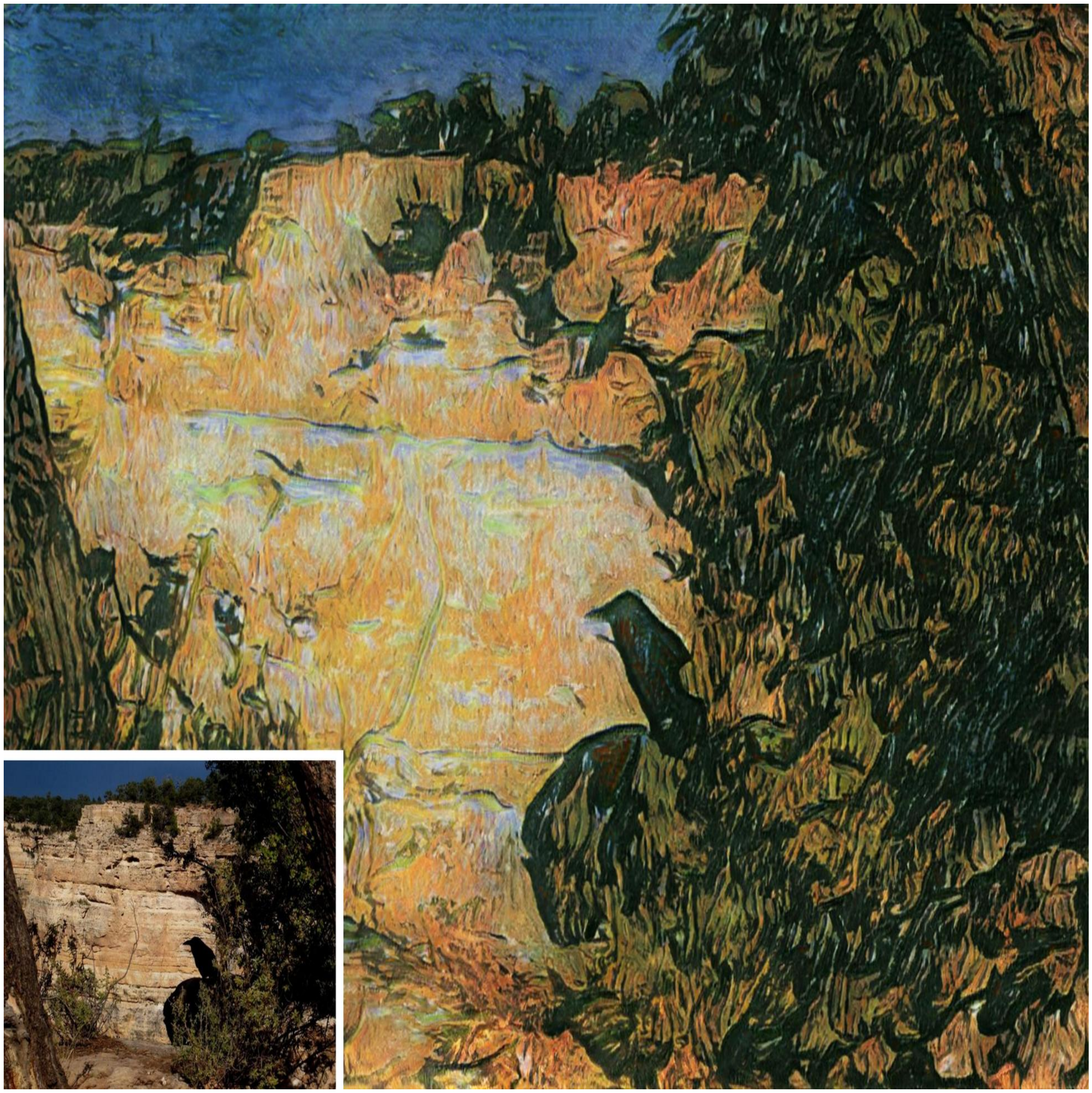
Figure S5. Image stylization for Vincent van Gogh using our approach.

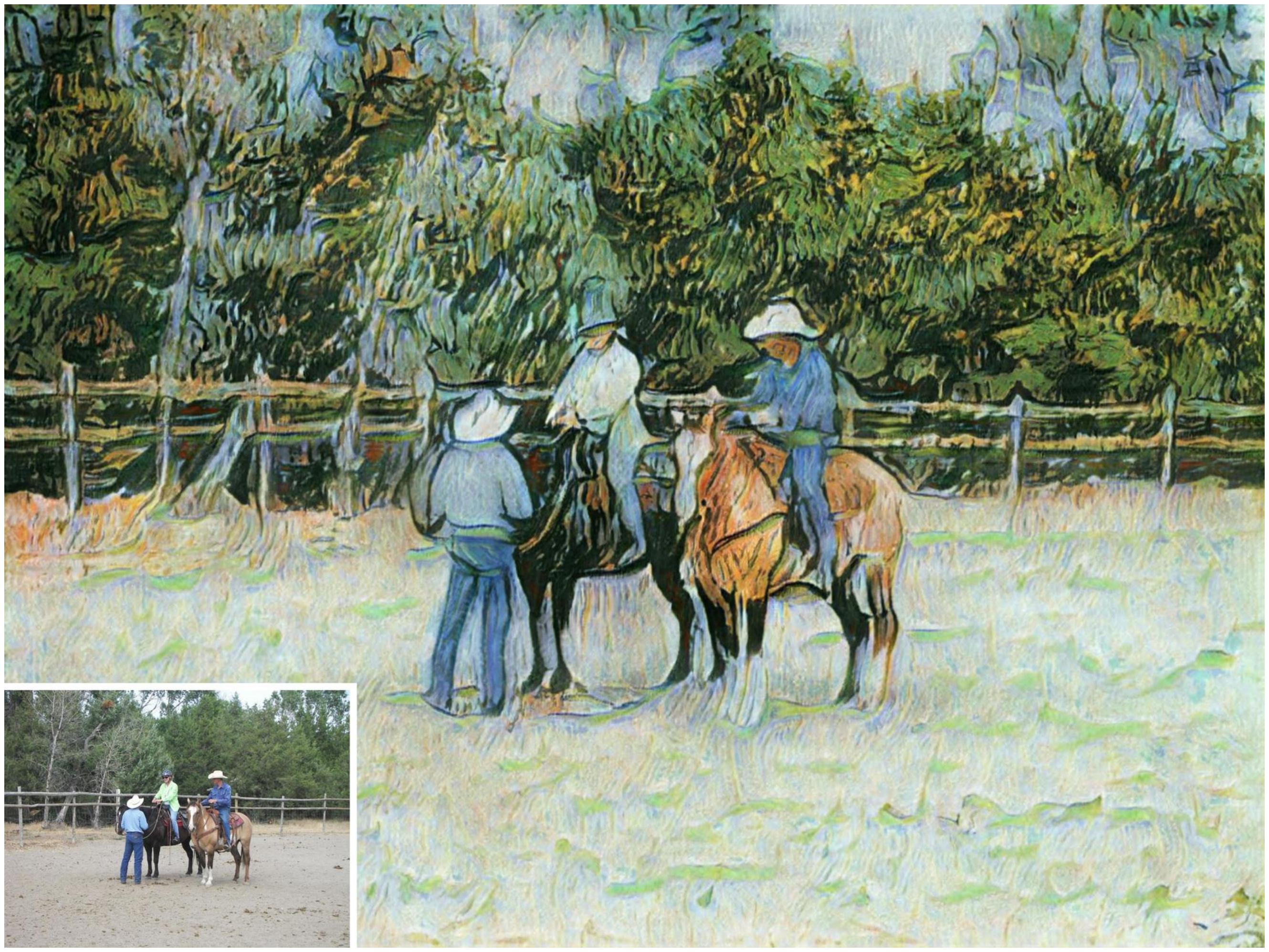
Figure S6. Image stylization for Vincent van Gogh using our approach.

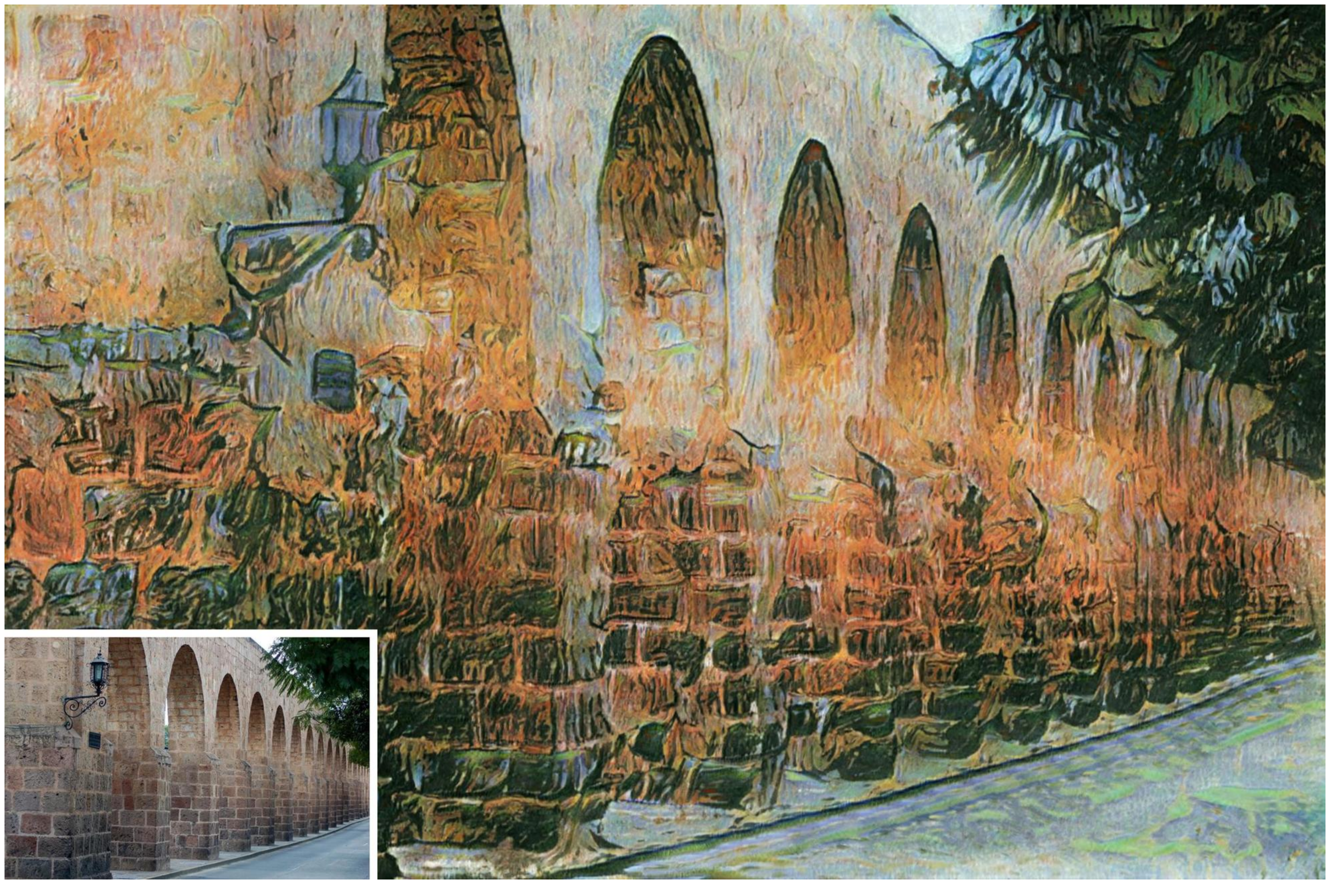

Figure S7. Image stylization for Vincent van Gogh using our approach.

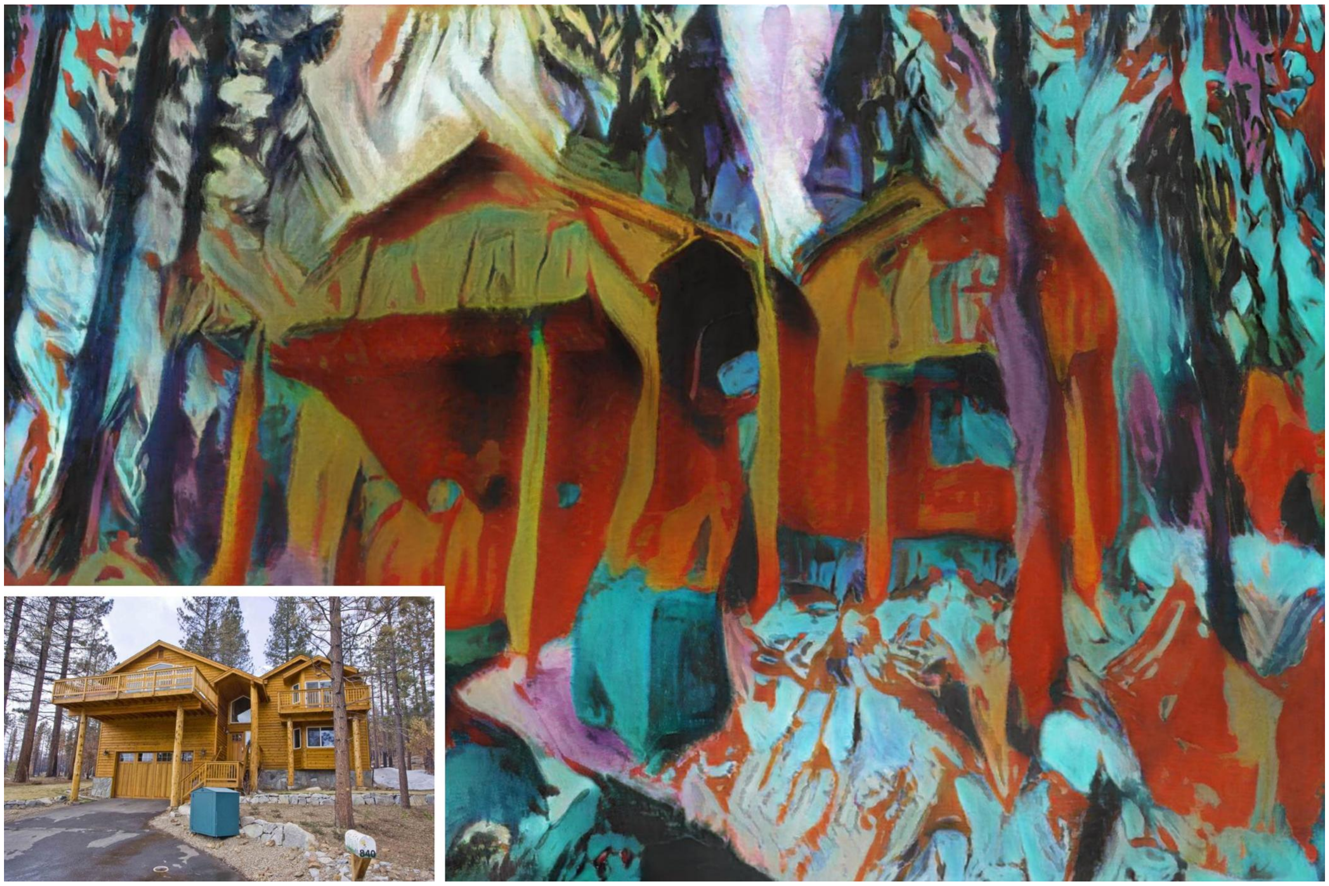
Figure S8. Image stylization for Ernst Ludwig Kirchner using our approach.

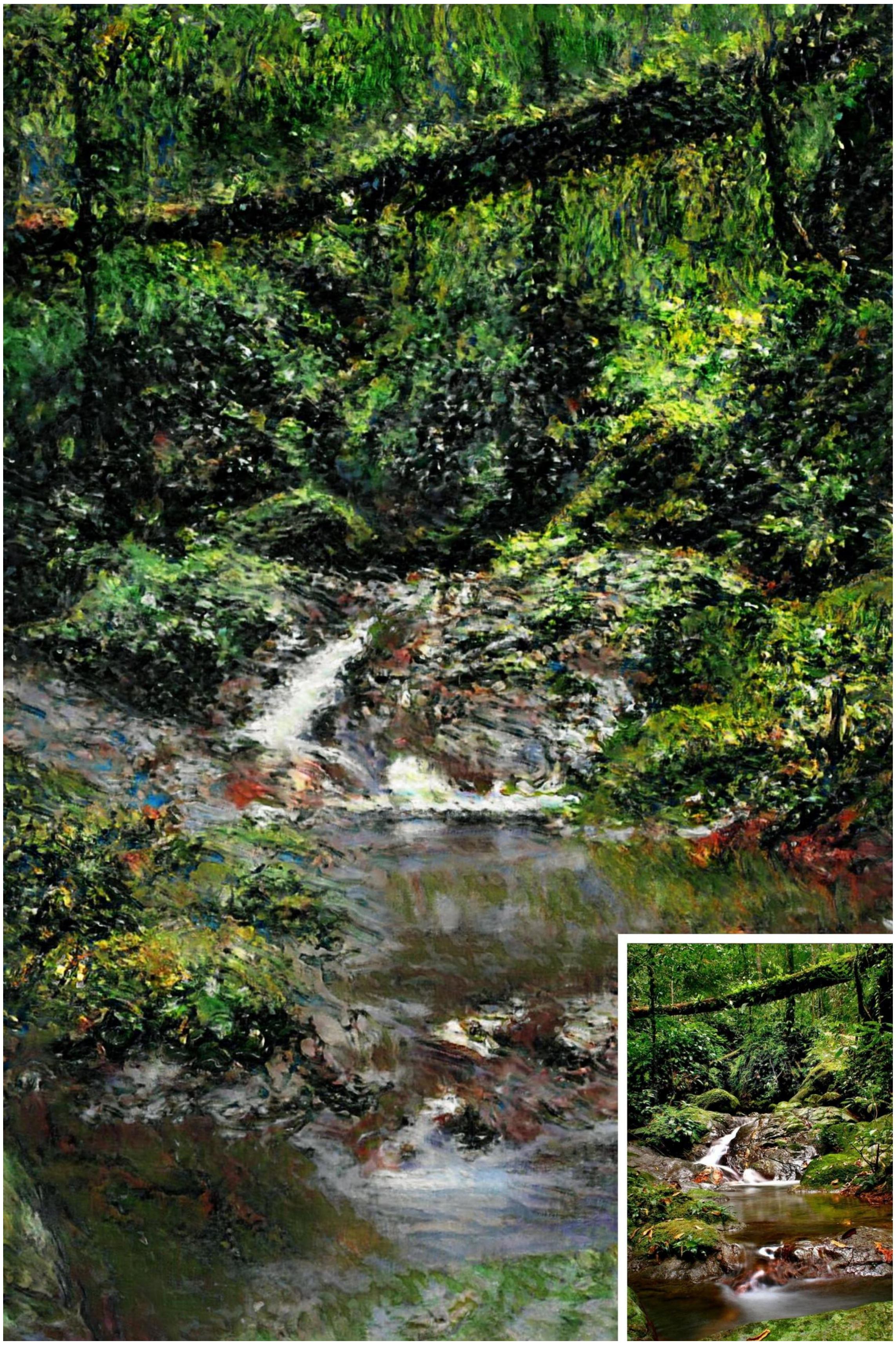

Figure S9. Image stylization for Claude Monet using our approach.

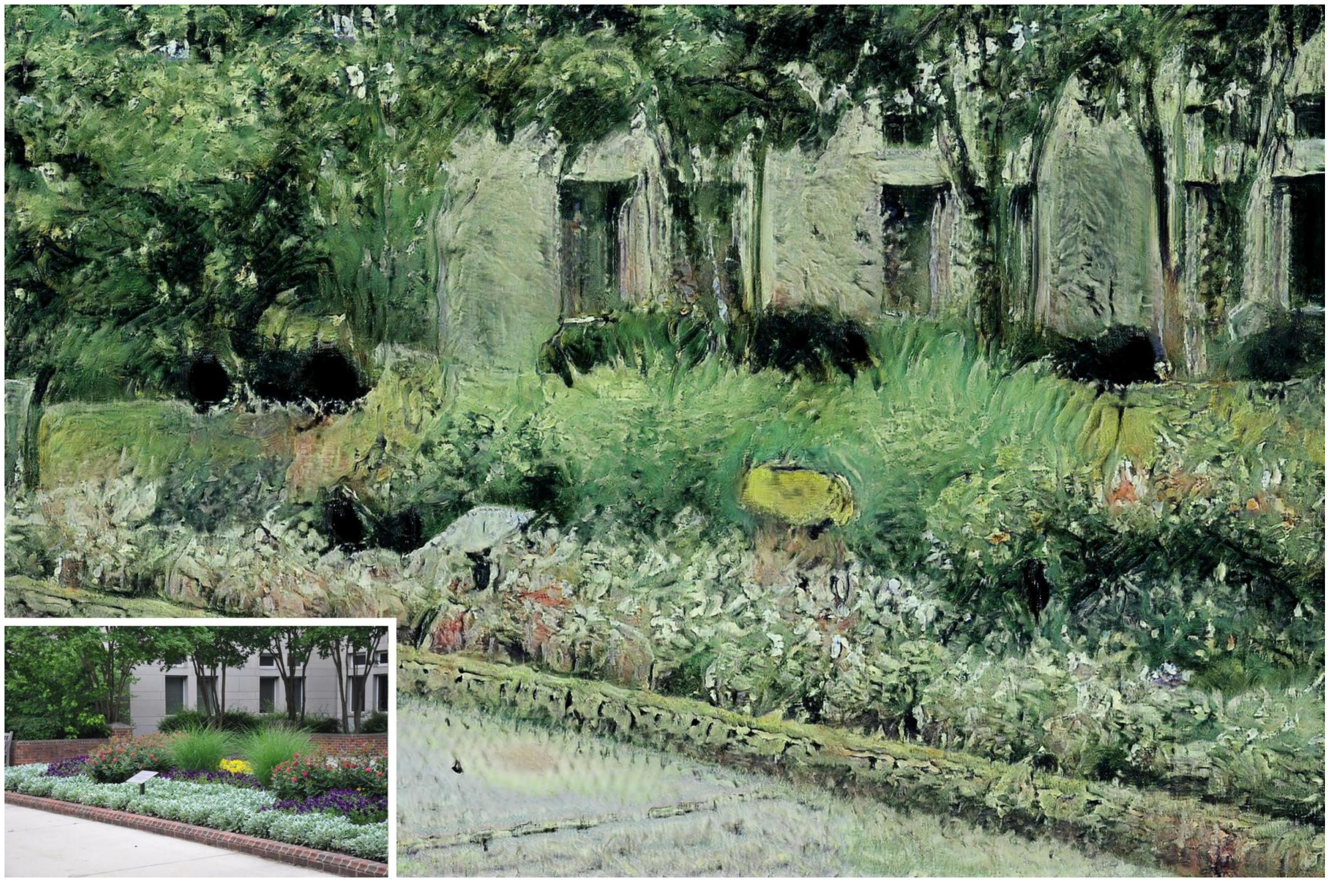
Figure S10. Image stylization for Berthe Morisot using our approach.

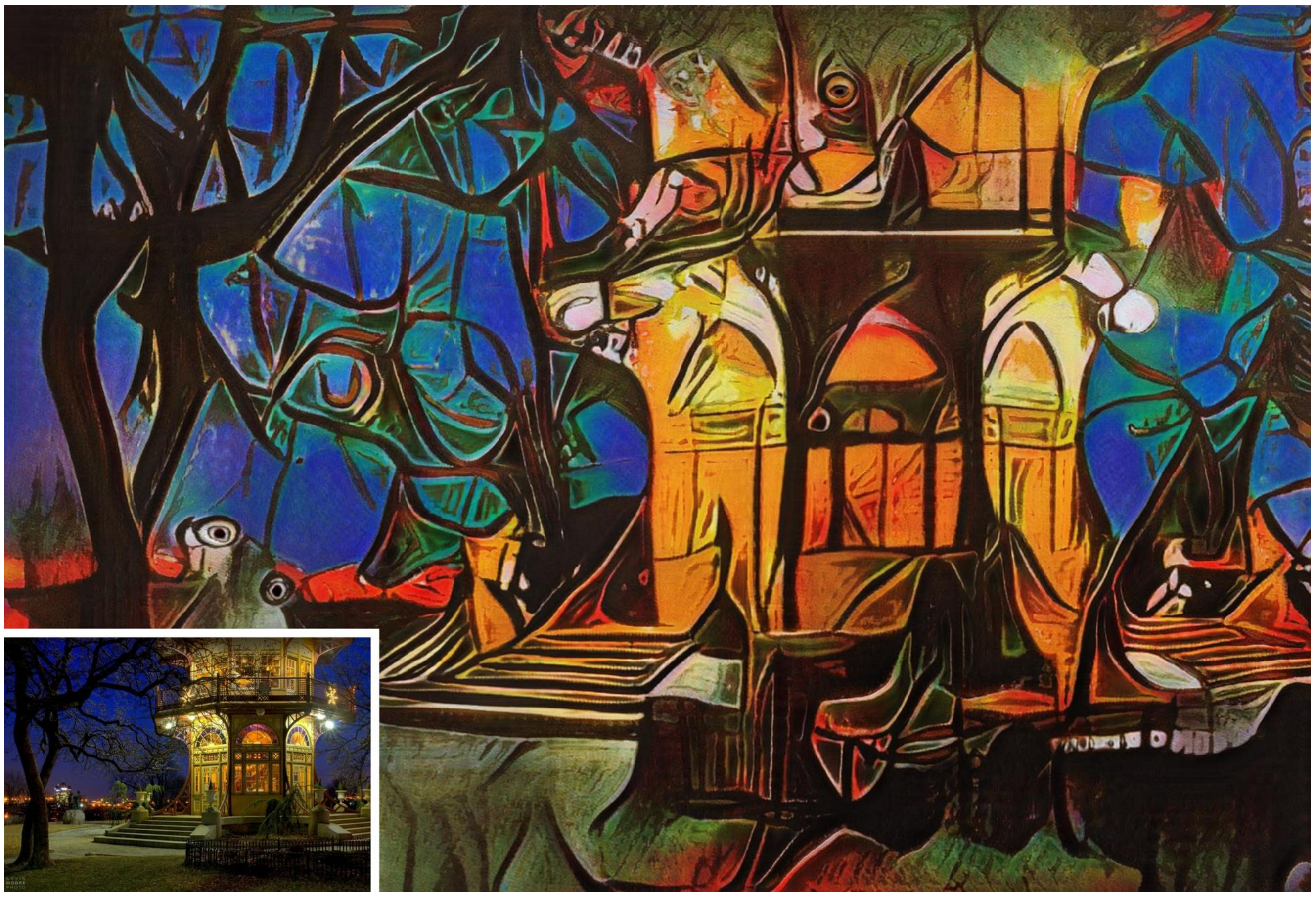

Figure S11. Image stylization for Pablo Picasso using our approach.

References

[1] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018. 2

[2] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[3] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer. A style-aware content loss for real-time hd style transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1, 2