Content and Style Disentanglement for Artistic Style Transfer
- Supplementary Material -

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

1. Training Details

Since we use an adversarial approach to train our model we utilize two competing networks: generator $\mathcal{G}$ and discriminator $\mathcal{D}$.

The generator $\mathcal{G}$ consists of three blocks: content encoder E_c, style encoder E_s and decoder D as depicted in the model diagram in Fig. 2 in the main submission. The parts E_c and D form a single fully-convolutional network, where decoder D is guided by the style vector $E_s(y)$ obtained from an input style image y by the style encoder E_s. The discriminator $\mathcal{D}$ is a fully-convolutional network producing spatial probability maps for image patches being real or fake (similar to PatchGAN [3]), and predicts a style label for the entire image.

For training, we use two datasets: Wikiart [2] dataset to sample real artworks with the artist label and the Places365 [6] dataset containing high-resolution content images.

The training is performed by alternating between the discriminator and generator steps based on the win ratio of the discriminator. We force the discriminator to win with a chance of 0.9; this approach has demonstrated a great success in [4].

Discriminator step. We construct a batch of images consisting of real artworks and generated images. Every real artwork has a ground truth artist label. We only update the discriminator's $\mathcal{D}$ weights to minimize the $\mathcal{L}_{adv}$.

Generator step. The batch for a single generator training step consists of tuples of elements $(y_1, s_1, y_2, s_2, c_1, c_2)$: y_1 - first style image (input artwork), s_1 - first style artist label, y_2 - second style image, s_2 - second style artist label, c_1,c_2 - first and second content images (input photos). Then the generator produces stylizations $(c_1|y_1), (c_1|y_2), (c_2|y_1), (c_2|y_2)$, where $(c_i|y_j)$ corresponds to the content c_i generated in the style of y_j.

To compute the $\mathcal{L}_{FPT-style}$ loss we use the triplet $(y_1, (c_1|y_1), (c_1|y_2))$. And to compute the $\mathcal{L}_{FPD}$ loss we use the triplet $(y_2, (c_1|y_2), (c_2|y_2))$. Other losses such as $\mathcal{L}_{pixel}$ and $\mathcal{L}_{FP-content}$ are straightforward to compute. We also aim to maximize the $\mathcal{L}_{adv}$ loss computed by the discriminator $\mathcal{D}$ while minimizing the style classification loss of the discriminator on stylized images.

To ensure reproducibility, we will publish the source code and pretrained models after the acceptance of the paper.

2. Qualitative Results

We provide additional stylizations for various artists and contents produced by our model. Generated images have a minimum side size of 1280 pixels. For comparison we also included the respective input content image. To demonstrate the performance and flexibility of our model we stylize four videos taken from the internet* in the styles of different artists. The videos show that our model generates smooth video stylizations with almost no flickering. Moreover, we apply our model to the input video in a frame-by-frame manner without any temporal smoothing or postprocessing. Please watch the stylized videos in 4K resolution for a better visual experience. Additional videos and other results could be found on our project page†.

In Fig. S1, we also provide an extended version of the patch quiz presented in the main paper. However, this time each row contains crop outs from real paintings, from stylizations obtained by our method and by other models, namely AST[4], Gatys et al[1] and CycleGAN[7].

3. Qualitative Ablations

To illustrate the influence of the fixpoint triplet style loss $\mathcal{L}_{FPT-style}$ we train our model without one. As has been shown quantitatively in the main paper, this model performs much worse on approximating the original artists style distribution and also has a lower deception rate. In the supplementary material, we illustrate the influence on the result, namely the stylized image: the model without the $\mathcal{L}_{FPT-style}$ loss cannot adapt to fine details of a query style sample and produces the same stylizations for different style images taken from the same artist. This is depicted in Fig. S2 and S3.

* Amalfi Coast Vacation Travel Guide — Expedia,
LISBON Tourism Ad Film,
Barcelona in 4K

† compvis.github.io/content-style-disentangled-ST/

Figure S1. This table highlights the similarity between our generated stylizations and original paintings in comparison to other existing methods. Each row contains patches from real artworks, stylizations obtained by our method and patches from stylized images produced by other methods: AST[4], Gatys et al.[1] or CycleGAN[7]. Try to guess which patch is generated and which is real. Answers are given on the very last page.

4. Style Embedding Space

In this section, we investigate the style embedding space to illustrate which artworks our network considers to be similar or dissimilar. Since the style space has a very high dimensionality we apply common dimensionality reduction techniques like PCA and t-SNE[5]. The embedding's projections for paintings by Vincent van Gogh and Paul Cezanne are depicted in figure S4. We observe that the style samples constitute a 1-dimensional manifold, where one half is dominated by the artworks of van Gogh and the other by artworks of Cezanne. In the middle part, however, the style vectors are slightly mixed. This behaviour is due to the fixpoint style triplet loss; without one we would have obtained two convergence centers - one for each artist.

Figure S2. The first row shows that using a model without the $\mathcal{L}_{FPT-style}$ loss for stylization produces identical stylized images even when taking different query style samples from van Gogh. In the second row, we illustrate that by including a $\mathcal{L}_{FPT-style}$ loss we obtain different stylizations for different query style samples.

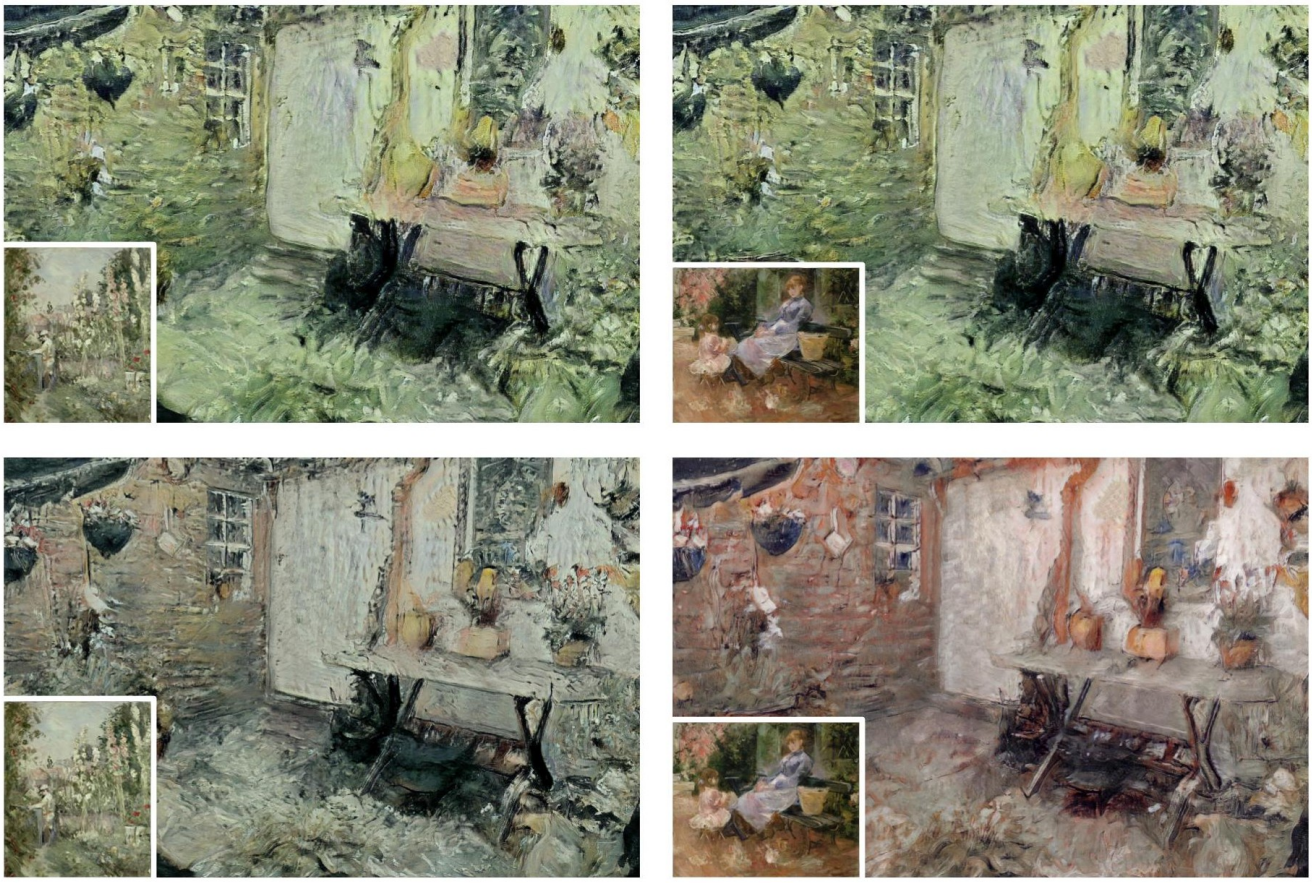

Figure S3. The first row shows that using a model without the $\mathcal{L}_{FPT-style}$ loss for stylization produces identical stylized images even when taking different query style samples from Morisot. In the second row, we illustrate that by including a $\mathcal{L}_{FPT-style}$ loss we obtain different stylizations for different query style samples.

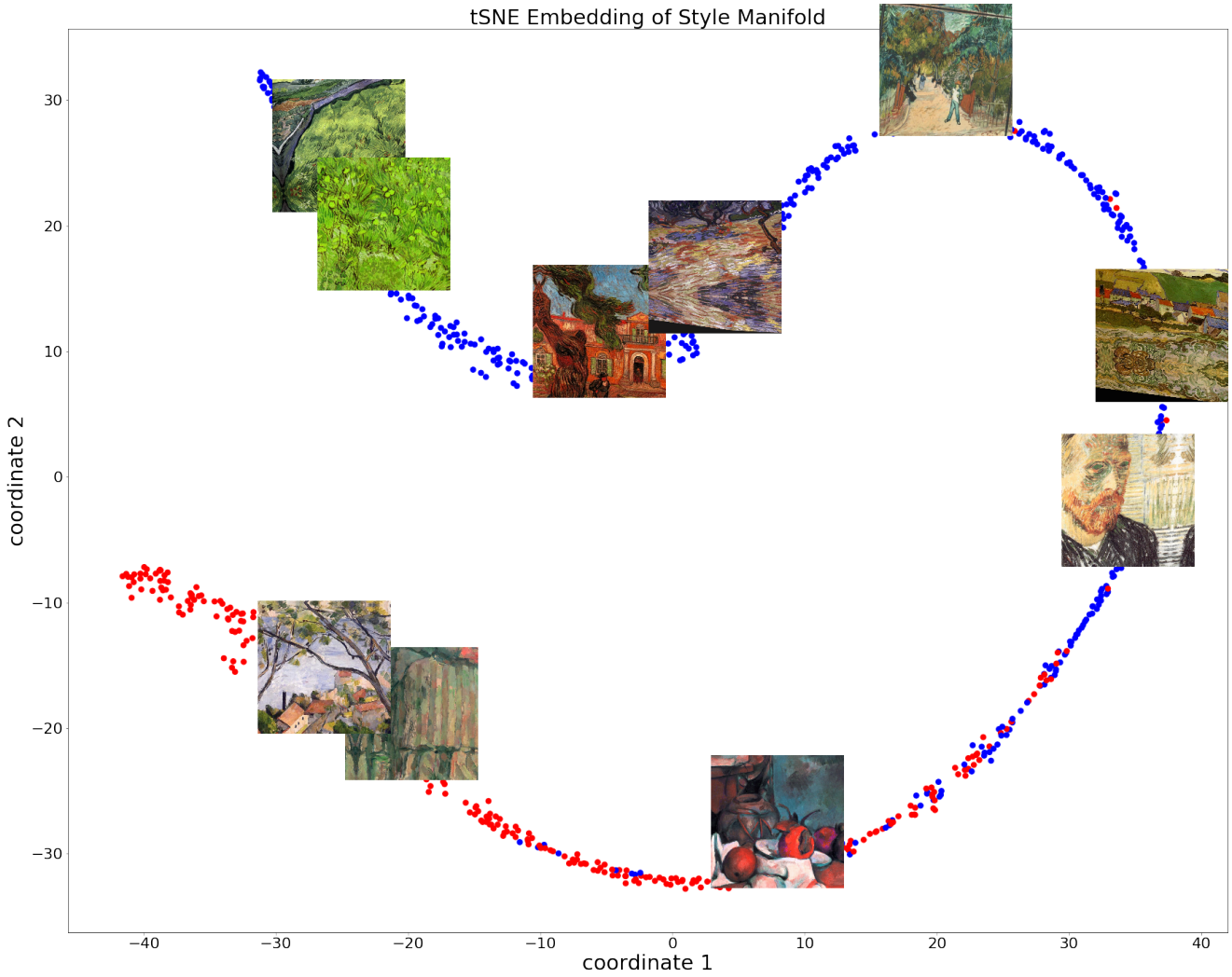

Figure S4. This figure depicts the t-SNE[5] embedding of the style manifold obtained by applying our encoder E_s to the paintings of Cezanne (red) and van Gogh (blue). We see that the style encoder splits the style space into a 1-dimensional manifold. The style space is split in half for two distinct artists with some overlap in the middle. Images lying nearby in the style space share similar colors and brushstroke dynamics.

Figure S5. Image stylized in the style of Paul Cezanne.

Figure S6. Image stylized in the style of Paul Cezanne.

Figure S7. Image stylized in the style of Paul Cezanne.

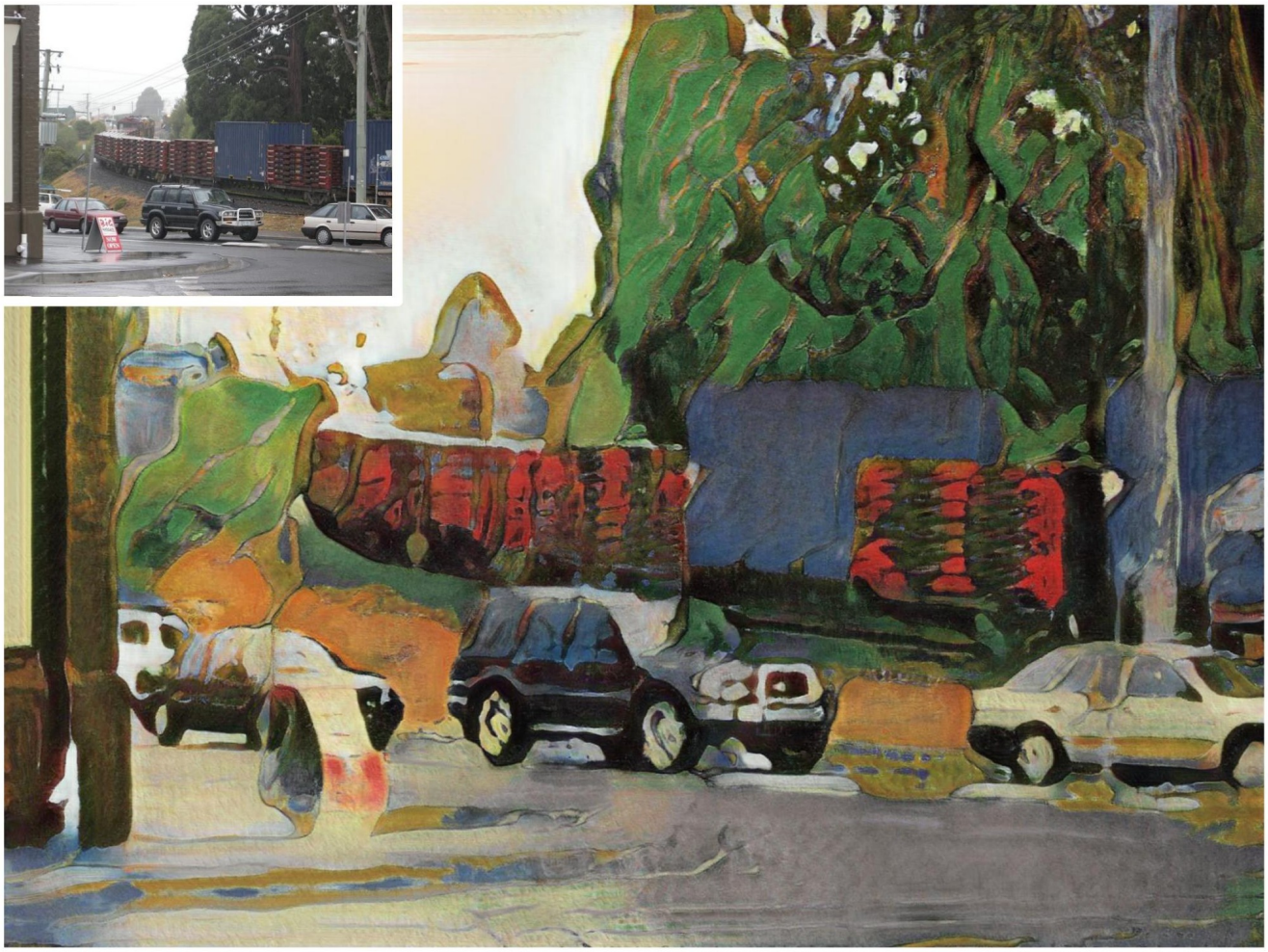

Figure S8. Image stylized in the style of Paul Gauguin.

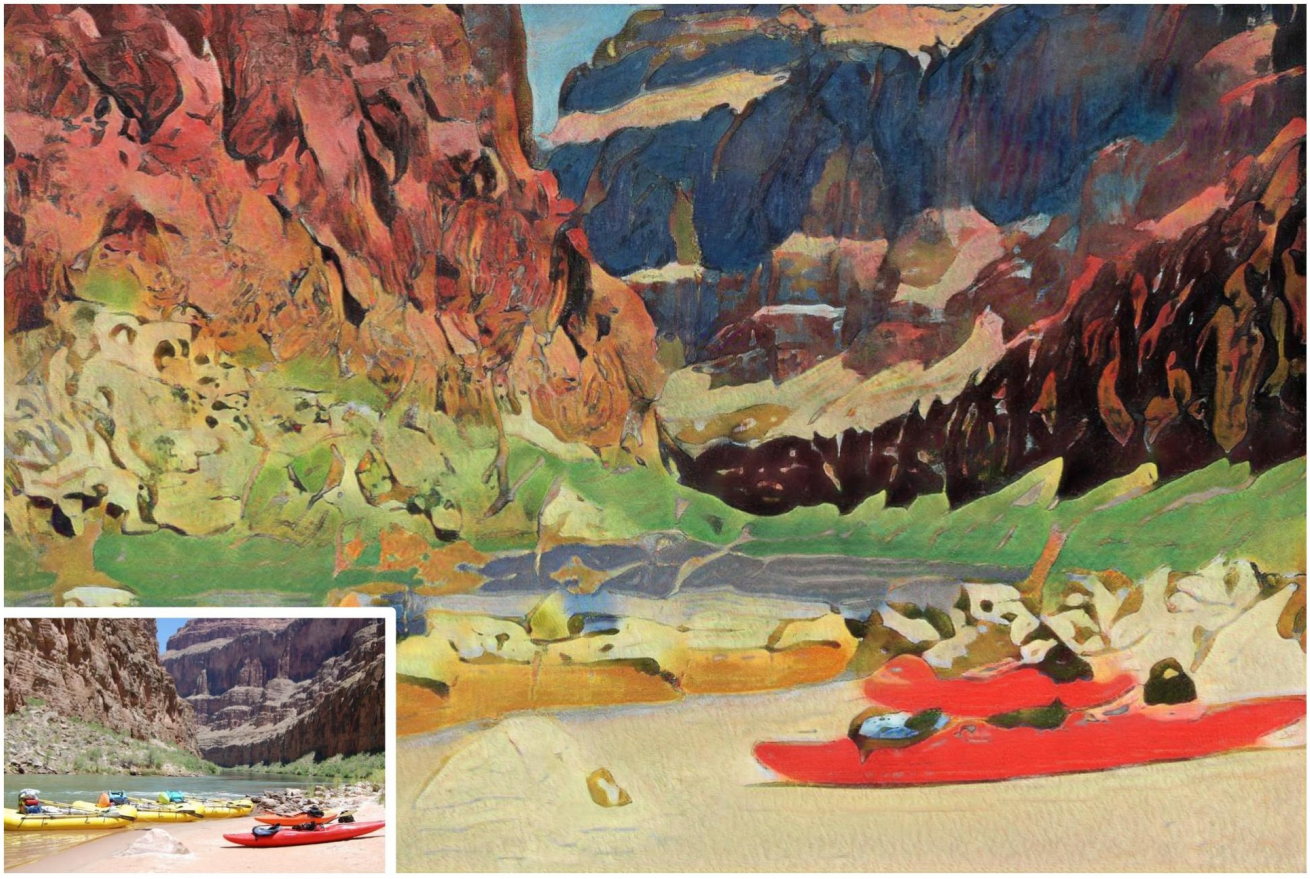

Figure S9. Image stylized in the style of Paul Gauguin.

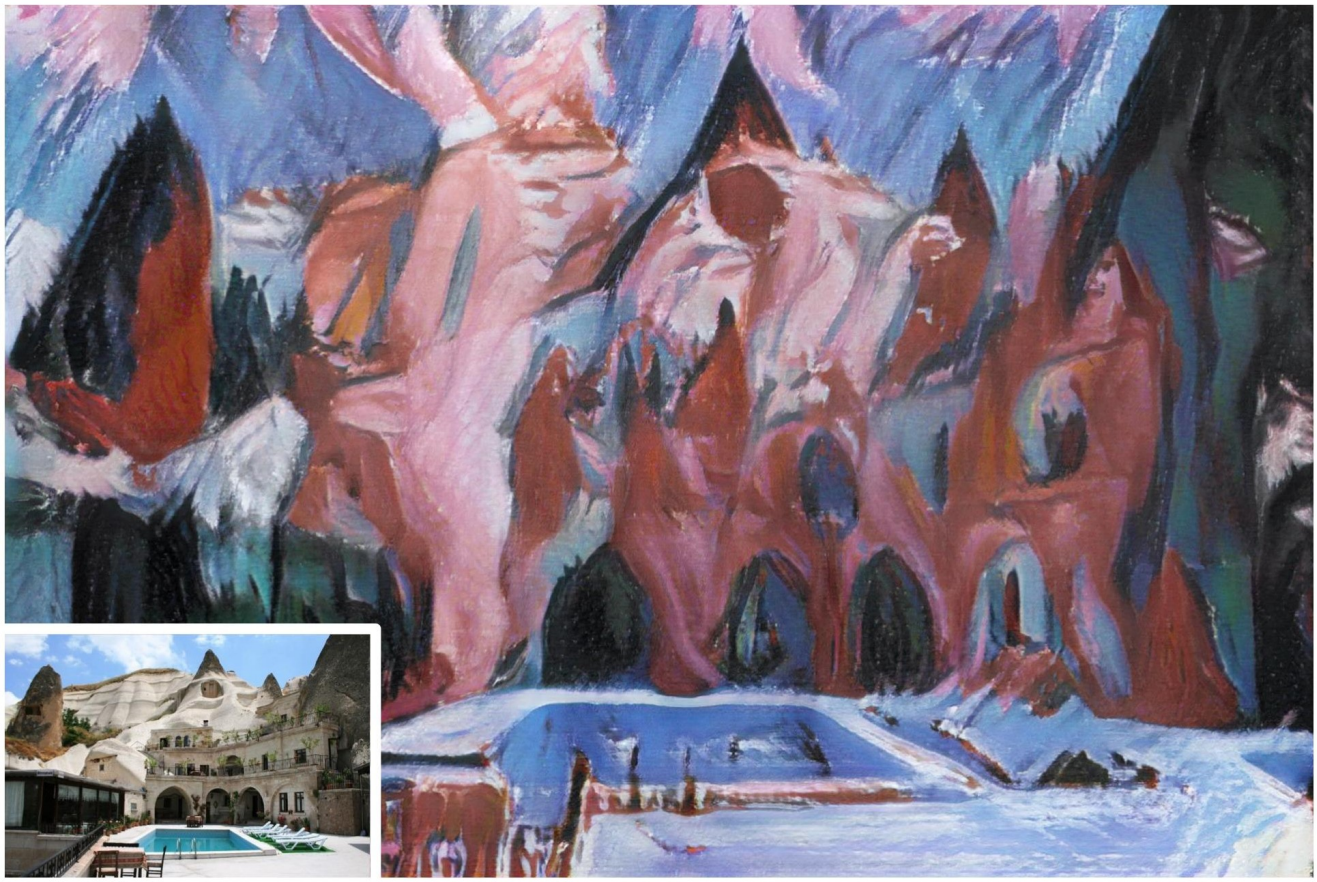

Figure S10. Image stylized in the style of Ernst Ludwig Kirchner.

Figure S11. Image stylized in style of Ernst Ludwig Kirchner.

Figure S12. Image stylized in the style of Claude Monet.

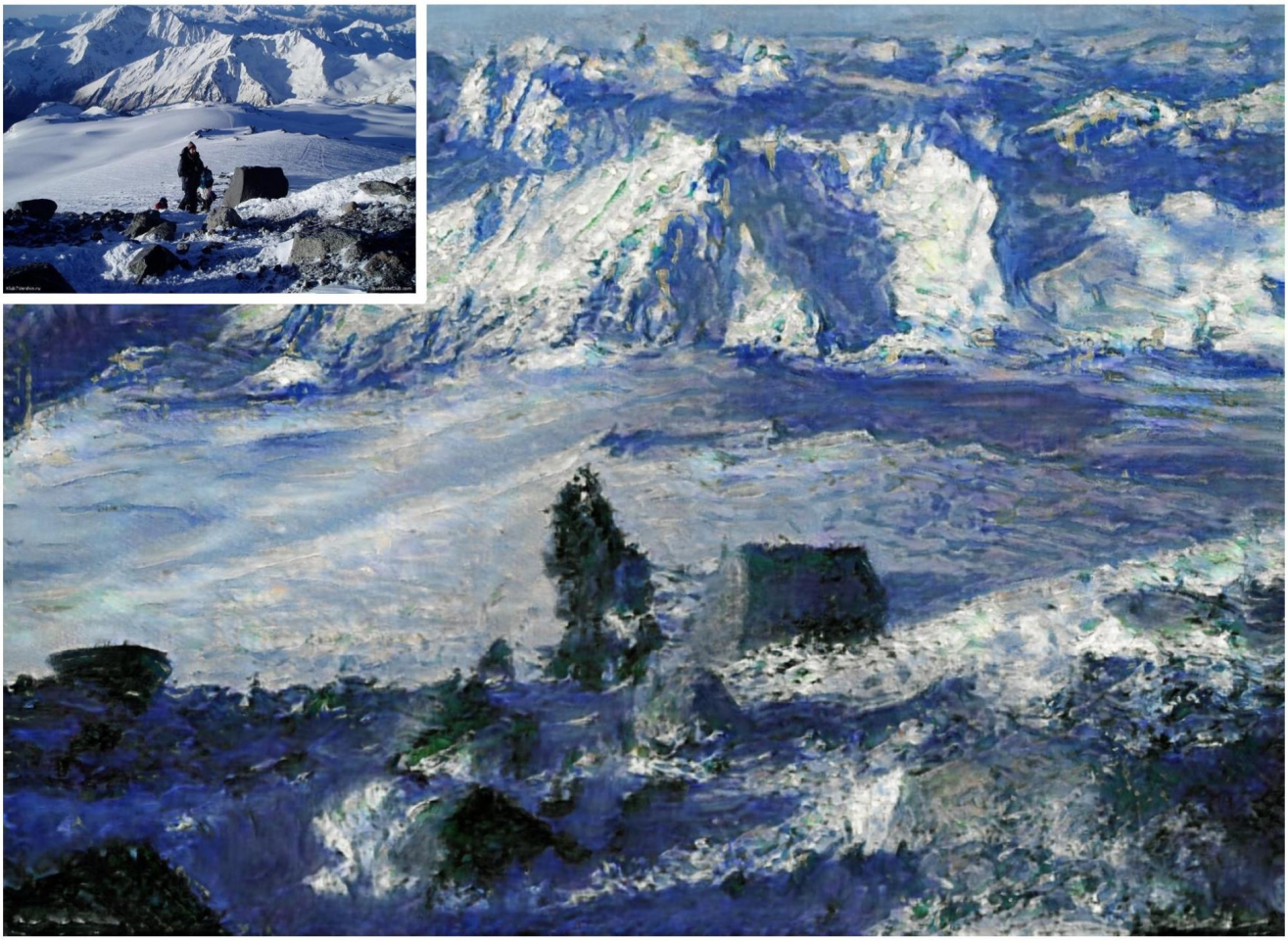

Figure S13. Image stylized in the style of Claude Monet.

Figure S14. Image stylized in the style of Berthe Morisot.

Figure S15. Image stylized in the style of Pablo Picasso.

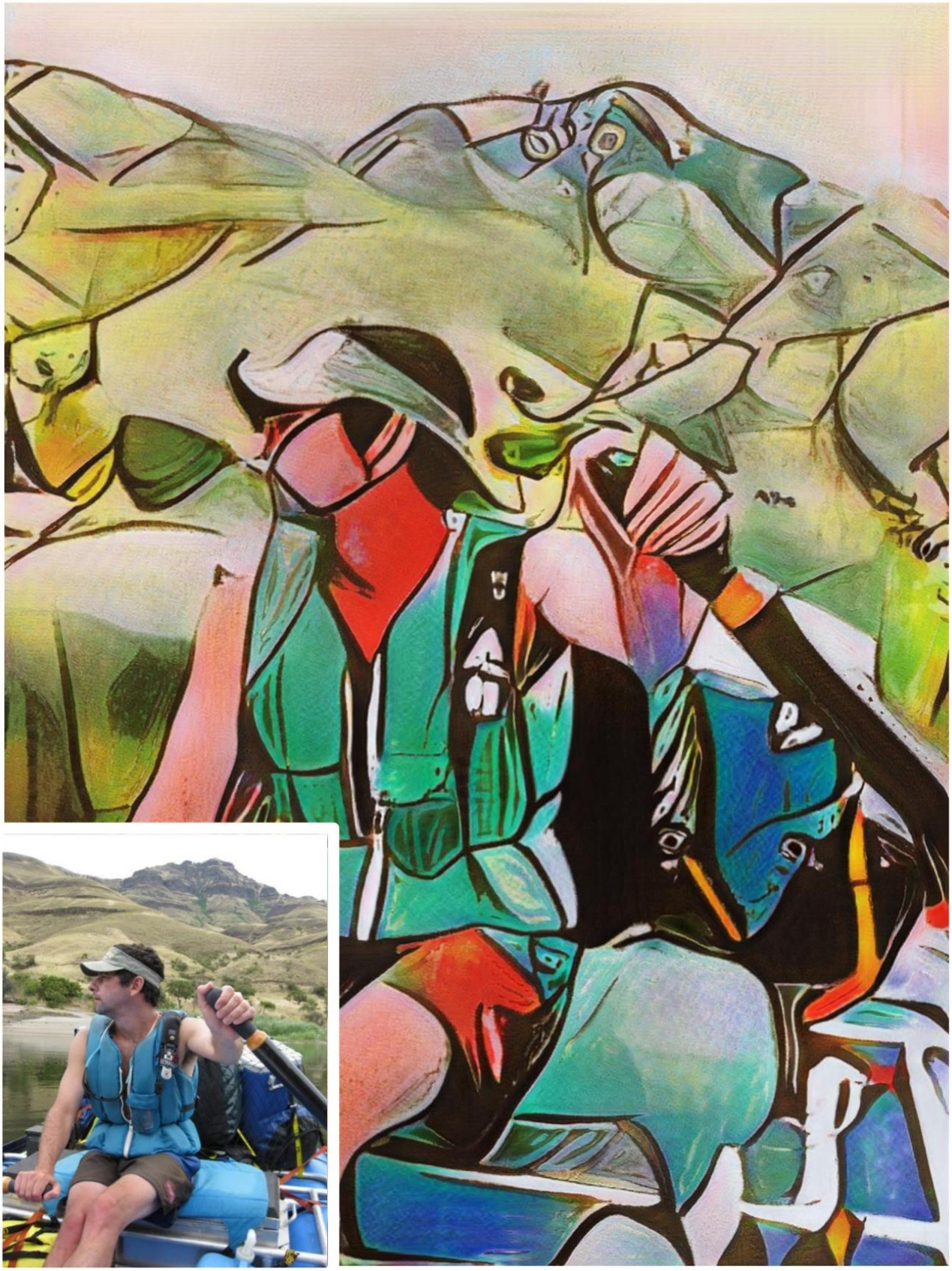

Figure S16. Image stylized in the style of Pablo Picasso.

Figure S17. Image stylized in the style of Vincent van Gogh.

Figure S18. Image stylized in the style of Vincent van Gogh.

References

[1] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE, 2016. 1, 2

[2] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013. 1

[3] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016. 1

[4] Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn 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

[5] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. 2008. 2, 5

[6] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 1

[7] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, 2017. 1, 2

Solution to Figure S1:
First: CycleGAN, Ours, real, Ours, real, AST, Ours, real, Gatys et al.
Second: AST, Ours, real, Ours, CycleGAN, real, Gatys et al., Ours, real.
Third: Gatys et al., Ours, Ours, real, Ours, Ours, real, CycleGAN, AST.
Fourth: AST, Ours, real, Gatys et al., Ours, real, CycleGAN, Ours, real.
Fifth: Ours, Gatys et al., real, AST, Ours, real, Ours, real, CycleGAN.