CLIPstyler: Image Style Transfer with a Single Text Condition Supplementary Materials

Gihyun Kwon[1] Jong Chul Ye[1,2]
Dept. of Bio and Brain Engineering[1], Kim Jaechul Graduate School of AI[2], KAIST
{cyclomon,jong.ye}@kaist.ac.kr

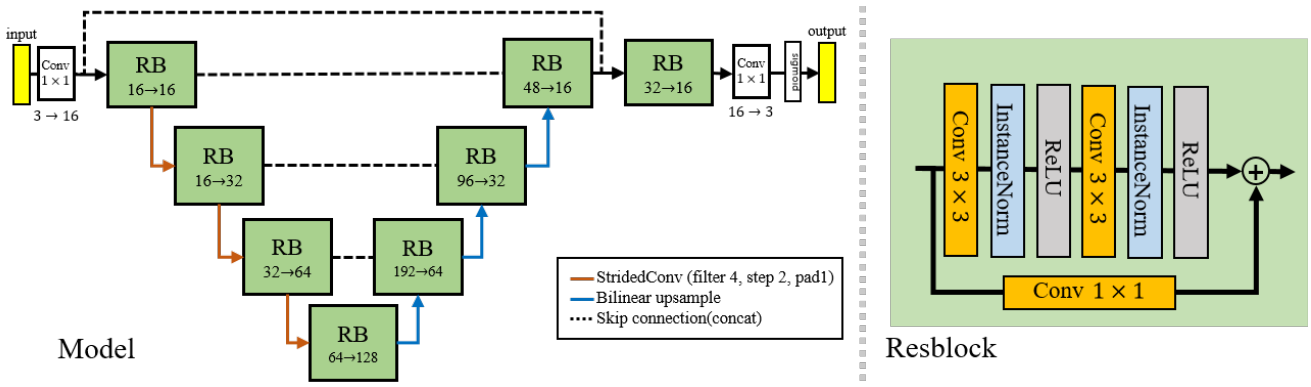

Figure 1. Our style network model architecture.

Supplementary Material

A. More details of implementation

Network architecture: For our style network f, we use a lightweight U-net which is described in the Fig. 1. We discovered that using residual block improves the content preservation and training stability. Since we train the network with only 200 iterations and the resolution of input image is relatively high, our network have to be lightweight. Therefore, we limited the maximum channel numbers to 128, and the highest resolution layer has only 16 channels. To better preserve the content information, we included another skip connection between input-output features.

Implementation details: Since CLIP model receives the images with resolution of 224×224, we resized all of the images including patch and whole image before feeding to the CLIP model. For augmentations, we use perspective augmentation which is implemented on Pytorch torchvision library. We directly used `RandomPerspective(distortion_scale=0.5)`. The random perspective function is implemented in `torchvision.transforms`.

Details for fast style transfer: For training our fast style transfer model, we used the cropped patches from DIV2K [1], and the crop size is 224. We used batch size of 4 and Adam optimizer with learning rate of 1×10^{-4}. Similar to our basic method, we used learning rate decay strategy. For augmentation, we applied random perspective augmentation for 16 times, therefore our total training patch number per iteration is $N = 64$. We also calculated directional CLIP loss L_{dir} with using patches before applying augmentations. Therefore, the loss functions of our fast style transfer is defined as:

$$L_{total} = \lambda_d L_{dir} + \lambda_p L_{patch} + \lambda_c L_c + \lambda_{tv} L_{tv},$$

which is same as the loss of our basic method. For hyperparameters, we set λ_d, λ_p, λ_c, and λ_{tv} as 1, 10, 1, and 1×10^{-4}, respectively. For threshold rejection, we set τ as 0.7.

For detailed implementation, please refer to our source code[1].

B. Additional comparison results

B.1. Comparison with other baselines

As a further comparison with other baselines, we show additional comparison results with two simple approaches:

[1] https://github.com/paper11667/CLIPstyler

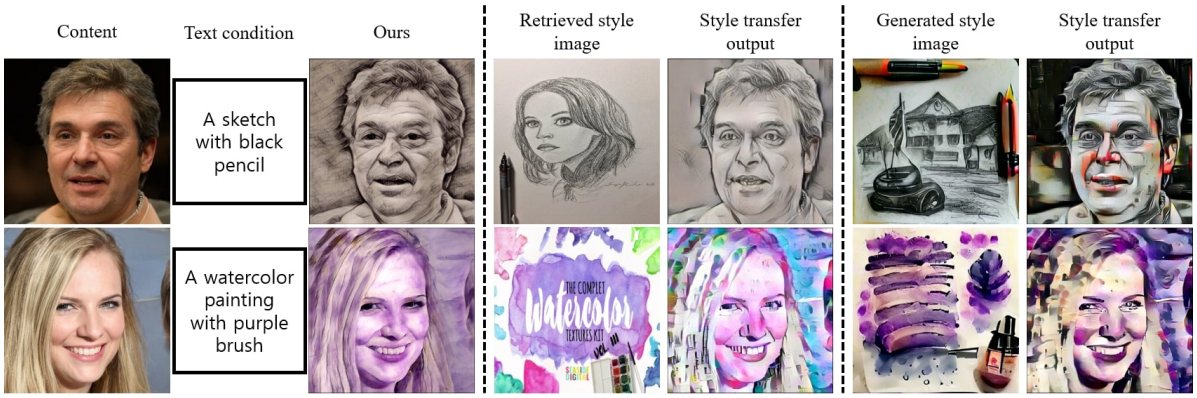

Figure 2. (a) Our results. (b) Retrieved style image and style transfer output. (c) Generated style image and style transfer output.

Methods	User study	
	Style ↑	Content ↑
Retrieve + Sty 1)	3.01	3.38
Text2Image + Sty 2)	2.80	3.61
StyleCLIP	1.47	**4.35**
StyleGAN-NADA	2.66	3.65
AdaIN+CLIP	**3.32**	2.97
VQGAN+CLIP	2.75	2.31
Ours	**3.78**	**4.18**

Table 1. User study results on various text-guided image manipulation models. The values marked with blue color refer to the best scores, and the values with red color are the second best scores.

1) to apply the existing style transfer method by using the text-retrieved image as a style image, and 2) using the image generated by the text-to-image generation model as a style image. For 1), we used the pre-trained text-to-image retrieval model[2] and chose the image with the highest CLIP score as a style image. For 2), we used the image generated through VQGAN+CLIP[3] as a style image. For image style transfer method, we chose widely used transfer model SANet [3] in both of 1) and 2).

In Fig. 2, we can observe that the outputs of baseline 1) and 2) both show reasonable results in the upper row. However, if the retrieved or generated style image is not adequate for style image (bottom row), we can see that the outputs are degraded. Although the frameworks are efficient in inference time, the results show that 1) and 2) methods have major disadvantages in that they are highly dependent on the performance of retrieval and generation models.

B.2. User study

Experiment Details: For quantitative comparison, we conducted a user study. For baseline models, we selected six different text-guided manipulation methods: StyleGAN-NADA [2], StyleCLIP [4], AdaIN+CLIP, VQGAN+CLIP, and the previous approaches 1), 2). Since the conventional style transfer methods use style images, we did not carry out experiments using these methods, and just selected the CLIP-based methods for fair comparison.

For user study, we generated 160 different stylized images with 10 different text conditions for each model (total 1,120 images). We used both of artistic style (e.g. "A sketch with black pencil") and non-artistic style (e.g. "Leather") text conditions. Since StyleGAN-NADA and StyleCLIP provided the models pre-trained on human face dataset, we used randomly sampled human face images as content images for fair comparison.

With generated images, we created 20 different questions. Specifically, to quantify the user preference, we asked participants questions about content preservation and the expression of textures that match the text conditions. In order to collect the detailed opinions of users, we used a custom-made opinion scoring system using Google Form. More specifically, we provided a total of 25 users with stylized images and asked them to rate the level of content preservation and text customization. We set minimum score as 1, and the maximum scores is 5. For each question, users can choose the scores among 5 different options: 1-very low , 2-low, 3-middle, 4-high, 5-very high. The 25 different subjects come from the age group between 20s and 40s, who are randomly recruited online. Then we reported the average values.

Results: Table 1 shows the user study results on various text-guided models. Our model outperformed the baseline models in content and style scores. More specifically, StyleGAN-NADA showed decent score in content preservation, but the stylization score was relatively lower than other baselines. StyleCLIP showed the best score in content preservation, but it showed the worst score in stylization. StyleGAN-NADA and StyleCLIP showed bad results in style change as they are strongly confined to the pre-trained dataset using a pre-trained StyleGAN. For baselines of 1) and 2), the method showed decent scores in content preservation, but the methods failed to successfully transfer the target style with showing relatively low values in style

[2]https://github.com/rom1504/clip-retrieval
[3]https://github.com/nerdyrodent/VQGAN-CLIP

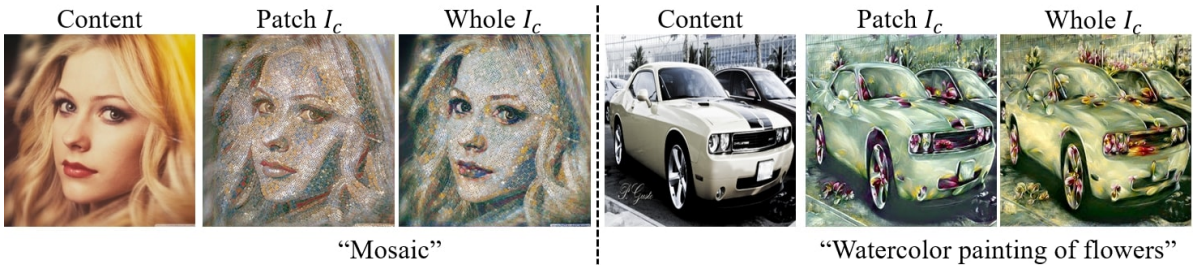

Figure 3. Comparison result with patch cropping on content image I_c. The results show there is almost no difference in perceptual quality.

Methods	CLIP score ↑
Retrieve + Sty 1)	0.2317
Text2Image + Sty 2)	0.2213
StyleCLIP	0.1982
StyleGAN-NADA	0.2252
AdaIN+CLIP	**0.2487**
VQGAN+CLIP	0.2249
Ours	**0.2515**

Table 2. Quantitative comparison results on patch-wise CLIP scores. Blue-second best, Red-best

Settings	Preference Score↑
no Augment	2.61
no Thresh	**3.02**
no L_{patch}	1.69
no L_{dir}	2.58
L_{total}	**3.92**

Table 3. User study for ablation study. Blue-second best, Red-best

scores.

In AdaIN+CLIP, the model shows much better style transfer performance with scoring the second best among all the models, but it has disadvantages in content preservation. For VQGAN+CLIP, the model shows degraded performance with the worst content preservation score, which means that the model hardly reflected the shape of the input contents.

For our model, we obtained the second best score in content preservation with little difference to StyleCLIP, and obtained the best score in stylization. The user study results show that our model showed best performance considering both of content preservation and stylization.

B.3. Patch-wise CLIP score

In order to strengthen the evaluation, we measured additional quantitative metrics. To measure the correspondence between text condition and texture, we calculated cosine similarity between the CLIP embeddings of output patches and target texts. With single output image of 512×512 resolution size, we randomly cropped 64 patches which have various resolution sizes (64×64 ∼ 224×224). As a validation set, we used the same images in the user study questions.

In Tab. 2, we show the averaged CLIP scores on various models. Again, our model outperformed baseline models with showing highest CLIP scores. More specifically, StyleCLIP showed the worst performance with the lowest CLIP score, and AdaIN+CLIP scored second best among all of the baseline models. This shows that the CLIP score results have almost same tendencies as the stylization score of user study.

C. Additional ablation studies

C.1. Patch crop on content images

In order to figure out whether patch cropping on content image I_c affects the output quality, we compared the outputs using proposed CLIP loss with cropped patches from both of content and output images. In Fig. 3, we see little difference in perceptual quality, therefore we did not crop the patches from content images due to computation time. Note that to obtain the loss with patches from the content image, more images should be embedded into the CLIP space, and it increases the run time (∼10 seconds in our case) for each style transfer.

C.2. User study

For more thorough ablations study, we conducted another user study for ablation study in Tab. 3. For baselines, we use four different settings: training without L_{dir}, without proposed L_{patch}, without proposed threshold rejection, and without augmentations. For each setting, we generated 80 different stylized images with 10 text conditions (total 400 images).

With generated images, we created 10 different questions. In order to collect the detailed opinions of users, we used a custom-made opinion scoring system using Google Form. More specifically, we provided a total of 20 users with stylized images and asked them to score the images

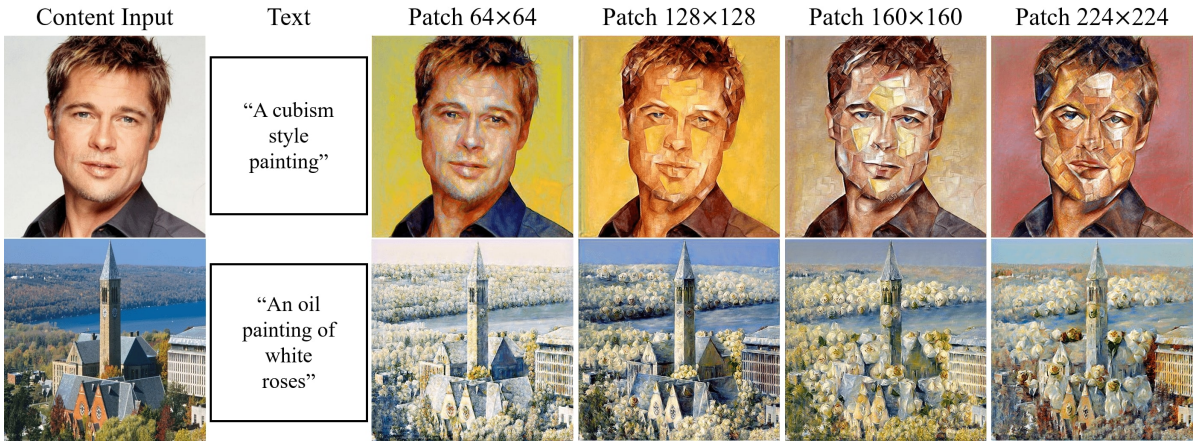

Figure 4. CLIPstyler outputs with different patch crop sizes. With a smaller patch size, we can obtain finer stylization results. With a large patch size, the outputs have large style patterns.

with overall preference. Users can choose the scores among 5 options (1-very bad, 2-bad, 3-neutral, 4-good, 5-very good). The results show that the users prefer our best setting. The 20 different subjects come from the age group between 20s and 40s, who are randomly recruited online.

The results show that our best setting outperforms other baselines with obtaining higher preference scores. We can see that without using our proposed patch-wise CLIP loss (no L_{patch}), the images showed the worst score among all of the baselines, which means that L_{patch} is the most important loss in stylization.

D. Effect of patch size

For calculating our proposed L_{patch}, we need to decide the proper patch crop size. Although we choose patch size of 128 as our default setting, we can obtain various effects with changing the patch size. The results in Fig. 4 shows the effect of different patch sizes. If we use larger patch size in training, we can have larger ‘brushstroke’ which can stylize the content image in a rough scale. With small patch sizes, we can apply much finer style patterns to the content images.

E. Failure cases

In this section, we show several failure cases of our model. First, as our model is based on random patch sampling and augmentations, failure cases often occur when bad patches are sampled. In the left panel of Fig. 5, we can see that there are artifacts which are caused by direct visualization of the text condition itself. For example, there are direct visualization of the text “Pop art” on the first image. Although we found out that using slightly larger value of total variation loss can partially alleviate the artifact, it is not a perfect solution. In our future work, we plan to improve the sampling algorithm so that the sampled patches are perceptually meaningful in training the network.

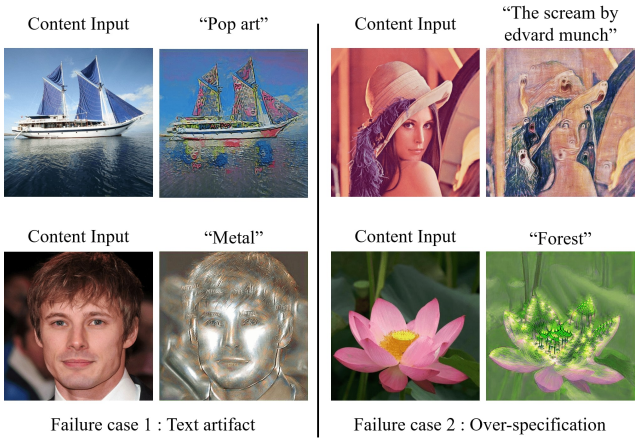

Figure 5. Failure cases of our CLIPstyler.

In the second case, output images with too specific text conditions may have unwanted patterns which are perceptually bad. In the right panel of Fig. 5, the model overly focused on reconstructing the text-conditioned target ‘object’, instead of applying the ‘texture’ of text conditions. Since this artifact is mainly caused by the embedding aspect of pre-trained CLIP model, we need to detour the effect with using different text conditions of similar meanings, such as “A painting of forest”, instead of “Forest”.

F. Additional results

More results of CLIPstyler: In Fig. 6, we show our text-guided style transfer outputs on various content images. The results clearly demonstrate that our method can apply realistic textures to content images. We also show addi-

tional results on our fast style transfer method in Fig. 7.

More comparison results: For further qualitative comparison, we provide style transfer outputs from various baseline models which use CLIP to manipulate the images. We compare the results from our model, Retrieve + Sty 1), Text2Image + Sty 2), StyleGAN-NADA, StyleCLIP, VQGAN+CLIP, and AdaIN+CLIP. Since StyleGAN-NADA and StyleCLIP are trained on human face dataset, we carried out experiments on various human face images. Part of provided results are also used in our user study questions. In Fig. 8 and 9, the results from our model show the best style transfer quality in both of content preservation and text-guided texture synthesis.

More high-resolution results: We also provide additional high-resolution output images from our fast style transfer method. The results in Figs. 10,11,12,13 shows the results. Our model can synthesize the realistic textures for high-resolution images from text conditions.

G. Limitations and Negative social impact

Although our method shows high-quality results with single text condition, there are several remaining technical issues. First, our transfer method requires network training with each given text condition, so real-time style transfer is still not possible. While our fast style transfer method can partially address this issue, we observe that the resulting quality of the style transfer is still inferior to our default single image based optimization. Second, as shown in our failure cases, low quality results can be produced when bad patches are sampled. As a future work, we will try to solve the problems explained.

Since our method manipulate the content images with various text conditions, when the user manipulate the content images with malicious words such as obscene expressions, it may bring negative social effects. In addition, when such malicious style transfer is applied to personal photos containing sensitive information, the impact may be greater.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 1

[2] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021. 2

[3] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5880–5888, 2019. 2

[4] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 2

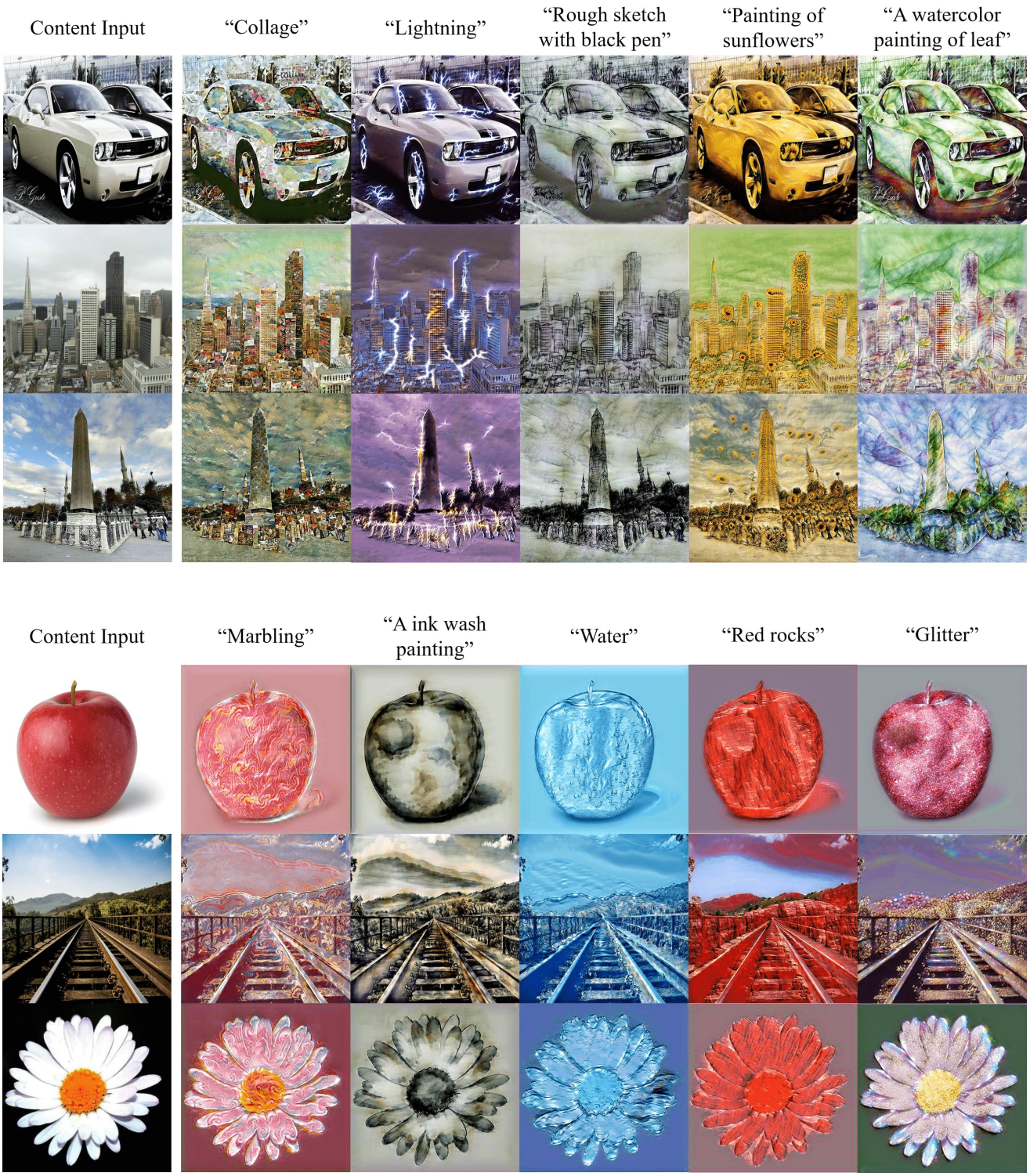

Figure 6. Additional results of our CLIPstyler.

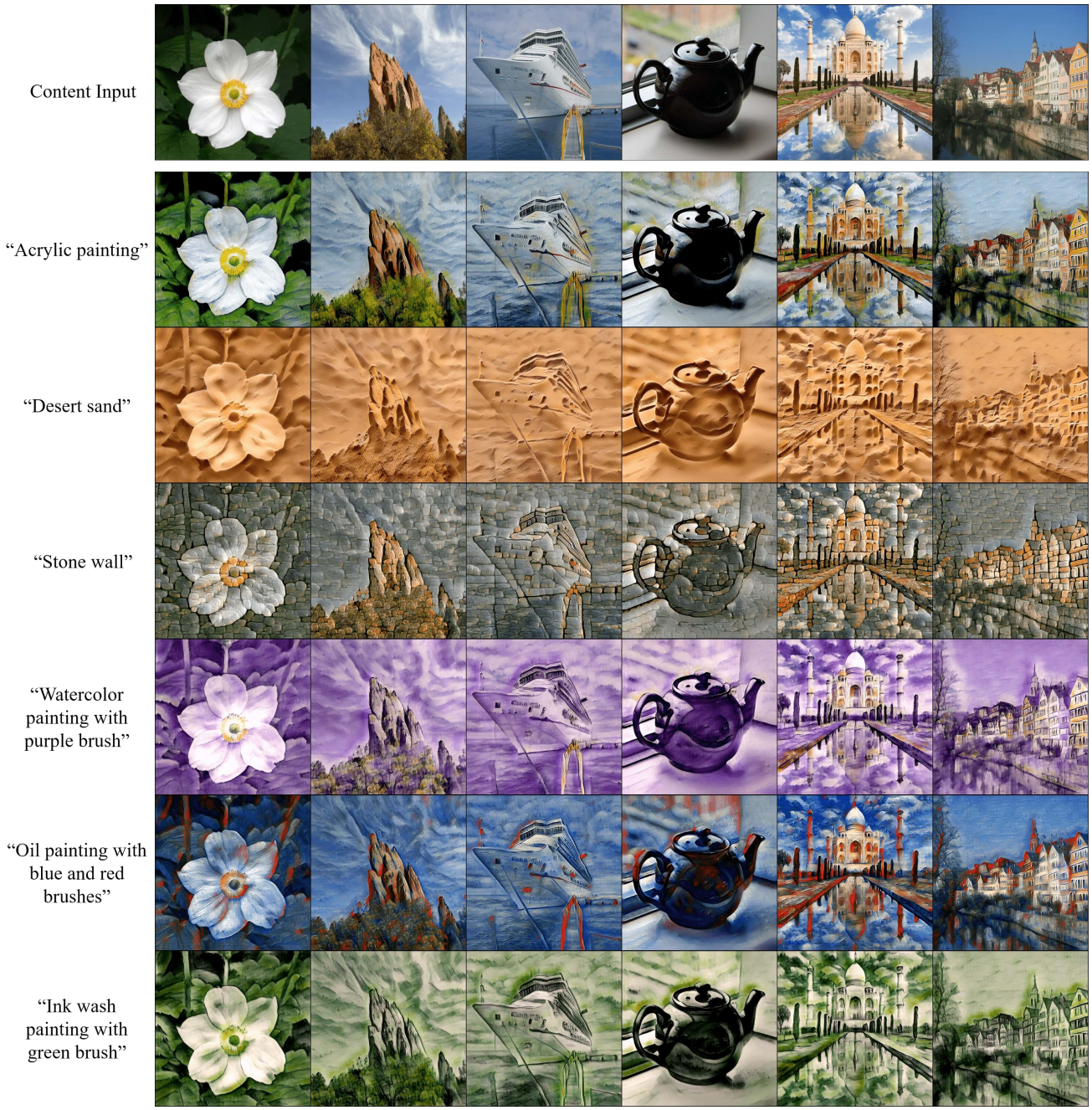

Figure 7. Additional results of our fast style transfer method.

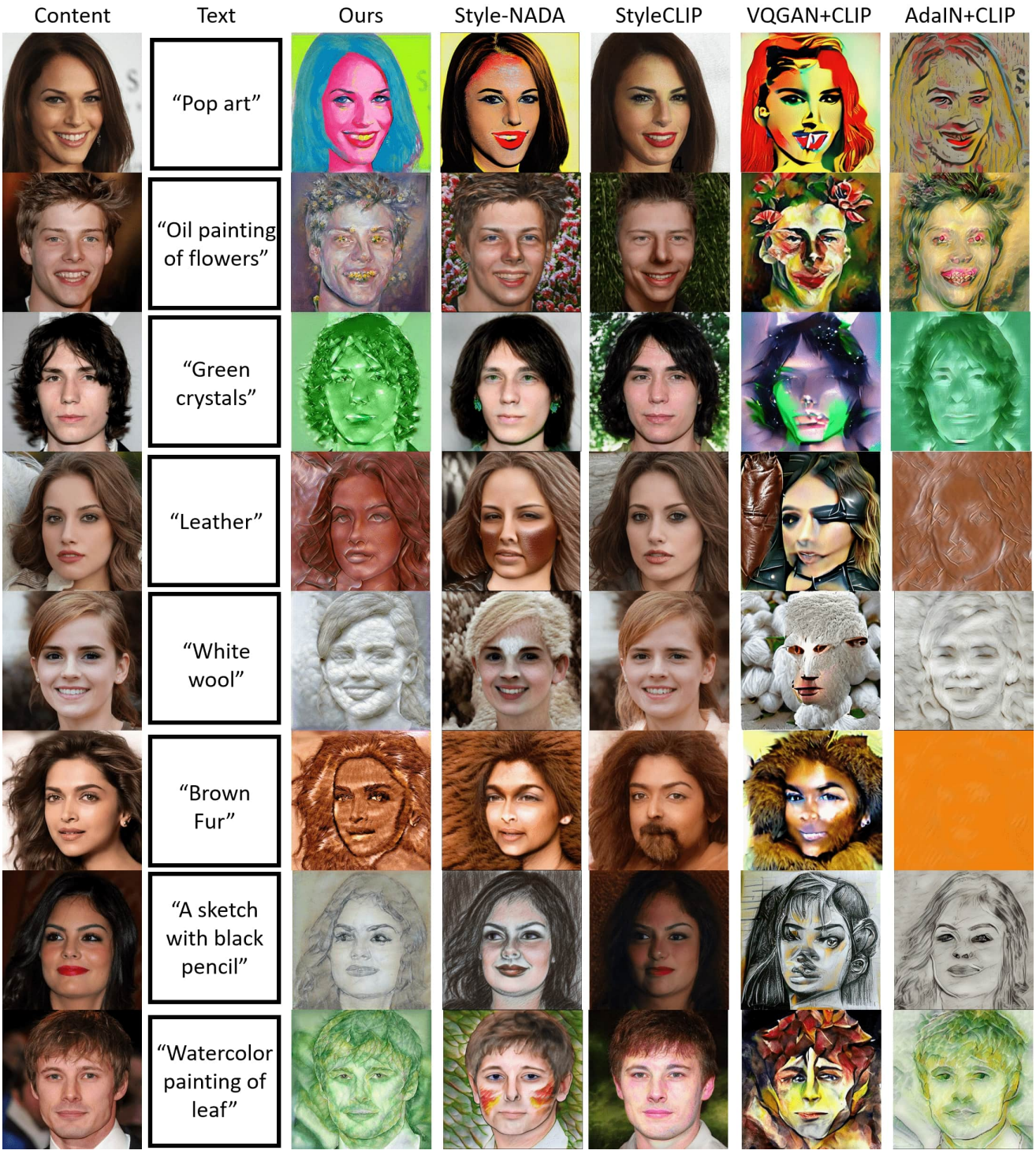

Figure 8. Additional comparison with baseline methods.

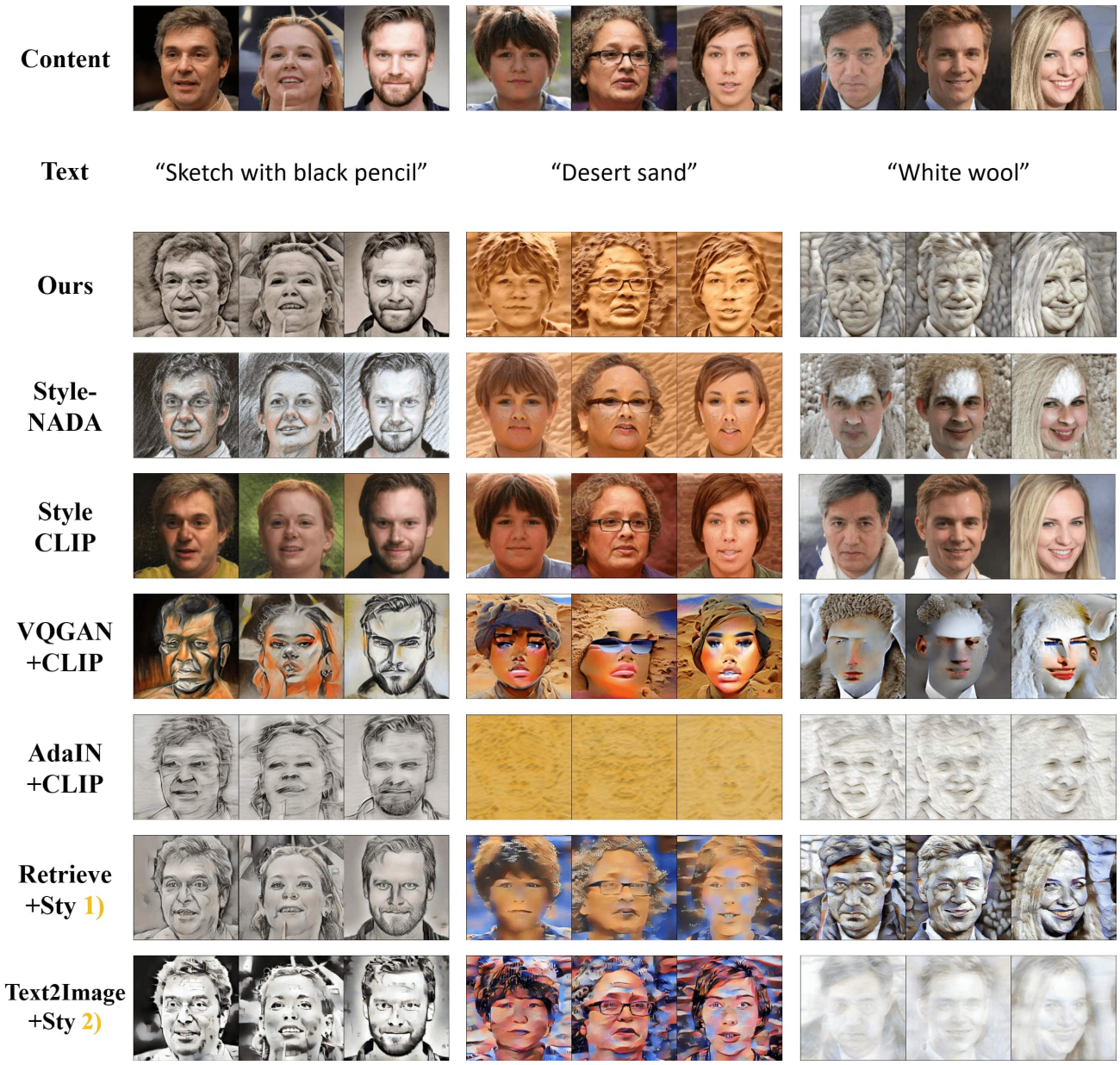

Figure 9. Additional comparison with baseline methods.

Figure 10. High resolution results from our fast style transfer method. The resolution size is 1920×2580. Up: Content image. Down: Output with text condition of "Watercolor painting".

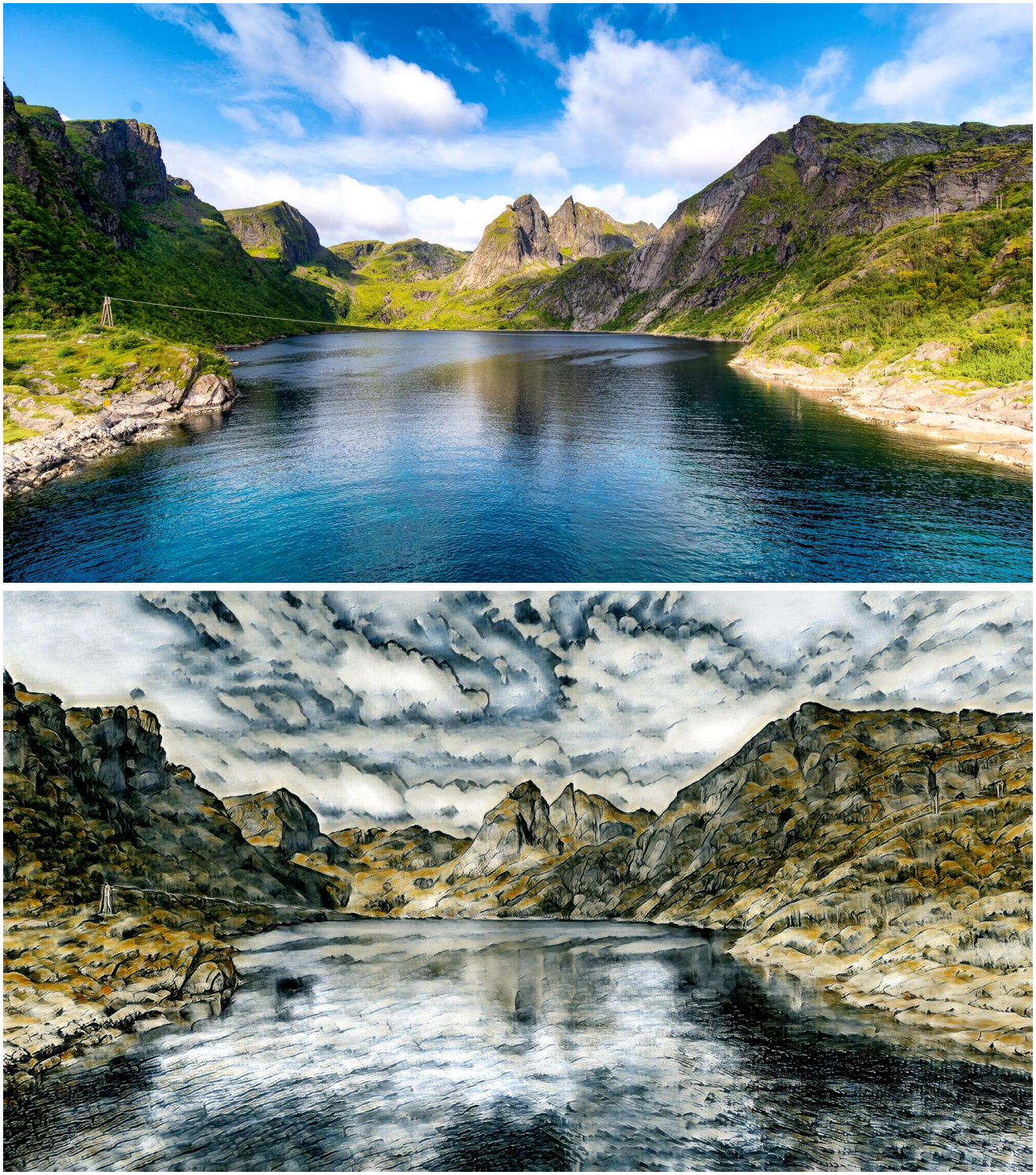

Figure 11. High resolution results from our fast style transfer method. The resolution size is 1128×2000. Up: Content image. Down: Output with text condition of “An ink wash painting”.

Figure 12. High resolution results from our fast style transfer method. The resolution size is 1600×2400. Up: Content image. Down: Output with text condition of "A fauvism style painting with bright color".

Figure 13. High resolution results from our fast style transfer method. The resolution size is 1656×2200. Up: Content image. Down: Output with text condition of "Snowy".