Deep Preset: Blending and Retouching Photos with Color Style Transfer (Supplementary Materials)

Man M. Ho
Hosei University
Tokyo, Japan
man.hominh.6m@stu.hosei.ac.jp

Jinjia Zhou
Hosei University
Tokyo, Japan
jinjia.zhou.35@hosei.ac.jp

1. Touched colors in test data

Although natural-to-retouched transformation is close to the real-world problem and should be evaluated only, in our test set, we have also included our out-of-distribution cases (inputs or references are touched before testing) for a fair comparison. We clarify which test subset has inputs and references (before applying presets for testing) with natural/touched colors in Table 1.

2. Network's technical details

The encoder T contains five Down-sampling Layers (DL) with Max Pooling in prior (excluding the first one), same as the encoder C. Meanwhile, the decoder G includes 5 Up-sampling Layers (UL) with Bi-linear Pooling (excluding the fifth layer), and the final convolutional module with *Tanh* activation function to synthesize the stylized output. To avoid aliasing, we adopt the works [11, 3] and apply a blur filter with a size of $[1, 2, 1]$ to all pooling modules. All convolutional modules in DL and UL have a Sample-based Evolving Normalization-Activation [6] followed. The linear L consists of 3 fully-connected layers with the Leaky ReLU activation function. The activation of its last layer is replaced by *Tanh* function to estimate the applied preset, as shown in Figure 3 of our main paper, and Figure 1 of this supplemental document. The first convolution module in each encoder uses a kernel size of 5×5 to observe features on a large receptive field; meanwhile, remaining convolution modules use a kernel size of 3×3.

3. Training details

We train our models using Adam optimizer [4] with the learning rate of 0.0001, momentum $beta_1 = 0.9$, $beta_2 = 0.999$, the batch size of 8. To make diversity training samples, we apply random crop with the size of 352×352, random rotation with degrees $90, 180, 270$, and finally random flip in both horizontal and vertical ways. All photos are normalized to the range $[-1, 1]$.

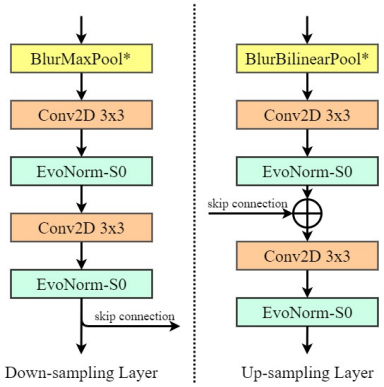

Figure 1. Illustration for our Down- and Up-sampling Layers, where Conv2D is a convolutional module, EvoNorm-S0 denotes Sample-based Evolving Normalization-Activation [6], $\oplus$ represents a summation, $*$ denotes that the *first layer* of both the encoders and the *fifth layer* of the decoder G do not include the blur pooling [11].

4. Out-of-distribution

Our work changes the base colors of the content instead of treating the content colors as a transfer destination. Plus, we only train our model on natural-to-retouched transformation. Therefore, we conduct experiments on "*stylizing a retouched content*" and "*what if we use an artistic image as reference*".

On retouched contents. Deep Preset is trained to convert a photo with natural colors to its retouched version; therefore, retouched-to-retouched transformation is an out-of-distribution case. As a result, our model implicitly learns cross-content color transformation from input to reference *in any style* and has the same behavior as a natural-to-retouched scheme, as shown in Figure 2. Our work thus outperforms others in the retouched-to-retouched domain.

On artistic styles. Artistic paintings are out-of-distribution since we only train on camera-taken photos. As a result, our work can slightly beautify a photo with a paint-

Table 1. We clarify the color status (touched or natural) of inputs and the original references before applying presets for our four test subsets DIV2K 1x100x10, DIV2K 10x10x10, FiveK 10x10x10, and Cosplay Portraits 10x10x10.

	DIV2K 1x100x10	DIV2K 10x10x10	FiveK 10x10x10	Cosplay Portraits 10x10x10
Inputs	Natural	Natural	Natural	Touched
References before applying presets	Touched	Natural	Natural	Natural

Figure 2. A result of stylizing a retouched content compared to previous works.

Figure 3. A result of stylizing a photo (natural) with paintings (by Yuumei Art).

ing as a reference. It proves that the contextual information of content is preserved while training, then blended by the color-shifting-like transformation. Therefore, the proposed Deep Preset can retain the structural details with homologous colors as reference. For example, in *left-to-right* order, our result is *brighter* with the first style, *the uniform* has *richer blue* with the second one, and the color tone slightly turns to *yellow*, especially *the girl's hair* with the third style, as shown in Figure 3.

5. Further discussion

Visual comparison between photos applied by the same/different preset. Retouching a group of photos by the same preset provides a homologous color style between photos in that group and the different color style from other groups. Please check our Figure 4 for an overall observation.

Illustration of the positive pair-wise loss function. Please check our Figure 5.

Illustration of our user study scenarios. Our user study includes three scenarios based on a two-alternative forced-choice (2AFC) scheme selecting A or B anchored by (i) ground-truth having the same content as A and B, (ii) reference having the different content from A and B, and (iii) users' favorite. Please check our Figure 6 for the illustration.

Automatic Beautification. The automatic colorization works can be treated as automatic beautification. They beautify a black-and-white by giving a plausible color based on trained data. However, they aim to synthesize the correct colors but retouched ones. Although the DeepPriors [13] provide a scheme to transfer the color from reference to the black-and-white photo, they still suffer from color mismatched, overflowed. In our case, end-users consider choosing a retouched reference having a similar context as their photo. It presents a scheme for the proposed Deep Preset to select a reference in well-retouched photos by matching perceptual information [12]. The photo is thus automatically retouched. Let's check our repository for the automatic beautification application.

Trade-off between preset prediction and positive pair-wise minimization in color transformation. As shown in Figure 7, the predicted presets from the *Linear L* without positive pair-wise (PP) loss give the promising transformation in mimicking the overall color of reference. For example, the predicted presets turns content to have *bluish tone* as *the costume* in the *third column*, *greenish tone* as *the grass* in *the last column*. However, with references retouched by the same preset, the predicted presets should provide the same color style as our hypothesis. Meanwhile, our presets with PP loss show the stability in generating a preset with various contexts, though it is worse in overall performance, as evaluated in our main manuscript. As mentioned, predicting an accurate preset is challenging. It leads to the difficulty of defining the features representing color transformation. Therefore, we reduce the expectation of preset prediction and concentrate on training the features to enhance color transformation for *Generator G* with PP loss. Our direct stylization thus outperforms the preset-based approach.

Failed Cases. Our stylization is failed when the reference falsifies color transformation (*blue* uniform to *purple* one) in *the first row*, or reference has an unstable light condition. For example, near-overexposed (*on the man's costume*) reference gives near-overexposed result in *the first row*, same as low-light reference in *the second row*, as shown in Figure 8. More fail cases can be found in Figures 10, 11.

Figure 4. Visual comparison between photos [2] retouched by three presets. *Top-left*: photos with natural colors, *others*: photos retouched by each preset.

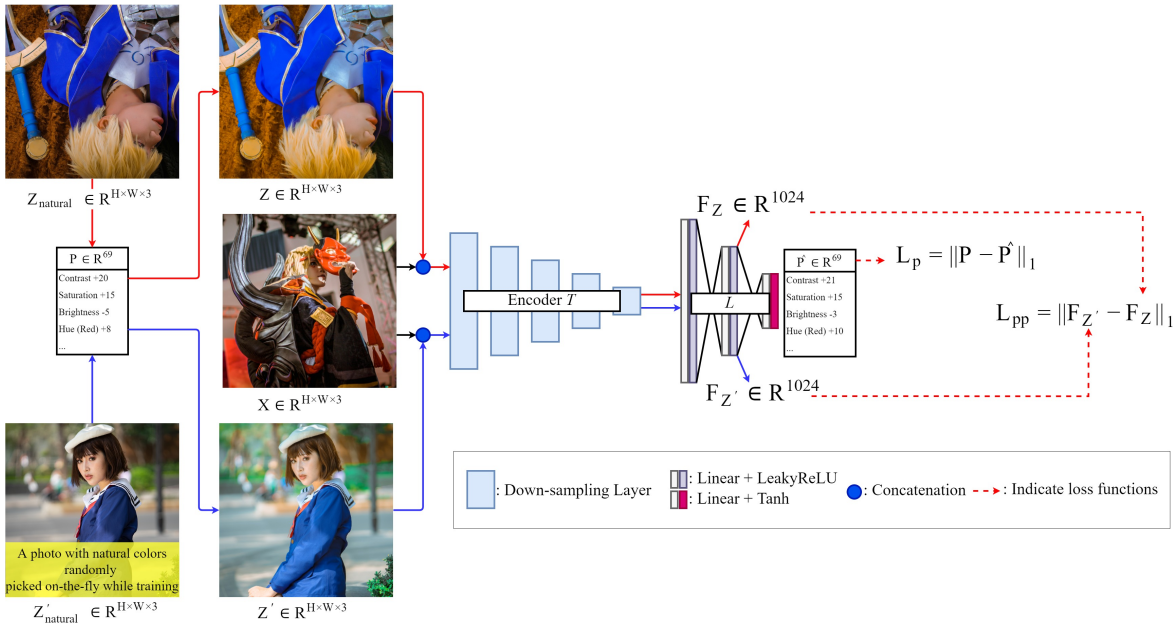

Figure 5. Illustration for our positive pair-wise loss function and preset prediction. In training, we randomly pick a natural-looking photo $Z'_{natural}$, which is retouched by the same preset P as the reference Z. The features representing color transformation by concatenation $X \bullet Z$ should be similar to $X \bullet Z'$.

6. Additional results

Additional quantitative results. We detail our result on the two subsets of DIV2K [9] 1-100 $\times$ 10 and 10-10 $\times$ 10 described in our main paper in Table 2.

Additional qualitative results. We show an example to prove that the proposed Deep Preset is capable to synthesize the similar color tone as Hue. Although the Saturation and Lightness/Values are sensitive, this work does not make the content worse. It potentially beautifies a photo with a new color style based on the reference, as shown in Figure 9. On overall, our work outperforms the previous works Reinhard's work [8], MKL [7], Deep Priors [13], FPS [5], WCT^2 [10], PhotoNAS [1] in color style transfer qualitatively (proved in the manuscript), and qualitatively as additional results shown in Figures 10, 11, 12, 13, 14, 15, 16. We also show a comparison on same/different context (*sky*)

Table 2. Quantitative comparison on DIV2K [9] in detail including 2 concepts: 1 content, 100 references, 10 presets (1×100×10) and 10 contents, 10 references, 10 presets (10×10×10).

Method		DIV2K (1×100×10)				DIV2K (10×10×10)				DIV2K (1×100×10 & 10x10x10)			
		H-Corr ↑	H-CHI ↓	PSNR ↑	LPIPS ↓	H-Corr ↑	H-CHI ↓	PSNR ↑	LPIPS ↓	H-Corr ↑	H-CHI ↓	PSNR ↑	LPIPS ↓
Reinhard et. al. [8]		0.3627	672.71	16.91	0.2459	0.2510	1158.55	14.62	0.2780	0.3069	915.63	15.77	0.2620
MKL [7]		0.3535	581.30	16.92	0.2550	0.3244	743.60	15.47	0.2664	0.3390	662.45	16.20	0.2607
Deep Priors [13]		0.5420	749.50	20.53	0.2033	0.5420	749.50	20.53	0.2033	0.5420	749.50	20.53	0.2033
FPS [5]		0.3856	1232.97	14.71	0.3025	0.3856	1232.97	14.71	0.3025	0.3856	1232.97	14.71	0.3025
WCT^2 [10]		0.3917	1269.91	16.40	0.2726	0.3917	1269.91	16.40	0.2726	0.3917	1269.91	16.40	0.2726
PhotoNAS [1]		0.4129	824.74	17.06	0.2559	0.4129	824.74	17.06	0.2559	0.4129	824.74	17.06	0.2559
Ours w/o PPL	Preset Prediction	0.6792	273.98	23.62	0.1176	0.6039	745.45	21.62	0.1102	0.6416	509.71	22.62	0.1139
	Generator	0.7188	**126.29**	23.79	0.1039	0.6678	990.82	21.96	0.1015	0.6933	558.56	22.87	0.1027
Ours w PPL	Preset Prediction	0.6573	145.19	23.12	0.1288	0.5815	**453.50**	20.94	0.1262	0.6194	**299.34**	22.03	0.1275
	Generator	**0.7269**	145.47	**24.01**	**0.0993**	**0.6743**	959.43	**22.24**	**0.0966**	**0.7006**	552.45	**23.12**	**0.0980**

Figure 6. Our three user study scenarios based on two-alternative forced-choice (2AFC) scheme.

in Figure 12.

Computational time. As shown in Table 3, our Deep Preset has a competitive inference time compared to Deep-Priors [13], FPS [5], WCT^2 [10], PhotoNAS [1].

Table 3. Computational time per 512×512 on Tesla V100.

Method	Computational time (second) per 512 × 512 image
Reinhard et. al. [8]	0.01
MKL [7]	0.05
DeepPriors [13]	1.5
FPS [5]	2.81
WCT^2 [10]	2.06
PhotoNAS [1]	0.33
Ours	0.95

Figure 7. Ablation study on the positive pair-wise (PP) loss for our Preset Prediction. We apply presets predicted by the trained models with (*bottom*)/without (*middle*) PP loss using references **retouched by the same preset** (*top*).

Figure 8. Our failed cases.

References

[1] Jie An, Haoyi Xiong, Jun Huan, and Jiebo Luo. Ultrafast photorealistic style transfer via neural architecture search. In *AAAI*, 2020.

[2] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*, pages 97–104. IEEE, 2011.

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for

Figure 9. Our results with color picker. The proposed Deep Preset can synthesize a similar Hue (color tone, pure pigment); however, Saturation and Lightness/Values are sensitive and our Saturation is higher than ground-truth. Color information template by Pinetools.

stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018.

[6] Hanxiao Liu, Andrew Brock, Karen Simonyan, and Quoc V Le. Evolving normalization-activation layers. *arXiv preprint arXiv:2004.02967*, 2020.

[7] François Pitié and Anil Kokaram. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. 2007.

[8] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[9] Radu Timofte, Shuhang Gu, Jiqing Wu, Luc Van Gool, Lei Zhang, Ming-Hsuan Yang, Muhammad Haris, et al. Ntire 2018 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[10] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9036–9045, 2019.

[11] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

[12] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[13] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017.

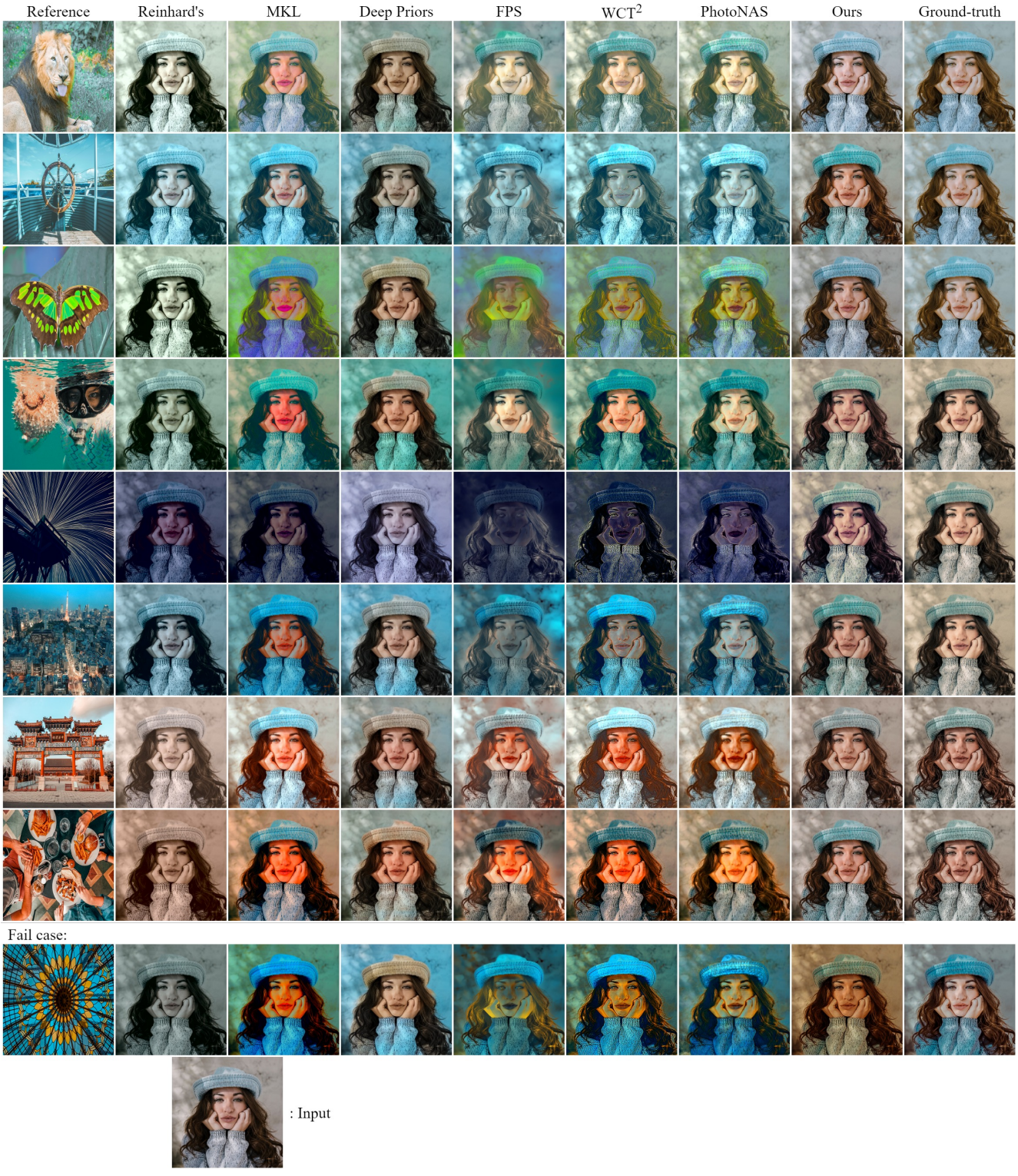

Figure 10. Additional result experimented on 1 content, 100 references, 10 presets (1000 samples) from DIV2K dataset [9]. The fail case shows our unexpectedly yellow tone.

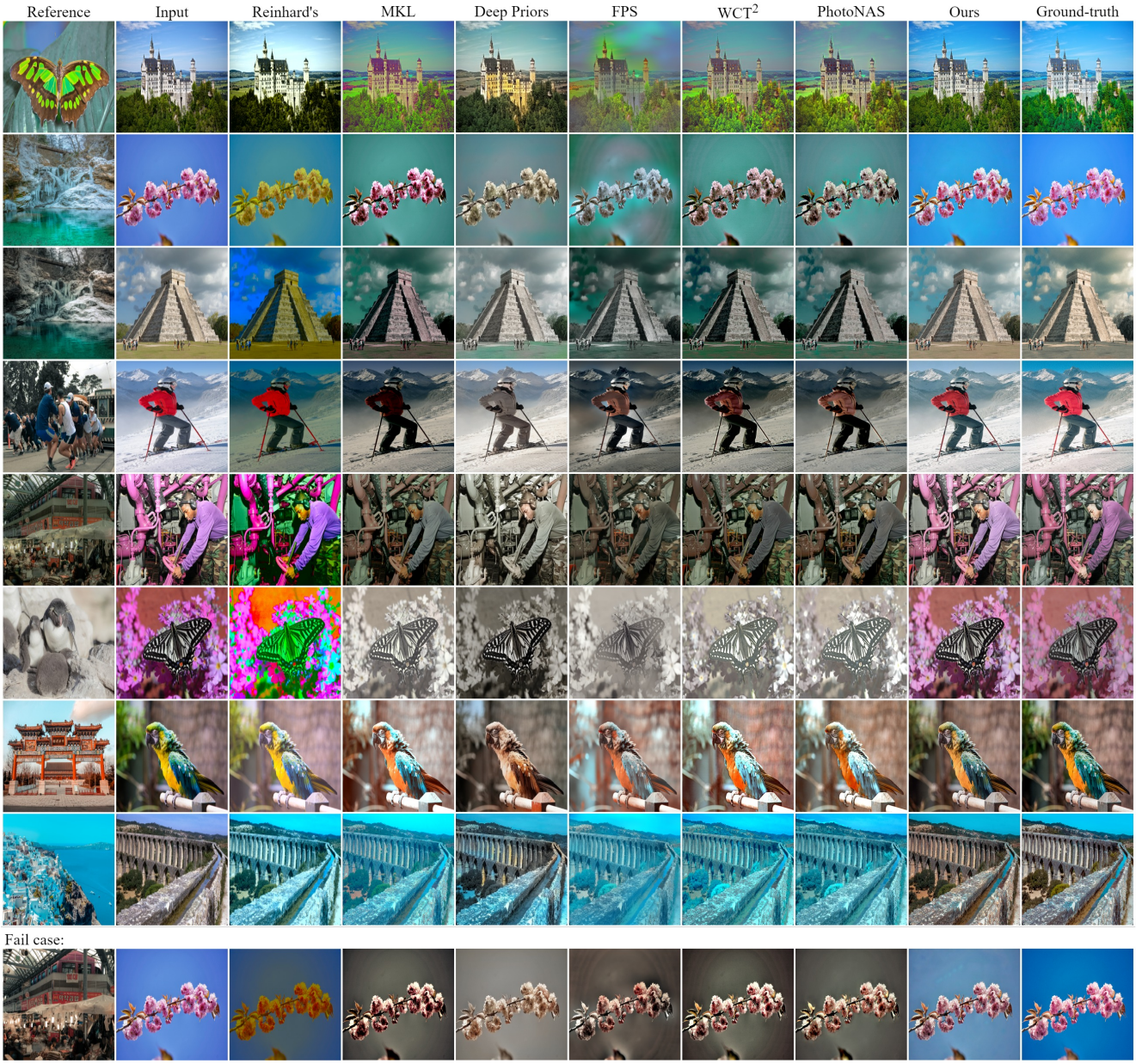

Figure 11. Additional result experimented on 10 contents, 10 references, 10 presets (1000 samples) from DIV2K dataset [9]. The fail case shows our unexpectedly dominant tone from reference.

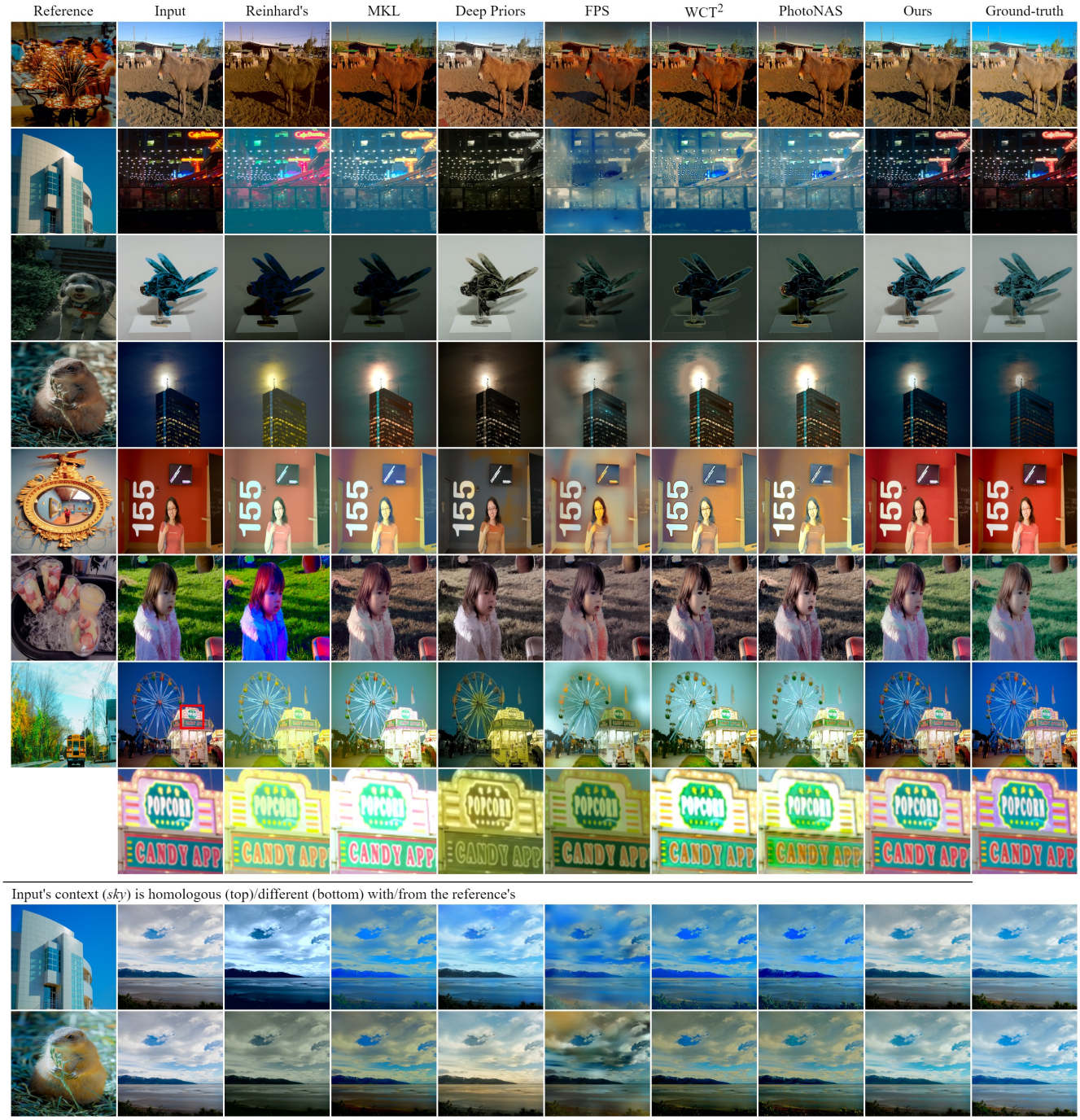

Figure 12. Additional result experimented on 10 contents, 10 references, 10 presets (1000 samples) from Fivek dataset [2] in various contexts (categories). We also evaluate in homologous/different context (*sky*) (*the last two rows*). Even the context is changed, our stylized output still be stable representing a color style.

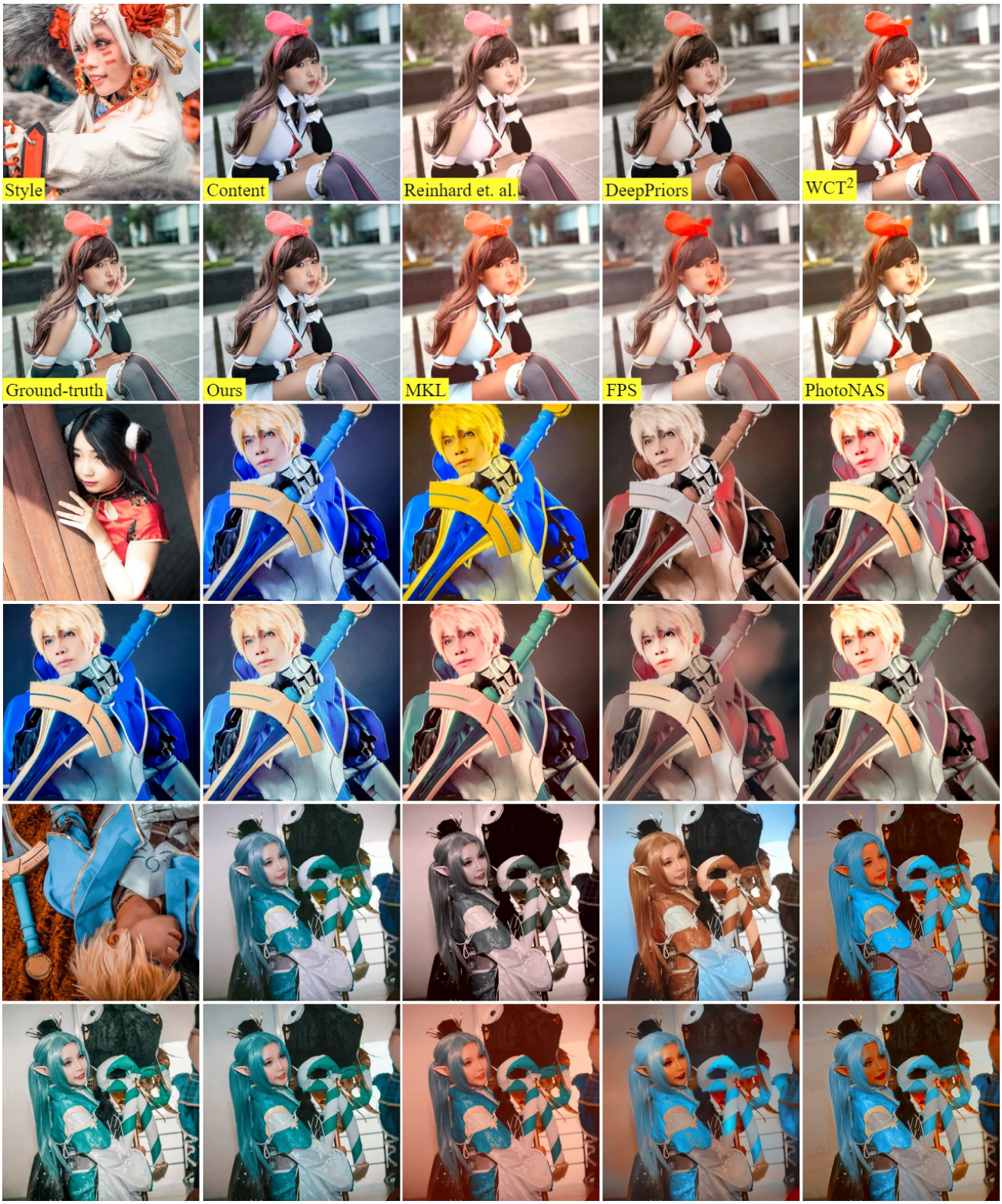

Figure 13. Additional result 1 experimented on 10 contents, 10 references, 10 presets (1000 samples) from Cosplay Portraits.

Figure 14. Additional results 2 experimented on 10 contents, 10 references, 10 presets (1000 samples) from Cosplay Portraits.

Figure 15. Additional results 3 experimented on 10 contents, 10 references, 10 presets (1000 samples) from Cosplay Portraits.

Figure 16. Additional results 4 experimented on 10 contents, 10 references, 10 presets (1000 samples) from Cosplay Portraits.