Enhancing Photo Animation: Augmented Stylistic Modules and Prior Knowledge Integration Supplementary materials

Zhanyi Lu, Yue Zhou(✉), and Ao Chen

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{luzhanyi,zhouyue,llykevin}@sjtu.edu.cn

1 Network Structures

This section introduces trainable network architectures, which are summarized in Tables 1, 2, 3, 4, and 5. These structures encompass the generator, global discriminator, local discriminator, Ada-CTSS module, and style evaluation network.

1.1 Photo Cartoonization Network

Generator The encoder of the generator employs downsampling to reduce the input dimensions to one-fourth of its original size. The IRB section comprises four inverted residual blocks [3]. Subsequently, the decoder restores the input back to its original dimensions. In the table below, CONV-(N,K,S) denotes a convolutional layer with N filters, a kernel size of K, and a stride of S. LReLU represents the leaky ReLU activation [5] function with alpha set to 0.2. Inverted ResBlock-(E,O) denotes an inverted residual block with an expansion ratio of E and an output channel count of O. UP- denotes upsampling using bilinear interpolation, while SCAN represents Style Correlation Adaptive Normalization.

Discriminator The discriminator is composed of both a global and a local discriminator, utilizing a network architecture akin to PatchGAN [2]. Within the following table, BN denotes batch normalization [1].

Ada-CTSS The Ada-CTSS module employs a spatial attention mechanism [4]. In the table below, TopK-S represents partitioning tensors with a stride of S, where the top K values in the patches are selected to form a mask. Subsequently, the mask is element-wise multiplied with the input image to obtain the corresponding patches within the image.

Table 1: The detail of generator architecture

Part	Input-Output Shape	Layer Information
Input	(h,w,3)→(h,w,3)	
Encoder	(h,w,3)→(h,w,32)	CONV-(N32,K7,S1), LReLU
	(h,w,32)→(h/2,w/2,64)	CONV-(N64,K3,S2), LReLU
	(h/2,w/2,64)→(h/2,w/2,64)	CONV-(N64,K3,S1), LReLU
	(h/2,w/2,64)→(h/4,w/4,128)	CONV-(N128,K3,S2), LReLU
	(h/4,w/4,128)→(h/4,w/4,128)	CONV-(N128,K3,S1), LReLU
	(h/4,w/4,128)→(h/4,w/4,128)	CONV-(N128,K3,S1), SCAN, LReLU
IRB	(h/4,w/4,128)→(h/4,w/4,256)	Inverted ResBlock-(E2,O256), SCAN, LReLU
	(h/4,w/4,128)→(h/4,w/4,256)	Inverted ResBlock-(E2,O256), SCAN, LReLU
	(h/4,w/4,128)→(h/4,w/4,256)	Inverted ResBlock-(E2,O256), SCAN, LReLU
	(h/4,w/4,128)→(h/4,w/4,256)	Inverted ResBlock-(E2,O256), SCAN, LReLU
Decoder	(h/4,w/4,256)→(h/4,w/4,128)	CONV-(N128,K3,S1), SCAN, LReLU
	(h/4,w/4,128)→(h/2,w/2,128)	UP-CONV-(N128,K3,S1), LReLU
	(h/2,w/2,128)→(h/2,w/2,128)	CONV-(N128,K3,S1), SCAN, LReLU
	(h/2,w/2,128)→(h,w,64)	UP-CONV-(N128,K3,S1), LReLU
	(h,w,128)→(h,w,64)	CONV-(N64,K3,S1), LReLU
	(h,w,64)→(h,w,64)	CONV-(N64,K3,S1), LReLU
	(h,w,64)→(h,w,64)	CONV-(N32,K7,S1), LReLU
	(h,w,64)→(h,w,3)	CONV-(N3,K1,S1), Tanh

Table 2: The detail of global discriminator architecture

Part	Input-Output Shape	Layer Information
Input	(h,w,3)→(h,w,3)	
Encoder	(h,w,3)→(h/2,w/2,32)	CONV-(N32,K3,S2),LReLU
	(h/2,w/2,32)→(h/2,w/2,32)	CONV-(N32,K3,S1),BN,LReLU
	(h/2,w/2,32)→(h/4,w/4,64)	CONV-(N64,K3,S2),LReLU
	(h/4,w/4,64)→(h/4,w/4,64)	CONV-(N64,K3,S1),BN,LReLU
	(h/4,w/4,64)→(h/8,w/8,128)	CONV-(N128,K3,S2),LReLU
	(h/8,w/8,128)→(h/8,w/8,128)	CONV-(N128,K3,S1),BN,LReLU
	(h/8,w/8,128)→(h/8,w/8,1)	CONV-(N1,K3,S2)

Table 3: The detail of local discriminator architecture

Part	Input-Output Shape	Layer Information
Input	(h/4,w/4,3)→(h/4,w/4,3)	
Encoder	(h/4,w/4,3)→(h/4,w/4,32)	CONV-(N32,K3,S1),BN,LReLU
	(h/4,w/4,32)→(h/8,w/8,64)	CONV-(N64,K3,S2), BN,LReLU
	(h/8,w/8,64)→(h/16,w/16,128)	CONV-(N128,K3,S2), BN,LReLU
	(h/16,w/16,128)→(h/16,w/16,256)	CONV-(N256,K3,S2), BN,LReLU
	(h/16,w/16,256)→(h/16,w/16,1)	CONV-(N1,K3,S2)

Table 4: The detail of Ada-CTSS architecture

Part	Input-Output Shape	Layer Information
Input	(h,w,8)→(h,w,8)	
Spatial Attention	(h,w,3)→(h,w,8)	CONV-(N8,K3,S1)
	(h,w,8)→(h,w,2)	AvgPool&MaxPool,Concate
	(h,w,2)→(h,w,1)	CONV-(N2,K1,S1),Sigmoid
TopK	(h,w,1)→(h,w,1)	TopK-S4
	(h,w,1)→(h/4,w/4,K)	EM

1.2 Style Evaluation Network

The style evaluation network and the photo cartoonization network use different data sources. For the anime dataset, we collected works from three categories: "Your Name" by Shinkai, "The Wind Rises" by Hayao, and "Paprika" by Satoshi Kon. Each category consists of about 1500 images with a resolution of 256*256 pixels. The photo dataset is comprised of the remaining data from [6].

We employed the Adam optimizer [32] to optimize the style evaluation network with a learning rate of 0.003. In the table below, we use the following abbreviations: GAP for global average pooling, FC for fully connected layers, and CLS to represent the number of anime classes.

Table 5: The detail of style evaluation network architecture

Part	Input-Output Shape	Layer Information
Input	(h,w,3)→(h,w,3)	
Encoder	(h,w,3)→(h/2,w/2,16)	CONV-(N16,K3,S2),LReLU
	(h/2,w/2,16)→(h/4,w/4,32)	CONV-(N32,K3,S2),LReLU
	(h/4,w/4,32)→(h/4,w/4,64)	CONV-(N64,K3,S2),LReLU
	(h/4,w/4,64)→(1,1,64)	GAP
	(1,1,64)→(N_CLS,)	FC-(64,N_CLS)

2 Analysis of the Style Evaluation Network

To further validate the reliability of features learned in the style evaluation network for anime style compared to FID, we conducted an analysis experiment as in Table 6.

Results show that directly freezing the InceptionV3 backbone for discriminating between photographs and anime images yields the lowest accuracy. This suggests that InceptionV3's features learned on ImageNet are unreliable for distinguishing photo and anime styles. Thus, FID is unsuitable as a quantitative style evaluation metric.

Table 6: Analysis of Style Evaluation Network, "frozen" denotes frozen backbone and trained classification layer only.

Model	+Edge	+Surface	+Texture	+Structure	Accuracy
InceptionV3(frozen)					87.62
InceptionV3(unfrozen)					**99.15**
Style Evaluation Network					97.32
	√				98.26
		√			98.69
			√		98.26
				√	98.71
		√	√		98.76
	√	√	√		98.91
		√	√	√	99.20
	√	√	√	√	**99.24**

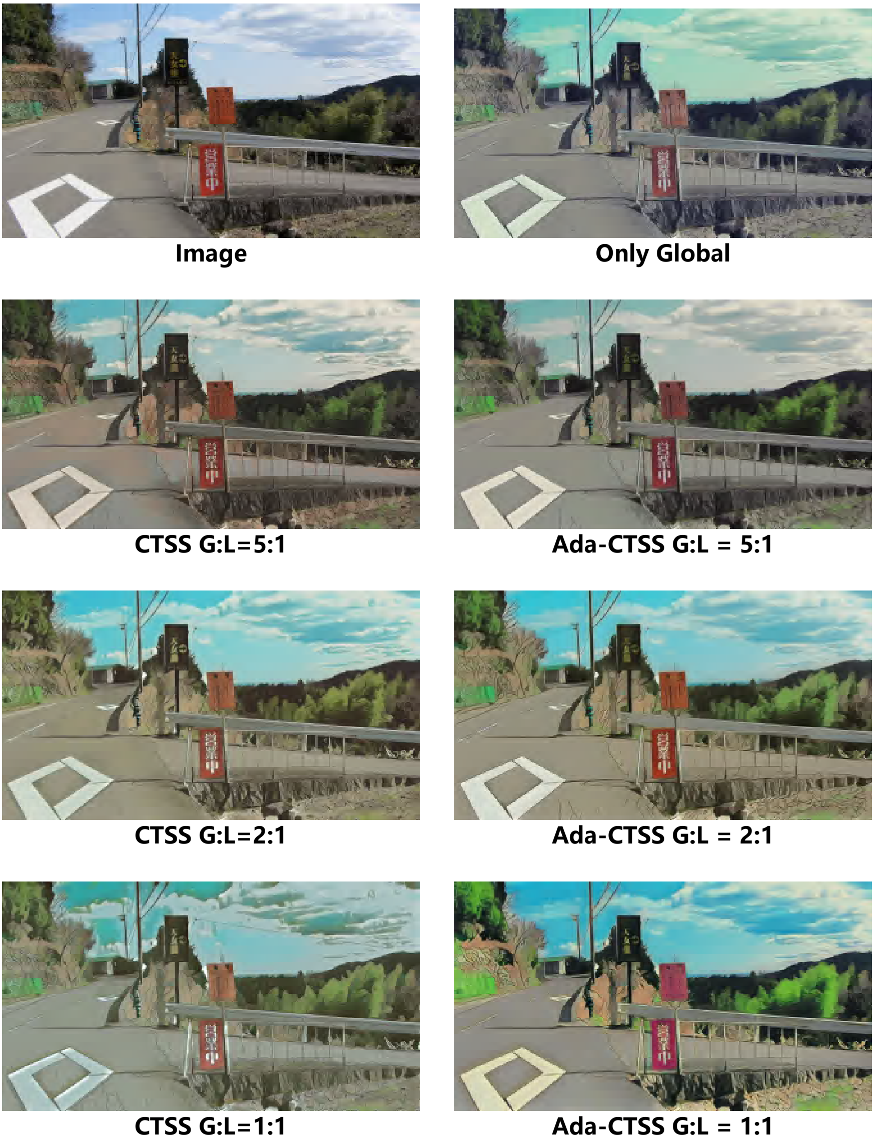
In contrast, the concise style evaluation network proposed in the paper achieves high accuracy, indicating it learned more discriminative features for photo-anime style discrimination. Combining anime-relevant representations achieves accuracy comparable to the unfrozen InceptionV3 network, albeit with only three convolutional layers. This reduces training and testing time compared to InceptionV3's deep structure.

3 Analysis of Global and Local Discriminator

This section examined the influence of the weights assigned to the global and local discriminators on the training outcomes. We evaluated the performance of CTSS and Ada-CTSS under different weight ratios for the global and local components.

Figure 1 and Figure 3 illustrate the results when the weight assigned to the global discriminator outweighs that of the local discriminator. Sole reliance on the global discriminator fails to adequately capture stylistic effects regarding local details, such as color, texture, and surface characteristics. Increasing the weight of the local discriminator enhances the stylistic effects in these aspects. However, a further increase in the weight of the local discriminator (Figure 2 and Figure 4) leads to potential over-stylization in localized regions, possibly due to the diminished capacity of the global discriminator to consider the overall image structure.

Moreover, it is noteworthy that Ada-CTSS, compared to CTSS, exhibits more substantial stylistic effects in color and texture when the global discriminator is relatively weak. Additionally, cluttered textures are reduced when the local discriminator is relatively strong. This highlights the adaptive nature of Ada-CTSS in selecting suitable local regions to enhance stylistic effects and its responsiveness to hyperparameter choices.



Image

Only Global

CTSS G:L=5:1

Ada-CTSS G:L = 5:1

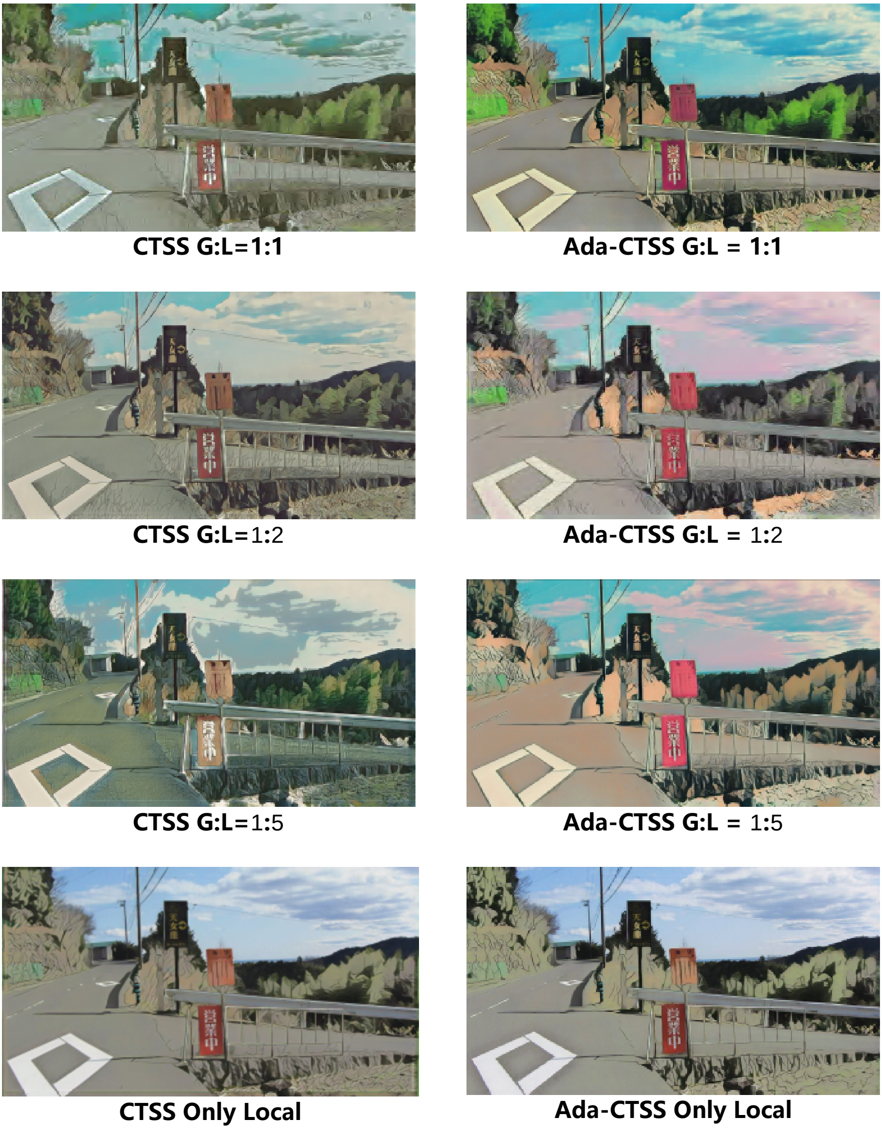
CTSS G:L=2:1

Ada-CTSS G:L = 2:1

CTSS G:L=1:1

Ada-CTSS G:L = 1:1

Fig. 1: Global and Local Weight Ratio Study (G:L represents weight proportion).



CTSS G:L=1:1

Ada-CTSS G:L = 1:1

CTSS G:L=1:2

Ada-CTSS G:L = 1:2

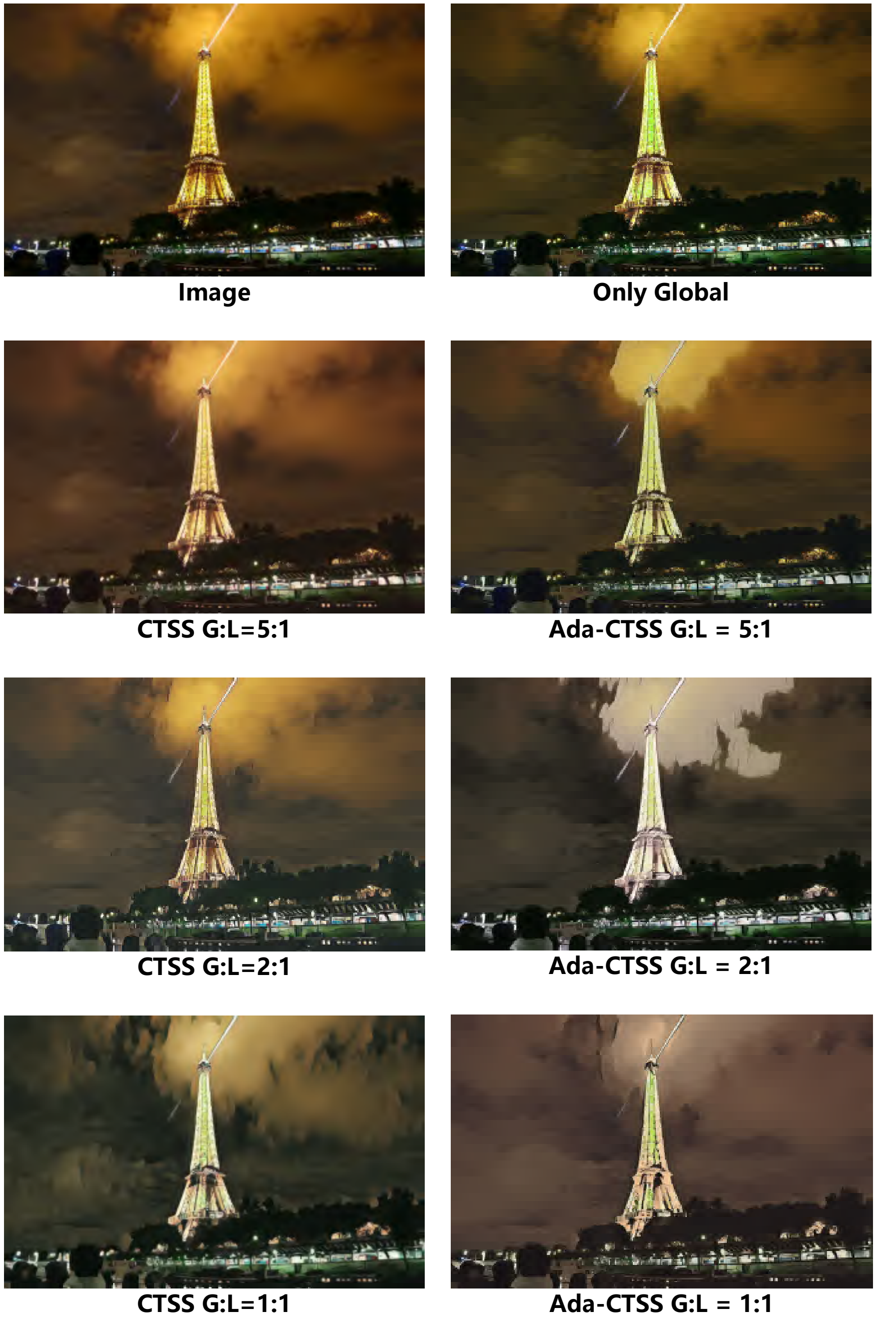
CTSS G:L=1:5

Ada-CTSS G:L = 1:5

CTSS Only Local

Ada-CTSS Only Local

Fig. 2: Global and Local Weight Ratio Study (G:L represents weight proportion).



Image

Only Global

CTSS G:L=5:1

Ada-CTSS G:L = 5:1

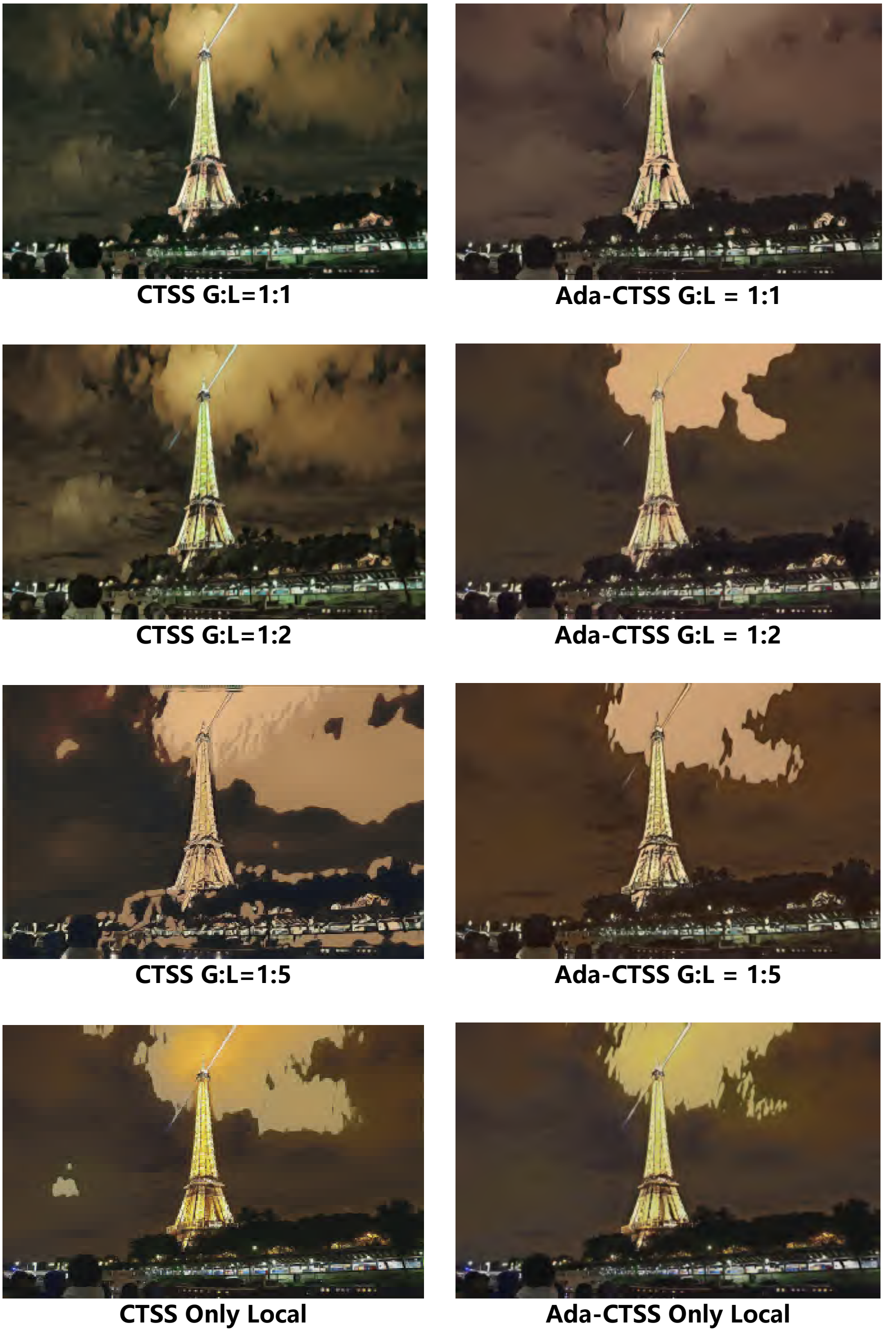
CTSS G:L=2:1

Ada-CTSS G:L = 2:1

CTSS G:L=1:1

Ada-CTSS G:L = 1:1

Fig. 3: Global and Local Weight Ratio Study (G:L represents weight proportion).



CTSS G:L=1:1

Ada-CTSS G:L = 1:1

CTSS G:L=1:2

Ada-CTSS G:L = 1:2

CTSS G:L=1:5

Ada-CTSS G:L = 1:5

CTSS Only Local

Ada-CTSS Only Local

Fig. 4: Global and Local Weight Ratio Study (G:L represents weight proportion).

4 Analysis of Refined Color Preservation Loss

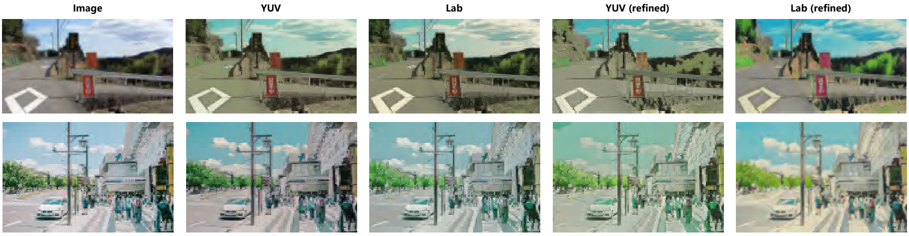


Fig. 5: Study on Refined Color Preservation Loss

This section investigates the impact of refined color preservation loss on photo-to-anime transformation. The effects are studied in the YUV and Lab color spaces, with a particular focus on the role of the enhanced loss formulation. The experimental results are visually depicted in Figure 5.

Without refined formulation, direct color loss computation between the original and generated images diminishes the stylization effect, resulting in generated images that closely resemble the colors of the originals. However, employing refined color preservation loss enables indirect calculation of color loss between anime images, allowing the generated images to adopt the color distribution of anime. The Lab color space, offering a more comprehensive color range than YUV, produces more vibrant and anime-like colors in the generated images.

5 More Results

References

1. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML) (2015) 1
2. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 1
3. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1
4. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) 1
5. Xu, B., Wang, N., Chen, T., Li, M.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 32nd International Conference on Machine Learning (ICML) (2015) 1

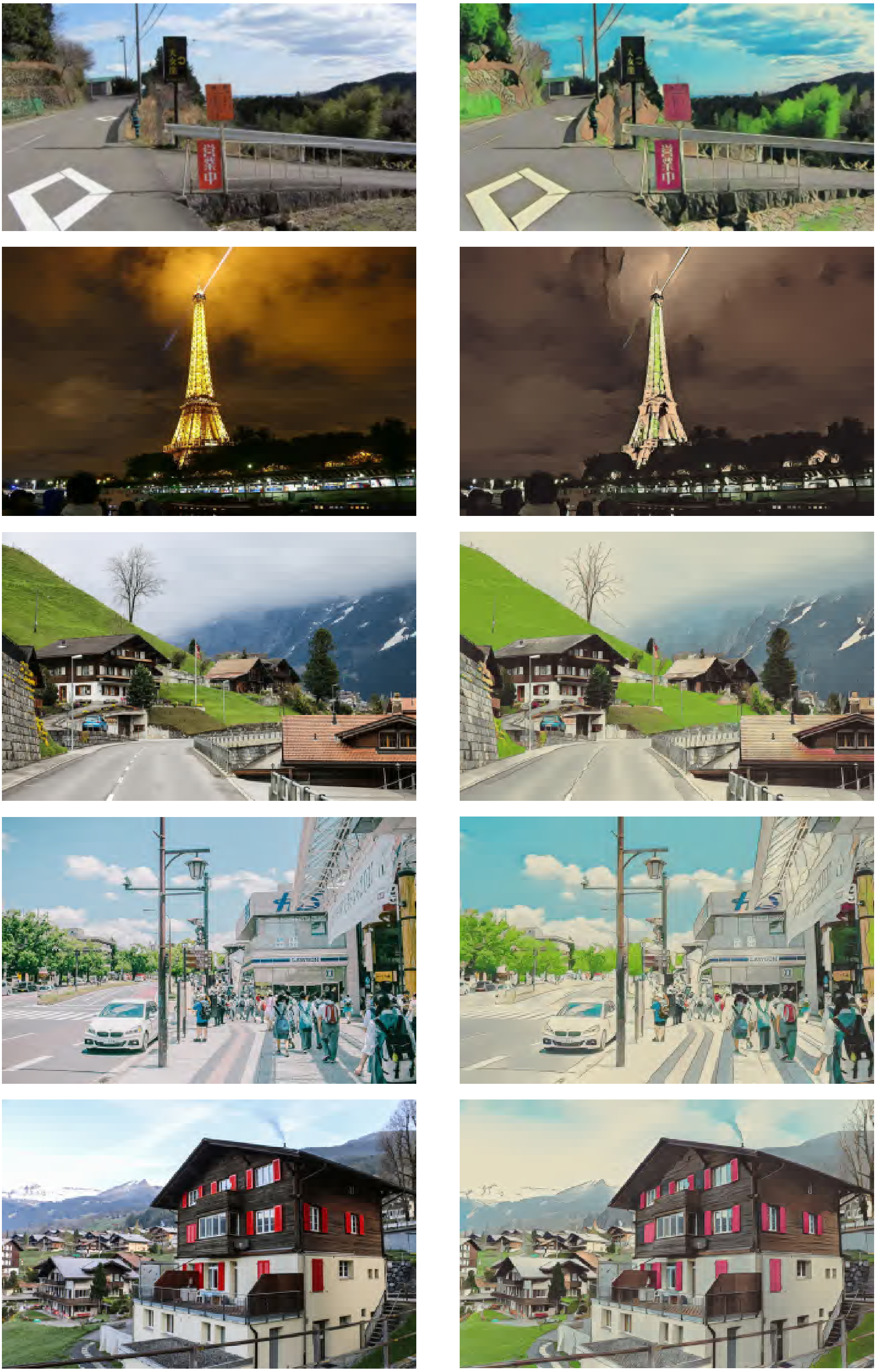


Fig. 6: More Results on Hayao Style.

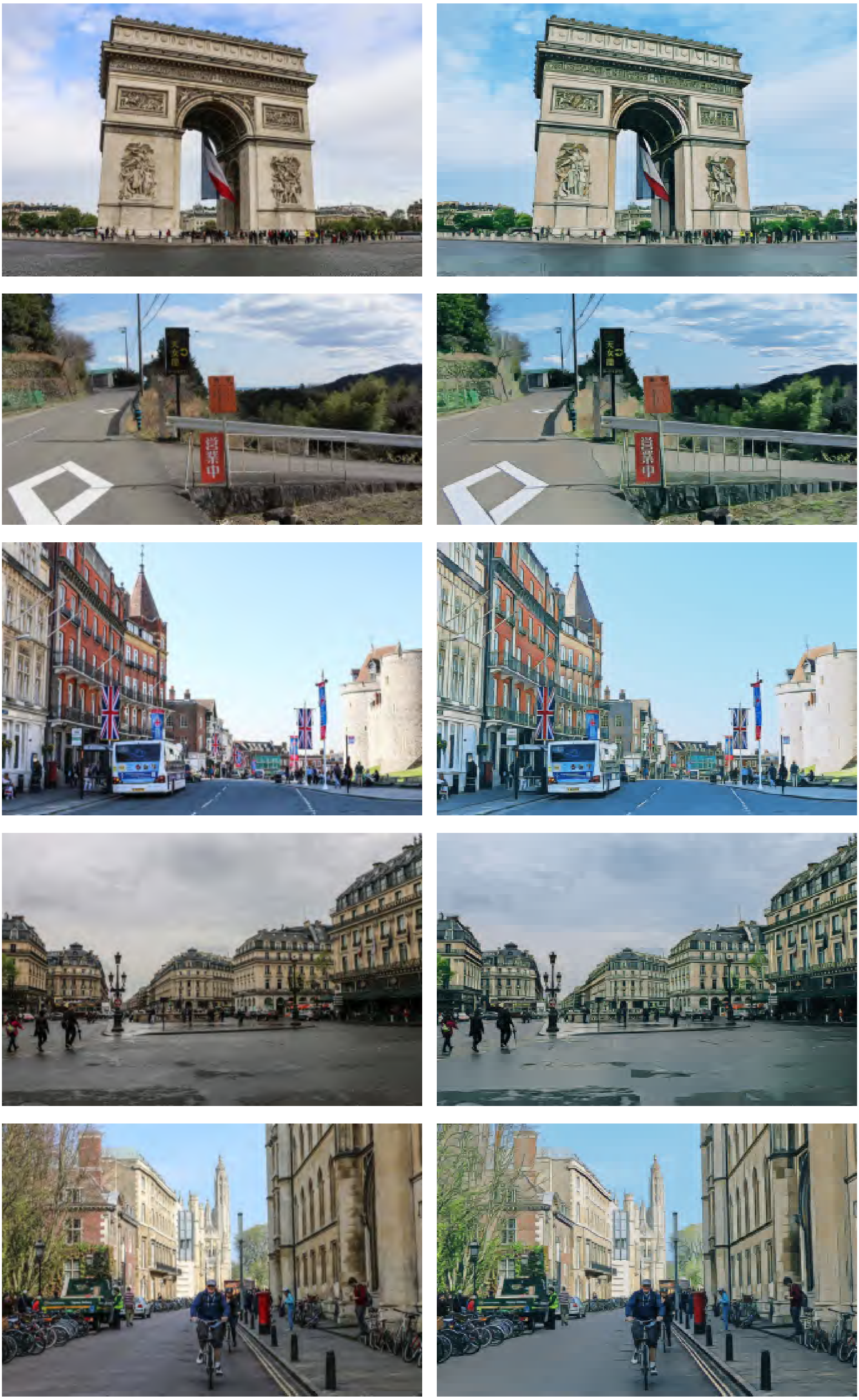


Fig. 7: More Results on Shinkai Style.