A. User Study

Considering the subjective nature of style transfer problem, we conduct user study to evaluate our model against related methods. Based on our collected 300 real-world-scene high-resolution photos, we generated cartoonized results with different methods. First, we showed participants a content image, i.e., a real photo. Second, we showed them two cartoonization results generated by our method and a random contrast method. Finally, we asked the participants which result has better cartoon effect. We separately repeated the above process for "The Wind Rises", "Dragon Ball", and "Crayon Shin-chan" datasets, respectively. For results of each dataset, we collected 2400 votes from 40 participants and present the voting results in Fig.11 , which shows percentage of preference of our method against related methods on different datasets. Overall, our method gained the most user preference votes, indicating the superiority of our model from subjective perspective.

Besides, we sample 40 high-resolution photos and evaluate corresponding cartoonized results trained over "The Wind Rises" dataset using CartoonGAN, AnimeGAN, WhiteBox, and our method. Then, we allow 20 participants to score them with 1-10 ratings from three dimensions: (1) style saliency, (2) abstraction degree, (3) content integrity. Finally, we calculate the average score across all samples over each dimension. Results reported in Fig. 10 indicate that our method has obvious advantages in presenting cartoon abstraction and vividness than related advanced methods. On the other hand, it reflects that despite significant superiority in style saliency, our method compromises some content integrity, due to the inherent content-style trade-off in style transfer related problem. We will focus on improving this in future work.

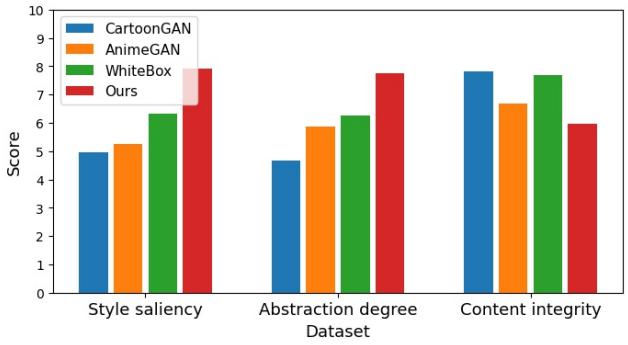


Figure 10. User preference scores of cartoonization results of different methods with respect to "style saliency", "abstraction degree", and "content integrity".

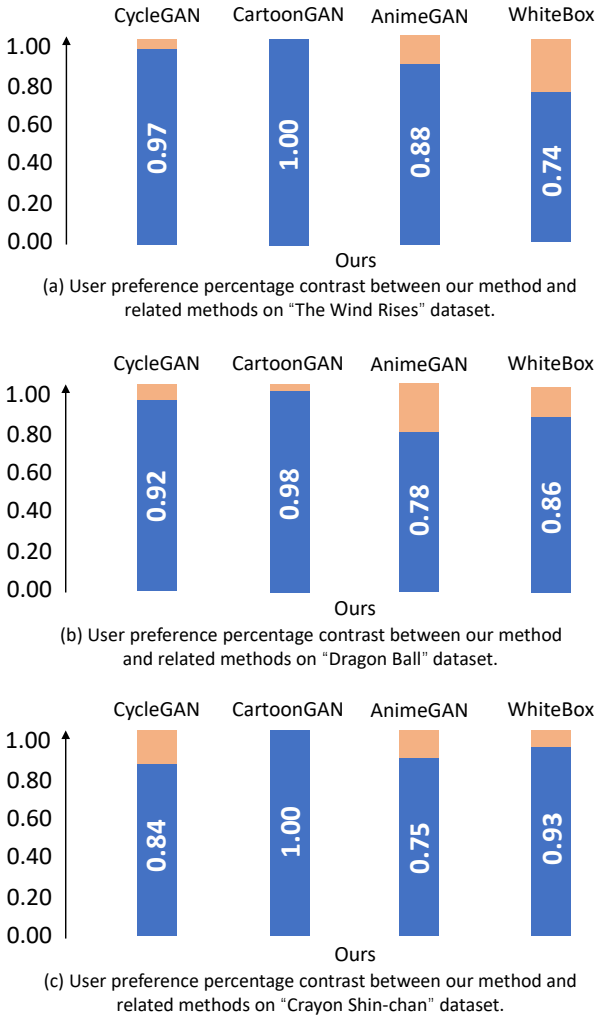


Figure 11. Percentage of user preference voting of our method against related methods on different cartoon datasets.

B. Network Details

The generator G follows an autoencoder structure where we use four residual blocks to bridge a downsampling path and an upsampling path. Both the image-level discriminator D_{img} and the patch-level discriminator D_{patch} adopt the PatchGAN (Isola et al., 2017) structure. Notations and network details are listed in Fig. 12.

C. More Qualitative Results

We append more high-resolution cartoonization results of our model evaluated on different datasets in this section. For all the produced results, the typical cartoon styles are sufficiently transferred.

Notations	
LN: layer normalization	lRelu: leaky Relu with α = 0.2
Conv_n(N)k(K)s(S): Convolutional layer with N filters, K×K kernel size, and stride S	
Upsample_n(N): Upsampling module consisting of following layers: nearest neighbor upsampling with factor 2→Conv_n(N)k(3)s(1)→LN→lRelu	
DConv_n(N)k(K)s(S): Depthwise convolutional layer with N filters, K×K kernel size, and stride S	

Generator (G)

Layers	Shape
Input	256x256x3
Conv_n(32)k(7)s(1), LN, lRelu	256x256x32
Conv_n(64)k(3)s(2), LN, lRelu	128x128x64
Conv_n(64)k(3)s(1), LN, lRelu	128x128x64
Conv_n(128)k(3)s(2), LN, lRelu	64x64x128
Conv_n(256)k(3)s(1), LN, lRelu	64x64x256
ResBlock	64x64x256
ResBlock	64x64x256
ResBlock	64x64x256
ResBlock	64x64x256
Conv_n(128)k(3)s(1), LN, lRelu	64x64x128
Upsample_n(128)	128x128x128
Conv_n(128)k(3)s(1), LN, lRelu	128x128x128
Upsample_n(64)	256x256x64
Conv_n(64)k(3)s(1), LN, lRelu	256x256x64
Conv_n(64)k(3)s(1), LN, lRelu	256x256x64
Conv_n(32)k(7)s(1), LN, lRelu	256x256x32
Conv_n(3)k(1)s(1), Tanh	256x256x3

Image-level discriminator (D_{img})

Layers	Shape
Input	256x256x3
Conv_n(32)k(3)s(1), LN, lRelu	256x256x32
Conv_n(64)k(3)s(2), LN, lRelu	128x128x64
Conv_n(128)k(3)s(2), LN, lRelu	64x64x128
Conv_n(256)k(3)s(2), LN, lRelu	32x32x256
Conv_n(1)k(3)s(1)	32x32x1

Patch-level discriminator (D_{patch})

Layers	Shape
Input	96x96x1
Conv_n(16)k(3)s(1), LN, lRelu	96x96x16
Conv_n(32)k(3)s(2), LN, lRelu	48x48x32
Conv_n(64)k(3)s(2), LN, lRelu	24x24x64
Conv_n(128)k(3)s(2), LN, lRelu	12x12x128
Conv_n(1)k(3)s(1)	12x12x1

ResBlock

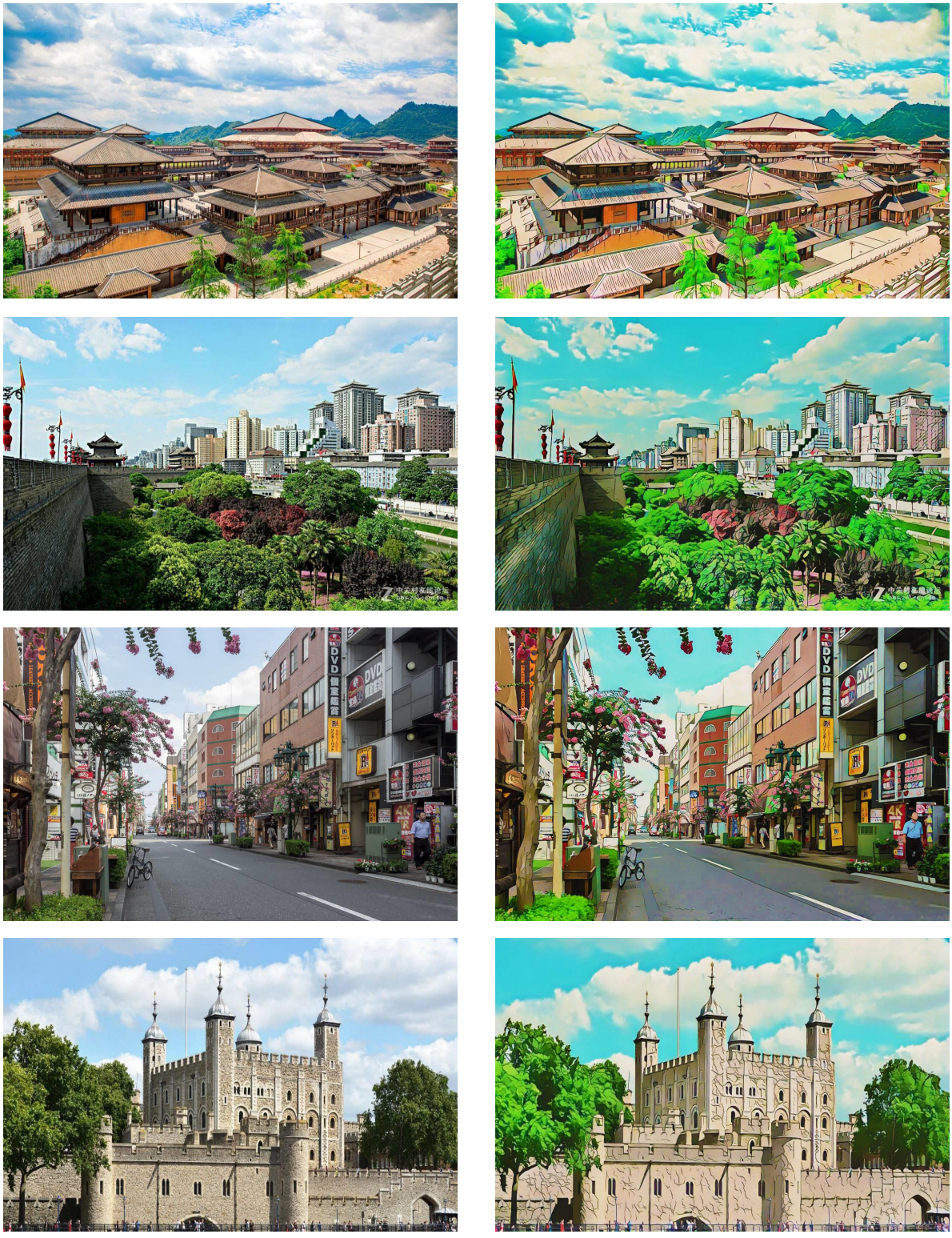
Layers	Shape
Input	64x64x256
Conv_n(128)k(1)s(1), LN, lRelu	64x64x128
DConv_n(128)k(3)s(1), LN, lRelu	64x64x128
Conv_n(256)k(1)s(1), LN	64x64x256
Add input	64x64x256

Figure 12. Network details of the generator G, the image-level discriminator D_{img}, and the patch-level discriminator D_{patch}.

More cartoonization results on “The Wind Rises” dataset

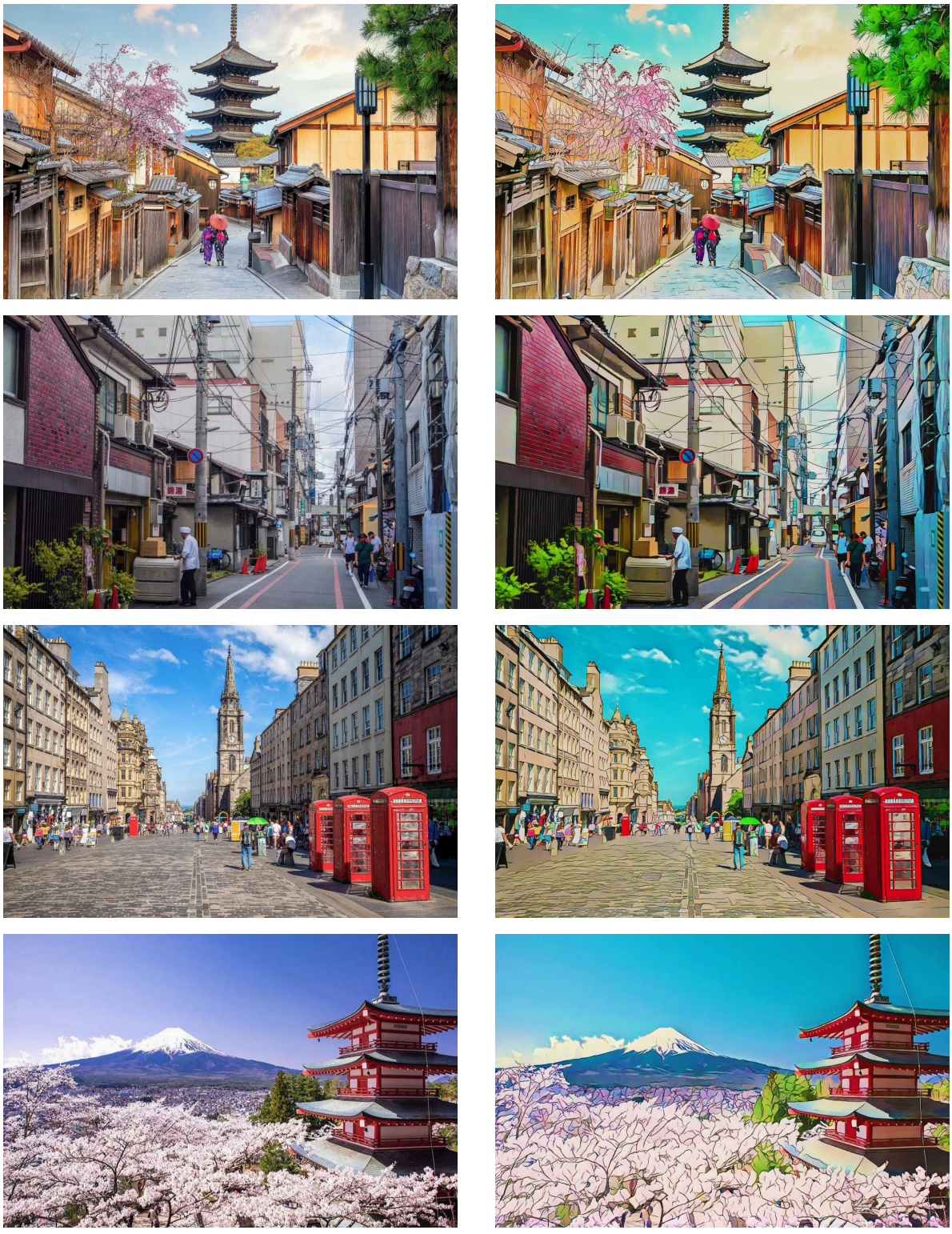
Input Photo Cartoonized Results

More cartoonization results on “The Wind Rises” dataset



Input Photo Cartoonized Results

More cartoonization results on “The Wind Rises” dataset



Input Photo Cartoonized Results

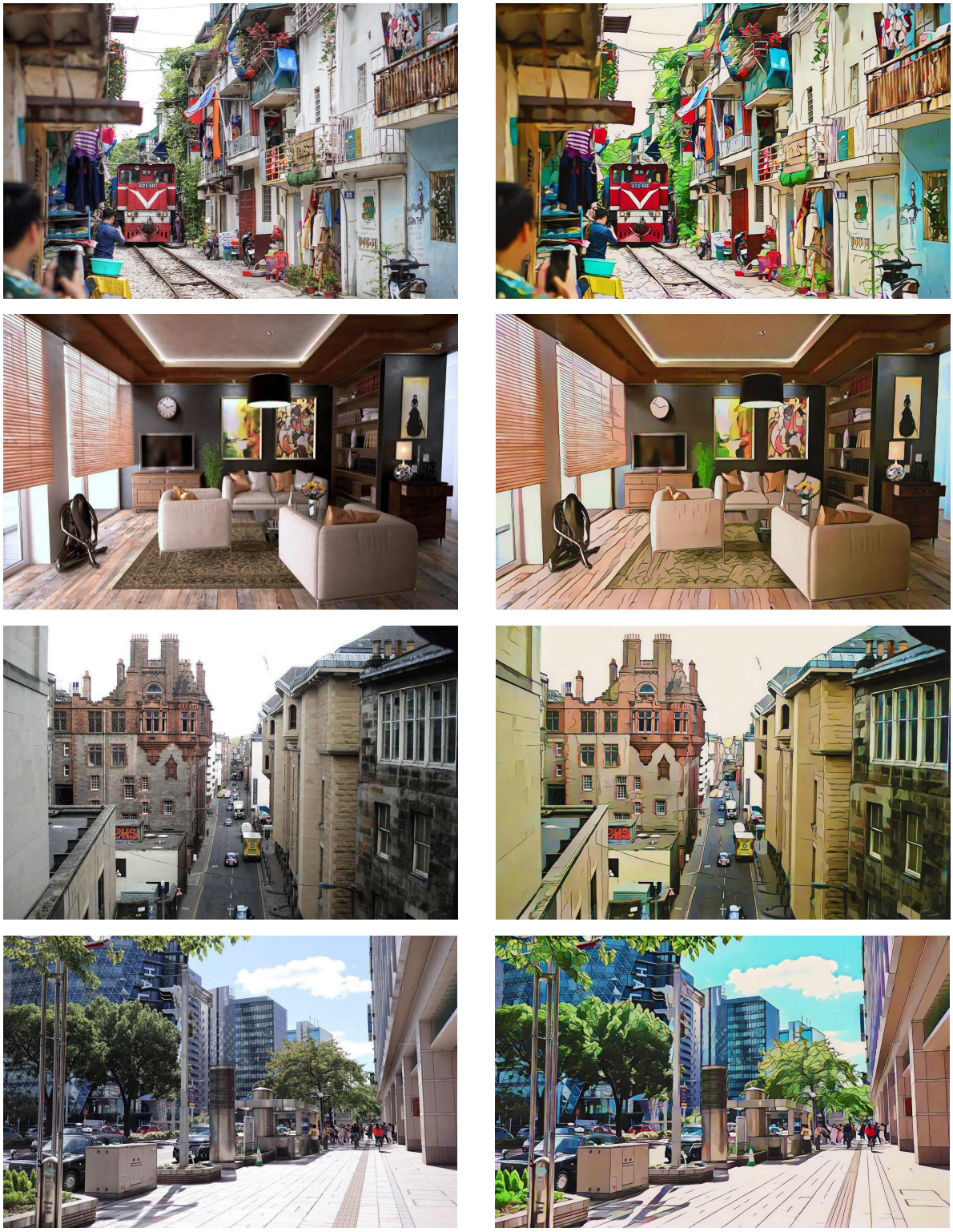
More cartoonization results on “The Wind Rises” dataset

Input Photo Cartoonized Results

More cartoonization results on “The Wind Rises” dataset

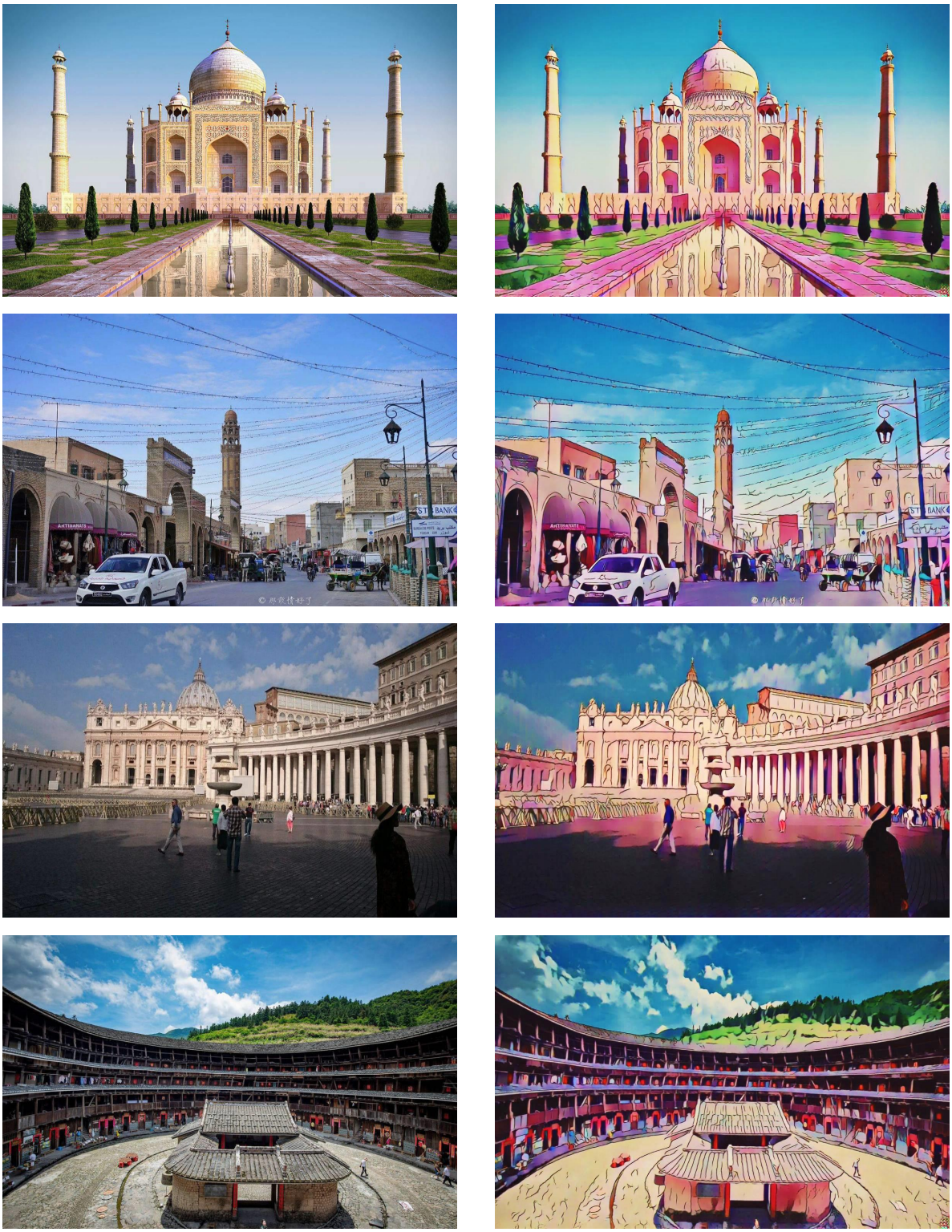
Input Photo Cartoonized Results

More cartoonization results on “The Wind Rises” dataset



Input Photo Cartoonized Results

More cartoonization results on “Dragon Ball” dataset



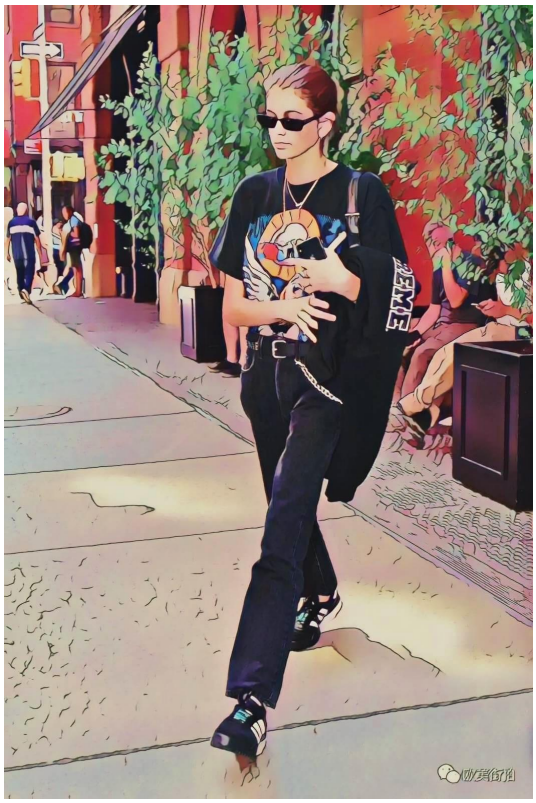
Input Photo Cartoonized Results

More cartoonization results on “Dragon Ball” dataset

More cartoonization results on “Dragon Ball” dataset

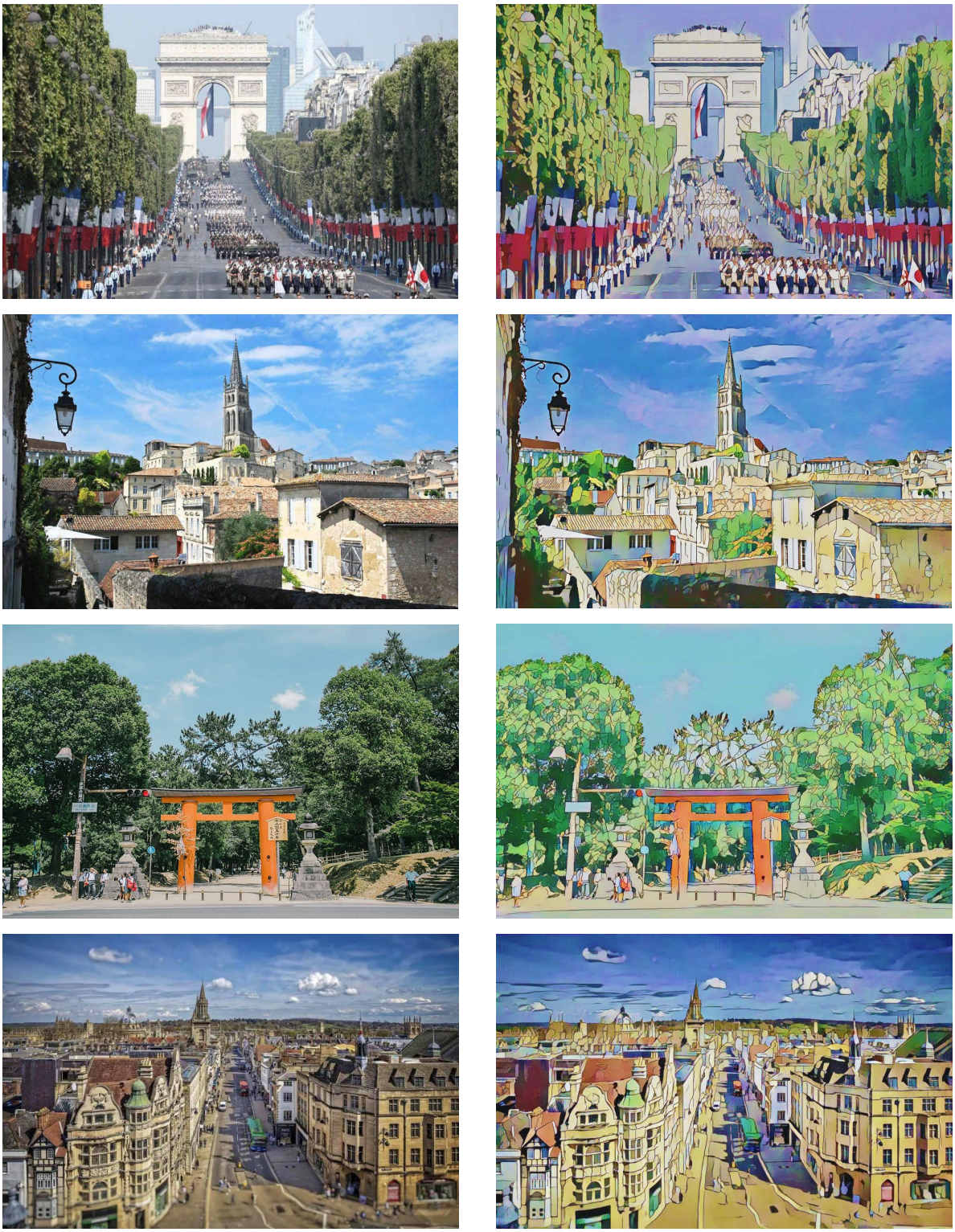






Input Photo Cartoonized Results

More cartoonization results on “Crayon Shin-chan” dataset



Input Photo Cartoonized Results

More cartoonization results on “Crayon Shin-chan” dataset

Input Photo Cartoonized Results

More cartoonization results on “Crayon Shin-chan” dataset

Input Photo Cartoonized Results