Semantic-aware Cartoon Style Transfer

Jinbin Bai
Nanjing University, China
jinbin5bai@gmail.com

Abstract

In the community of computer vision, style transfer usually refers to the extraction of texture information from the source image and then transfer and synthesis into the target image. However, the existing style transfer methods ignore information such as the hierarchical structure of the input image, and consider the entire parts of input image belong to the same distribution. The results show that a trained generative adversarial network can only convert the entire image from one distribution to another specific distribution, which cannot produce satisfactory results for images with complex hierarchies and structures.

In this paper, we propose a new method for specific style transfer by combining different semantic parts, such as foreground characters and background content, with different representation weights, which is of great significance in the fields of painting and cartoon generation. We assist in training adversarial networks by adding a pre-trained semantic segmentation network, and control and adjust our framework by observing different semantic parts of the generated image and comparing their distribution with the various distributions we hope to obtain.

Qualitative comparison and quantitative analysis show that our method significantly improves the sense of hierarchy of the generated image, and different parts of the image have different textures, structures and color distributions. For example, our method makes the texture of the generated foreground characters more dense, while the texture of the background content becomes sparse, which makes the generated image look better than the state-of-the-art methods. Finally, the ablation study proved the impact of each component in our framework.

The code will be made publicly available at https://github.com/JB-Bai/Semantic-aware-Cartoon-Style-Transfer.

1. Introduction

In the community of computer vision, style transfer usually refers to the extraction of texture style information (i.e., painting) from the source image and then transfer and synthesis into a stylized output with the content information (i.e., an arbitrary portrait or landscape) of the target image.

In the past, re-drawing an image in a particular style requires a well-trained artist and lots of time. Since the mid-1990s, the art theories behind the appealing artworks have been attracting the attention of not only the artists but many computer science researchers. There are plenty of studies and techniques exploring how to automatically turn images into synthetic artworks. As a result, style transfer is usually studied as a generalised solution to these problems. However, the common limitation of traditional computer vision methods is that they only use low-level image features (i.e., a histogram or feature matching) and often fail to capture image structures effectively.

Recently, inspired by the power of Convolutional Neural Networks (CNNs)[11], Gatys et al. [5] first studied how to use a CNN to reproduce famous painting styles on natural images. They proposed to model the content of one image as the feature responses from a pre-trained CNN, and further model the style of an artwork as the summary feature statistics. Their experimental results demonstrated that a CNN is capable of extracting content information from an arbitrary photograph and style information from a well-known artwork.

Since the algorithm of Gatys et al. does not have any explicit restrictions on the type of style images and also does not need ground truth results for training, it breaks the constraints of previous approaches. The work of Gatys et al. opened up a new field called Neural Style Transfer (NST), which is the process of using Convolutional Neural Network to render a content image in different styles.

However, style is an abstract concept and is formed by a complex combination of low-level features and high-

Figure 1. Semantic mismatch of style transfer result. the images from left to right are the content image, the style image and the result, respectively.

level semantics. For instance, mapping the texture of a brick to an apple is unnatural in terms of semantics. Consequently, the results of a style transfer look unrealistic. Meanwhile, the texture of a brick when mapped onto a wall is natural and acceptable from the perspective of an ordinary observer. However, traditional style transfer methods exhibit a apparently semantic mismatch. This yields stylized images that look very different from the users' expectations. Fig. 1 shows an example of a failure caused by a semantic mismatch. Although the general style of the source was adequately reflected in the target, its visual effect was unrealistic because the styles were transferred from different semantic regions. For example, the road region received portions of the sky color, which is a consequence of utilizing spatial-invariant feature statistics, such as a Gram matrix, as described by Li et al. [12]. Because a Gram matrix is used to model spatially invariant characteristics extracted from a CNN, it can represent the local style of a source image. Meanwhile, the geometrical content or spatial semantics are ignored.

Cartoon is a popular art form that has been widely applied in diverse scenes. Modern cartoon animation workflows allow artists to use a variety of sources to create content. Some famous products have been created by turning real-world photography into usable cartoon scene materials, where the process is called image cartoonization.

Generative Adversarial Network(GAN)[6] is a state-of-the-art generative model that can generate data with the same distribution of input data by solving a min-max problem between a generator network and a discriminator network. It is powerful in image synthesis by forcing the generated images to be indistinguishable from real images. GAN has been widely used in conditional image generation tasks, such as style transfer [5], image cartoonization [4], etc.

In this study, we focus on resolving a semantic mismatch in specifical style transfer —- image cartoonization, which meansbetween a source (style) image and a target (content) image. In our method, we adopt adversarial training architecture and use two discriminators to enforce the generator network to synthesize images with the same distribution as the target domain. Besides, we adapt an existing semantic segmentation method to extract semantically meaningful regions from both the output images and the style (cartoon) images. To conclude, our contributions are as follows:

- We propose a semantic module for describing various semantic regions and matching these regions in different image sets. The proposed semantic loss can effectively release the mismatching issues.
- A GAN-based image cartoonization framework is optimized with the guide of semantic module for different semantic regions.
- Extensive experiments have been conducted to show that our method can generate high-quality cartoonized images. Our method outperforms existing methods in qualitative comparison and quantitative comparison.

2. Related Work

2.1. Style Transfer

Inspired by the power of Convolutional Neural Networks (CNNs), Gatys et al. [5] first proposed to exploit CNN feature activations to recombine the content of a given photo and the style of famous artworks. The key idea behind their algorithm is to iteratively optimize an image with the objective of matching desired CNN feature distributions, which involves both the image's content information and artwork's style information. Their proposed algorithm successfully produces

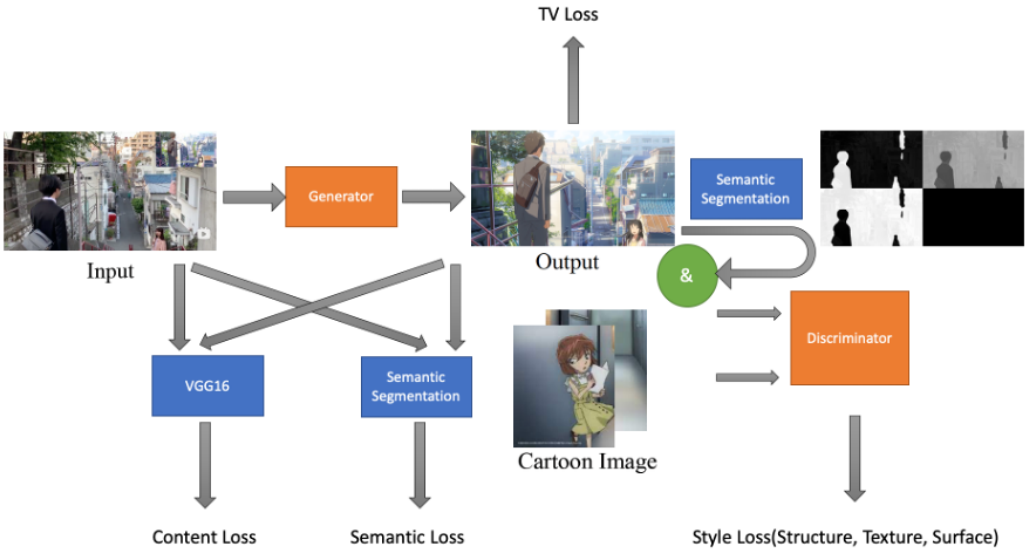


Figure 2. An overview of our proposed network.

stylised images with the appearance of a given artwork.

The first model-optimization-based neural style transfer algorithm is proposed by Johnson et al. [10]. The main idea of this work is to pre-train a feed-forward style-specific network and produce a stylized result with a single forward pass at testing stage. Johnson et al. [10]'s design roughly follows the network proposed by Radford et al. [15] but with residual blocks as well as fractionally strided convolutions.

The algorithms of Johnson et al. [10] achieves a real-time style transfer. However, His algorithm design basically follows the algorithm of Gatys et al. [5], which makes them suffer from the same aforementioned issues as Gatys et al. [5]'s algorithm (e.g., a lack of consideration in the coherence of details and depth information).

2.2. Generative Adversial Network

Different from other generative models, e.g. variational auto-encoder, generative adversarial networks is a more powerful and excellence framework to synthesis a sharp, realistic and photographic image result. GAN [6] is proposed by Goodfellow to introduce an adversarial process between the two related neural networks, e.g. generator and discriminator, which can be treated as a minimax two-player game which reaching the Nash balance for generator and discriminator. Recently, with its rapid development and optimization, GANs [22, 9, 2] achieve a state of art in style transfer, image painting, etc. For instance, Pix2pix [9] learns image to image translation task in a supervised manner. It combines an adversarial loss with L1 loss, thus requires paired data samples. To alleviate the problem of obtaining data pairs, CycleGAN [22] preserve key attributes between the input and the translated image by using a cycle consistency loss.

Besides, CartoonGAN [4] is designed for image cartoonization, in which a GAN framework with a novel edge loss is proposed, and achieves good results in certain cases. But using a black-box model to directly fit the training data decreased its generality and stylization quality, causing bad cases in some situations. To address the above-mentioned problems, Wang et al. [19] made extensive observations on human painting behaviors and cartoon images of different styles, then propose to decompose images into several cartoon representations, including the surface representation, the structure representation, and the texture representation. Users can adjust the style of model output by balancing the weight of each representation. In order to make the image color reconstruction better, AnimeGAN [3] first convert the image color in RGB format to the YUV format to build the color reconstruction loss.

3. Proposed Method

The architecture of our image cartoonization framework is shown in Fig. 2. The backbone image cartoonization (specifical style transfer) model is a GAN framework comprises a generator G and couples of discriminators D_i, with the semantic module to extract semantic regions in the middle. Pre-trained VGG network [17] is used to extract high-level features and to

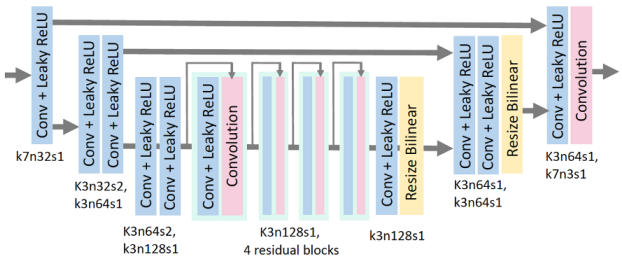


Figure 3. Generator Network.

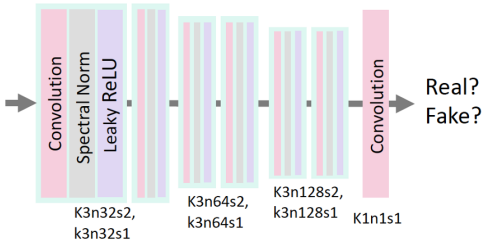


Figure 4. Discriminator Network.

impose spatial constrain on global contents between extracted structure representations and outputs, and also between input photos and outputs. Pre-trained segmentation models [21] are also used as the semantic module to exact different semantic regions. Weight for each component can be adjusted in the loss function, which allows users to control the output style and adapt the model to diverse use cases.

3.1. Network Architecture

We show the architecture of generator network and discriminator network in the above Fig. 3 and Fig. 4.

The generator network is a fully-convolutional U-Net-like [16] network. We use convolution layers with stride 2 for down-sample and bilinear interpolation layers for up-sample to avoid checkerboard artifacts. The network consists of only three kind of layers: convolution, Leaky ReLU (LReLU) [13] and bilinear resize layers. This enables it to be easily embedded in edge devices such as mobile phones.

PatchGAN [9] is adapted in the discriminator network, where the last layer is a convolution layer. Each pixel in the output feature map correspond to a patch in the input image, with the patch size equals to the perceptive field, and is used to judge whether the patch belongs to cartoon images or generated images. The PatchGAN enhances the discriminative ability on details and accelerates training. Spectral normalization [14] is placed after every convolution layer (except the last one) to enforce Lipschitz constrain on the network and stabilize training.

3.2. Loss Function

Our loss fucntion (1) can be formulated as weighted sum of multiple existing losses.

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \sum_{i=1}^{N}(\mathcal{W}_N * \mathcal{L}_{\text{style}N}) \\ &+ \mathcal{W}_{tv} * \mathcal{L}_{tv} \\ &+ \mathcal{W}_{content} * \mathcal{L}_{content} \\ &+ \mathcal{W}_{semantic} * \mathcal{L}_{semantic}\end{aligned} \tag{1}$$

3.2.1 Content Loss

The content loss $L_{content}$ is used to ensure that the generated result and the input image are semantically invariant, and the sparsity of L_1 norm allows for local features to be cartoonized. By calculating the input image and the generated result on a pre-trained VGG feature space, we can get

$$\mathcal{L}_{content} = ||VGG_n(G(\mathrm{I}_p)) - VGG_n(\mathrm{I}_p)|| \tag{2}$$

I_p represents input image and VGG_n represents feature extracted from the n-th layer on the VGG network.

3.2.2 Semantic Loss

The semantic loss $L_{Semantic}$ is used to ensure that the generated result and the input image have same semantic location distribution. By calculating the input image and the generated result on a pre-trained semantic segmentation network [21], we can get

$$\mathcal{L}_{semantic} = ||UNet(G(\mathrm{I}_p)) - UNet_n(\mathrm{I}_p)|| \tag{3}$$

I_p represents input image and $UNet$ represents semantic location information extracted from the a semantic segmentation network, i.e, a pre-trained UNet network.

3.2.3 Style Loss

The style loss $\mathcal{L}_{style}$ includes two parts, surface loss and texture loss.

$$\mathcal{L}_{\text{style}} = \lambda_1 * \mathcal{L}_{\text{surface}} + \lambda_2 * \mathcal{L}_{\text{texture}} \tag{4}$$

Cartoon painting style usually have smooth surfaces similar to cartoon images. To smooth images and meanwhile keep the global semantic structure, a differentiable guided filter [20] is adopted for edge-preserving

filtering. Denoted as $\mathcal{F}_{dgf}$, it takes an image $\boldsymbol{I}$ as input and itself as guide map, returns extracted surface representation $\mathcal{F}_{dgf}(\boldsymbol{I}, \boldsymbol{I})$ with textures and details removed.

A discriminator D_s is introduced to judge whether model outputs and reference cartoon images have similar surfaces, and guide the generator G to learn the information stored in the extracted surface representation. Let $\boldsymbol{I}_p$ denote the input photo and $\boldsymbol{I}_c$ denote the reference cartoon images, we formulate the surface loss as:

$$\begin{aligned}\mathcal{L}_{\text{surface}}\left(G, D_s\right) = & \log D_s(\mathcal{F}_{dgf}(\boldsymbol{I}_c, \boldsymbol{I}_c)) \\ & + \log\left(1 - D_s\left(\mathcal{F}_{dgf}\left(G\left(\boldsymbol{I}_p\right), G\left(\boldsymbol{I}_p\right)\right)\right)\right)\end{aligned} \tag{5}$$

As luminance and color information make it easy to distinguish between cartoon images and real-world photos. We propose a random color shift algorithm $\mathcal{F}_{rcs}$ to extract single-channel texture representation from color images, which retains high-frequency textures and decreases the influence of color and luminance.

$$\mathcal{F}_{rcs}\left(\boldsymbol{I}_{rgb}\right) = (1-\alpha)\left(\beta_1 * \boldsymbol{I}_r + \beta_2 * \boldsymbol{I}_g + \beta_3 * \boldsymbol{I}_b\right) + \alpha * \boldsymbol{Y} \tag{6}$$

In Equation 6, $\boldsymbol{I}_{rgb}$ represents 3-channel RGB color images, $\boldsymbol{I}_r, \boldsymbol{I}_g and \boldsymbol{I}_b$ represent three color channels, and Y represents standard grayscale image converted from RGB color image. We set

$$\alpha = 0.8,\ \beta_1,\ \beta_2,\ \beta_3 \sim \boldsymbol{U}(-1, 1)$$

.

The random color shift can generate random intensity maps with luminance and color information removed. Then a discriminator D_t is introduced to distinguish texture representations extracted from model outputs and cartoons, and guide the generator to learn the clear contours and fine textures stored in the texture representations. Let I_p denotes the input image and I_c denotes the reference cartoon images, we formulate the textural loss as:

$$\begin{aligned}\mathcal{L}_{\text{texture}}\ \left(G, D_t\right) = & \log D_t\left(\mathcal{F}_{rcs}\left(\boldsymbol{I}_c\right)\right) \\ & + \log\left(1 - D_t\left(\mathcal{F}_{rcs}\left(G\left(\boldsymbol{I}_p\right)\right)\right)\right)\end{aligned} \tag{7}$$

3.2.4 Total-variation Loss

The total-variation loss $\mathcal{L}_{tv}$ [1] is used to impose spatial smoothness on generated images. It also reduces high-frequency noises such as salt-and-pepper noise. In Equation 8, H, W, C represent spatial dimensions of images.

$$\mathcal{L}_{tv} = \frac{1}{H * W * C}\left\|\nabla_x\left(G\left(\boldsymbol{I}_p\right)\right) + \nabla_y\left(G\left(\boldsymbol{I}_p\right)\right)\right\| \tag{8}$$

4. Experiments

4.1. Setup

We implement our network details with PyTorch. We use Adam algorithm to optimize the two networks. During training, the learning rate is set to $2 * 10^{-4}$, and the batch size is set to 16. We firstly pre-train the generator for 100 epochs, and then jointly optimize the generator and discriminator based on the framework. After every 100 epochs, we will stop to observe the loss values and the performance of the generated images, then adjust the weight of different loss functions, and stop training until the quality of outputs is satisfactory enough.

4.2. Datasets

For the real-world images, we chose Flickr Image Dataset [8], which contains about 31.8k pictures of real scenes. For the cartoon style images, we collected about 3k key frames from the works of Kyoto animation, P.A. Works, Shinkai Makoto, Hosoda Mamoru, and Miyazaki Hayao.

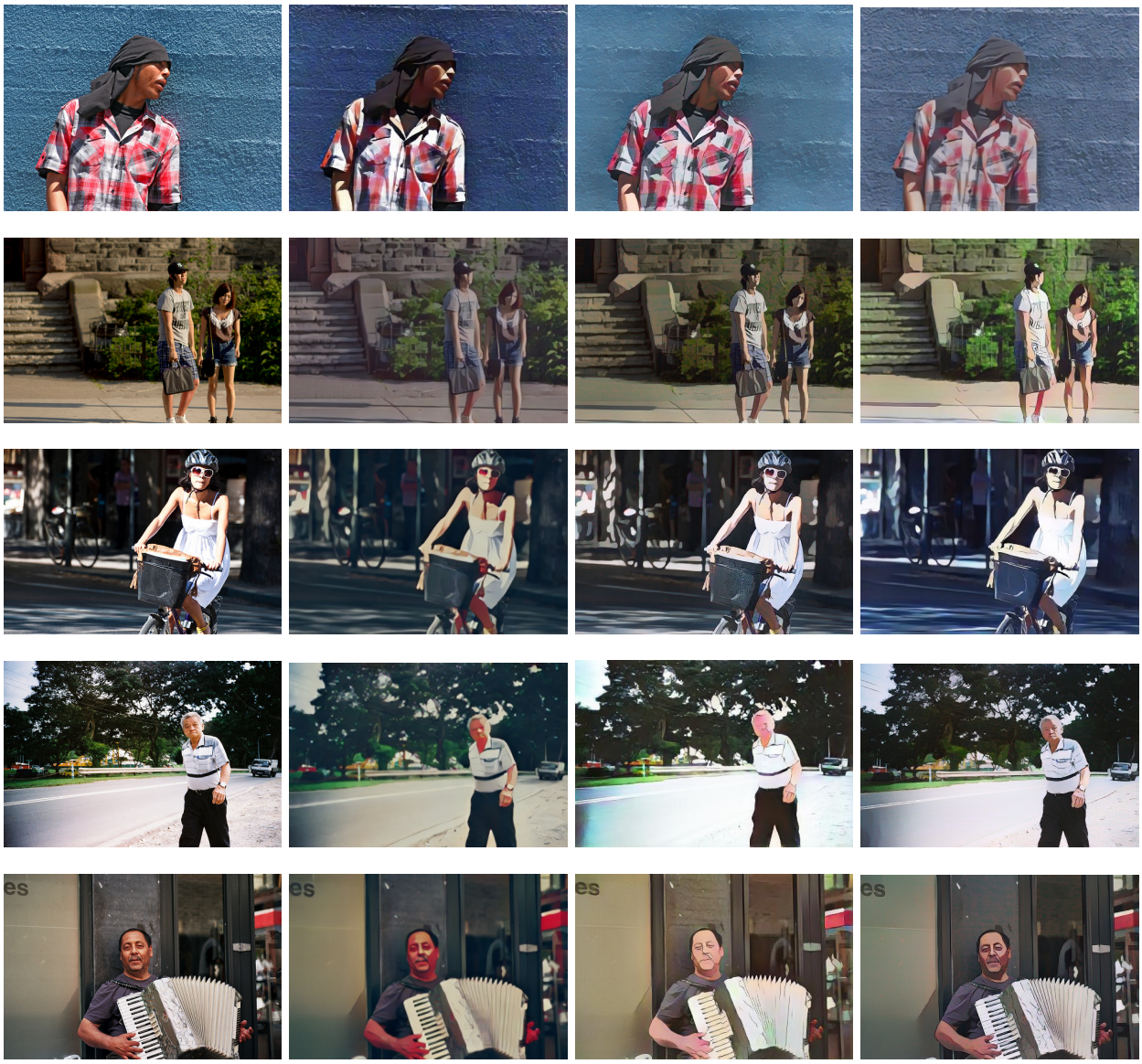
Finally, we crop and resize all images to 256*256 resolution as the training set.

4.3. Qualitative Comparison

Comparisons between our method and previous methods are shown in Figure 5. We can observe that the images cartoonGAN generated are blurry and have some color deviation. Though the white-box framework solves these problems, it still can not generate images that have both satisfactory foreground characters and background contents. For example, the method Wang uses either has a poor background cartoonize effect, or the character's been generated roughly and lose lots of details. Our method not only takes into account the quality of image generation, but also achieves high-quality generation of the respective semantic dimensions of the foreground and background. we generate more natural cartoonized grass and walls and detailed facial features.

4.4. Quantitative Evaluation

Frechet Inception Distance (FID) [7] is wildly-used to quantitatively evaluate the quality of synthesized images. Pre-trained Inception-V3 model [18] is used to



(q) Original Photo (r) CartoonGAN (s) Wang (t) Ours

Figure 5. Results of our method compared with CartoonGAN[4] and Wang[19]

extract high-level features of images and calculate the distance between two image distributions. We use FID to evaluate the performance of previous methods and our method.

As is shown in Table 1, our method generates images with the smallest FID to cartoon image distribution, which proves it generates results most similar to cartoon images. The output of our method also has the smallest FID to real-world photo distribution, indicating that our method loyally preserves image semantic and content information.

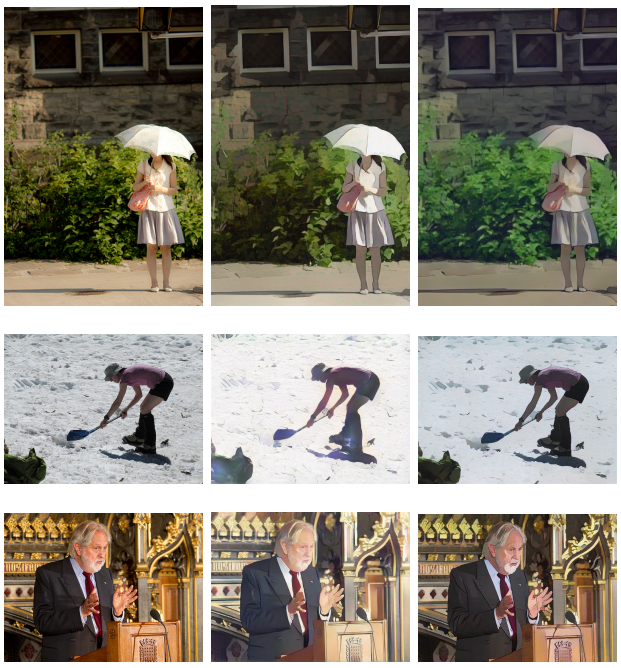
Method	FID with Cartoon	FID with Real
CartoonGAN	143.55	149.88
Wang	124.72	145.34
Ours	122.51	135.82

Table 1. Quantitative Evaluation Results. Ours is better.

4.5. Ablation Study

We show the results of ablation studies in Figure 6. Ablating the semantic module causes less natural cartoonized grass and snowfield, and lose some facial details.So in conclusion, semantic module helps improve

the cartoonizaiton ability of our method.



(g) Original Photo (h) W/O Semantic Module (i) With Semantic Module

Figure 6. Ablation study by removing semantic module.

5. Conclusion

In this paper, we propose a semantic module for style transfer especially image cartoonization framework based on GAN, which can generate high-quality cartoonized style images from real-world photos. Our semantic module is used for describing various semantic regions and matching these regions in different image sets. Extensive quantitative and qualitative experiments have been conducted to validate the performance of our framework. Ablation studies are also conducted to demonstrate the influence of our semantic module.

6. Acknowledgement

We would like to thank Hung-yi Lee for teaching us knowledge of generative adversarial network and Fei-Fei Li, Limin Wang for teaching us knowledge of computer vision. We would also like to thank Zhiming Xu, Ziteng Gao and Yutao Cui for giving us many research advice. We would also like to thank Zhimeng Guo for his connection to U.S.

References

[1] H. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. IEEE Transactions on Image Processing, 14(10):1647–1659, 2005. 5

[2] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Gan dissection: Visualizing and understanding generative adversarial networks, 2018. 3

[3] J. Chen, G. Liu, and X. Chen. Animegan: A novel lightweight gan for photo animation. In K. Li, W. Li, H. Wang, and Y. Liu, editors, Artificial Intelligence Algorithms and Applications, pages 242–256, Singapore, 2020. Springer Singapore. 3

[4] Y. Chen, Y.-K. Lai, and Y.-J. Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9465–9474, 2018. 2, 3, 6

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style, 2015. 1, 2, 3

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014. 2, 3

[7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 5

[8] Hsankesara. Flickr image dataset. https://www.kaggle.com/hsankesara/flickr-image-dataset, 2018. 5

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 3, 4

[10] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, 2016. 3

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc. 1

[12] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer, 2017. 2

[13] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In in ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013. 4

[14] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks, 2018. 4

[15] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016. 3

[16] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 3

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision, 2015. 5

[19] X. Wang and J. Yu. Learning to cartoonize using white-box cartoon representations. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8087–8096, 2020. 3, 6

[20] H. Wu, S. Zheng, J. Zhang, and K. Huang. Fast end-to-end trainable guided filter. In CVPR, 2018. 4

[21] P. Yakubovskiy. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2020. 4

[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020. 3