Cartoon Images Generation based on GAN

Ya Wang
021037801c
2022/07/01

Abstract

Generative Adversarial Network (GAN) has many applications in the image field, among which there are a lot of interesting researches on image generation and image translation (such as cycleGAN, pix2pix, styleGAN, and u-get-it). This project implements an approach for generating high-quality cantonized images from real images based on a generative adversarial network (GAN) framework. In addition, this report presents the comparison and quantitative evaluation of the cartoon images generating performance between different methods.

1. Introduction

Image-to-image translation, one of the most classical generative model applications is a class of vision and graphics problems where the goal is to learn a mapping between input and output images using a set of image pairs. In 2017, Zhu er all proposed a method (Cycle GAN)[8] for learning to transform images from a source domain X to a target domain Y without a paired training set (which can convert images from both domains to each other), which lead to a significant improvement for GAN applications in the unpaired image transformation. However, it's difficult to achieve in relatively large geometric transformations. Nowadays image to image translation based on GAN architecture has achieved great results, among which there is a lot of interesting research on image generation and image translation (such as Cycle GAN, pix2pix, styleGAN, and u-get-it). The application of GAN in the field of cartoons is of great help to cartoon designers and animators, a good algorithm can greatly reduce the work of the artist and there has been a lot of related work in recent years.

Creating drawings of real-world scenes is a time-consuming and labor-intensive task, which requires professional drawing skills. Therefore, technology that can automatically transform real-world photos into high-quality cartoon images can be very helpful for people who want to create cartoons: it can save them a lot of time and help them focus more on their creations.

1.1 Problem statement:

Cartoon imagery is an abstraction of reality, rather than adding textures and borderlines, it needs to be highly simplified from the complex construction of real-world imagery. Transforming pictures of real-life scenes into cartoon-style images, which is valuable and challenging in computer graphics. And the two key challenges for transforming pictures of real-life scenes into cartoon-style images are: 1. Cartoon style has its own unique characteristics, highly simplified and abstract. 2. Cartoon images have distinct edges, smooth colors, and relatively simple textures, which cause great challenges to current methods based on texture descriptor loss functions.

1.2 Research hypothesis

It is hypothesized that high-quality cartoon images could be generated by taking unpaired photos and cartoon images for training.

The goal for my project is to transform real photos into high quality cartoon images through a GAN framework by taking unpaired photos and cartoon images for training. The future work could be video transformation from real scenario to cartoon style.

2. Research contributions

Generative Adversarial Networks (GANs) [14] have received increasing attention from academia and industry since they were designed by Ian Goodfellow et al. Year 2014. And GAN has achieved great success in image generation, among them, GAN has many meaningful and interesting research and applications in the field of cartoon image translation. Such as:

(1) Cartoon coloring, Zhang Luming et al. [2] developed style2paints, which is one of the open-source papers with the best coloring effect.

(2) Generating different animated background images from real landscape images, Yang Chen et al [3] proposed a cartoon GAN for this application. To optimize the effect, they especially strengthened the consideration of edge information in loss to ensure clear edges.

Based on the work of cartoon GAN, there were many works got significant performance. In 2019, Chen et al. proposed a model named as AnimeGAN [7], which improved network structure to get a lightweight generative adversarial mod, and also introduced three loss functions: grayscale style loss, color reconstruction loss, and grayscale adversarial loss, to generate images with better visual effects. In 2020, Wang et al. proposed a method for image cartoon representation [1], in which the structure, surface and texture of the image are processed through three white boxes, and finally an image transformation method that is superior to other methods is obtained. The model is very fast and can be adapted to a variety of painting styles. In 2021, Shu, et al. [2] designed the network structure for the multi-style cartoon image conversion task, in which the generation network of multi-Style CartoonGAN consists of a common encoder and multiple decoders. Two simple yet effective loss functions are proposed in the GAN-based architecture. The new method is much more efficient than existing training methods.

(3) Cartoon face transformation, Wu Ruizheng et al. [3] implemented a new generative model, in which, they extracted the face features according to the position coordinates to optimize the cartoon image transformation. Using unpaired training data, high-quality images of cartoon faces can be generated. Chong et all demonstrate a method named as GANs N'Rose that takes the content code of a face image as input and outputs an anime image with a variety of randomly selected style codes. The core idea of GANs N'Rose is to define content as where things are and style as how things look. This can

be achieved by using the idea of data augmentation (select a set of related data augmentations, under all conditions: style is constant, content is variable).

Although all of these works achieve excellent results, there are still some obvious problems. And the main problems include: 1) a large number of training pictures are essential for the model training, the cartoon images converted from a small amount of data are very close to reality and are not highly generalized 2) a large number of parameters of the GAN network require a large memory and computation capacity.

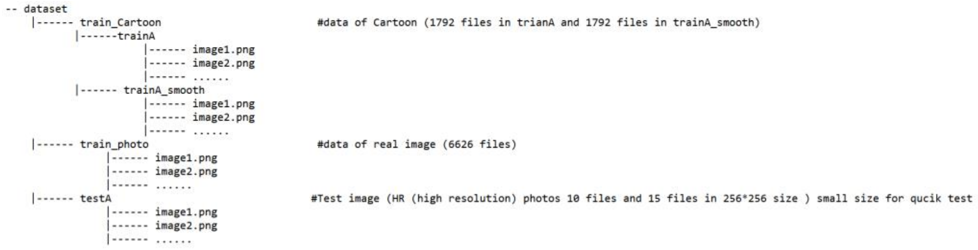
3. Materials and methods

3.1 DATA

The training data is the open source published by Xin Chen [14], which contains real-world photos and cartoon images. All the training images are resized and cropped to 256×256, which show as Fig.1.

Train_photos: 6626 photos are included since 5400 are used for trainning and 1226 are used for test

Train_Cartoon: 1792 cartoon images from the movie “Song of the Wind”.



```
-- dataset
   |------ train_Cartoon                      #data of Cartoon (1792 files in trianA and 1792 files in trainA_smooth)
         |------trainA
                  |------ image1.png
                  |------ image2.png
                  |------ ......
         |------ trainA_smooth
                  |------ image1.png
                  |------ image2.png
                  |------ ......
   |------ train_photo                        #data of real image (6626 files)
            |------ image1.png
            |------ image2.png
            |------ ......
   |------ testA                             #Test image (HR (high resolution) photos 10 files and 15 files in 256*256 size ) small size for qucik test
            |------ image1.png
            |------ image2.png
            |------ ......
```

Figure 1 Dataset structure

3.2 Methods

I studied and tried different image cartoonization methods, among which the Cartoon-GAN proposed by Chen et al [6] seems to work very well and has gained the attention and further research of many researchers. The white-box Cartoon GAN proposed by Wang[1] and the AnimeGAN proposed by chen[7] made a significant improvement for photo cartoonization problem based on Cartoon-Gan. In this project, I implemented the Cartoon-GAN, and trained white box cartoonization and AnimeGAN based on same dataset. And as a comparison, the performance of different algorithm will be evaluated by the FID (Frechet Inception Distance) methods. This section will focus on the model of Cartoon-Gan.

3.2.1 GAN (Generative adversarial network)

A generative adversarial network (GAN) has two parts: generator creates target images from noise or baseline image; discriminator determines whether created image is real or fake. Informally, the generator tries to fool the discriminator, and the discriminator tries to keep from being fooled.

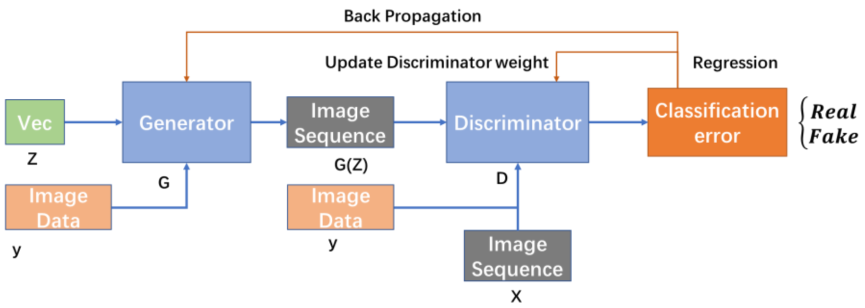


Figure 2 The architecture of GAN frame

Let L be the loss function, G* and D* be the weights of the networks, the objective of GAN architecture is to solve the optimize function:

$$(G^*, D^*) = \arg\min_{G}\max_{D} \mathcal{L}(G, D) \qquad (1)$$

3.2.2 Cartoon-GAN model

Cartoon-GAN consists of two CNN, the architecture of Cartoon-GAN could be seen as Fig.1.

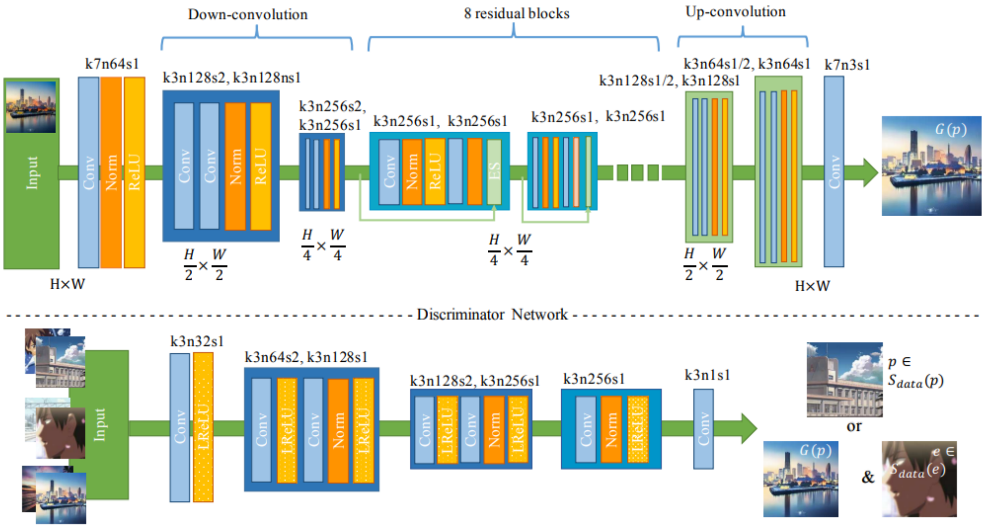


Figure 3 The architecture of the generator and discriminator in the Cartoon-GAN, in which k is the kernel size, n is the number of feature maps and s is the stride in each convolutional layer, 'norm' indicates a normalization layer [1]

As show in Fig.3, we could see that, the network structure of the generator is similar to that of the autoencoder, first down sampling and then up sampling. The structure of the discriminator is similar to the ordinary CNN. The key feature of the Cartoon GAN is the introduction of two loss function: semantic loss (semantic content loss) defined as an $\ell 1$ sparse regularization and an edge-promoting adversarial loss for preserving clear edges [6]. In which, the GAN-based framework with one generator and one discriminator. Therefore, the loss function $L(G, D)$ in Eq. (1) consists of 2 parts: the adversarial loss $L_{adv}(G, D)$ and the content loss $L_{con}(G, D)$:

$$\mathcal{L}(G,D) = \mathcal{L}_{adv}(G,D) + \omega\mathcal{L}_{con}(G,D), \qquad (2)$$

The loss function Eq. (2) is used to train the discriminator and the generator. In the adversarial part, discriminator D tries to classify whether the images come from the real target anime domain or the output of generator. During the generator training, the generator G was trained to transform anime pictures from the photos from real world scenes, where the discriminator classifies the generated image as fake. By jointly optimizing with features learned from these two cartoon representations (loss), we can get a model for high quality transforming cartoon images.

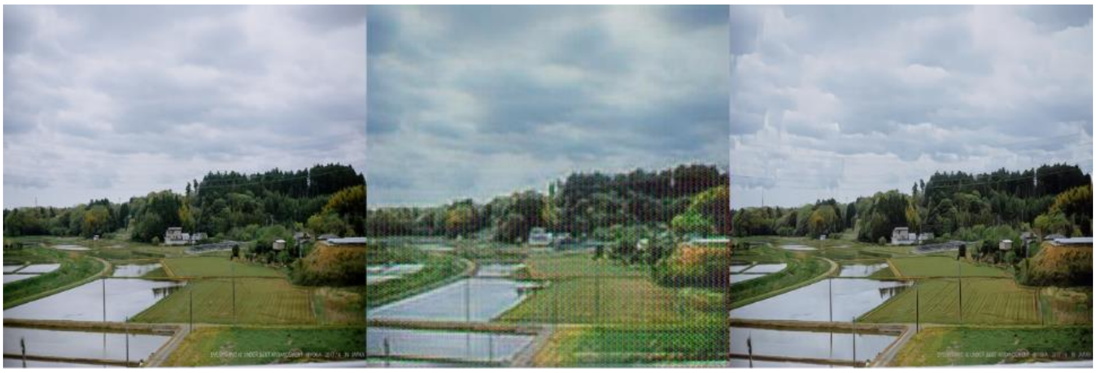
3.3 Pre-train Model

In order to make the output images from the generator have the similar content of the original photos, the pre-trained VGG19 [13] is used as the perceptual network to obtain the loss of the perceptual feature of generated images and original photos.

4. Experiments and Results

4.1 Experiment and Results

Initially, I set ω equal to 10 as the value given in the paper [6]. After running 100 and 500 epochs (100 batch in each epoch, batch size=16), the content preserved good, but the generated images didn’t show cartoon styles. I tried to solve that by increasing the training data, but the output image didn’t improve. Since the aim of the project concentrates on cartoon style. As the adversarial loss is more important for the cartoon-effect (as showing in the Eq. (2)), I set a much lower ω =0.1, and ω =2, to balance the values of the adversarial loss $L_{adv}(G,D)$ and the content loss $L_{con}(G,D)$.



Real Photo

ω =10, Epoch =100,Batch = 100

ω =10,Epoch=500,Batch =100

Figure 4 The output Cartoon images with initial setting w=10

The final setting for Cartoon GAN is

--500 epochs --500 batch in each epoch --batch size=16

--pretrain_epochs 1 --content_lambda 2 --pretrain_learning_rate 2e-4

--g_adv_lambda 8. --generator_lr 8e-5 --discriminator_lr 3e-5

--style_lambda 25.

The parameter settings of white-box GAN and Anime GAN are the same as those in the open-source paper.

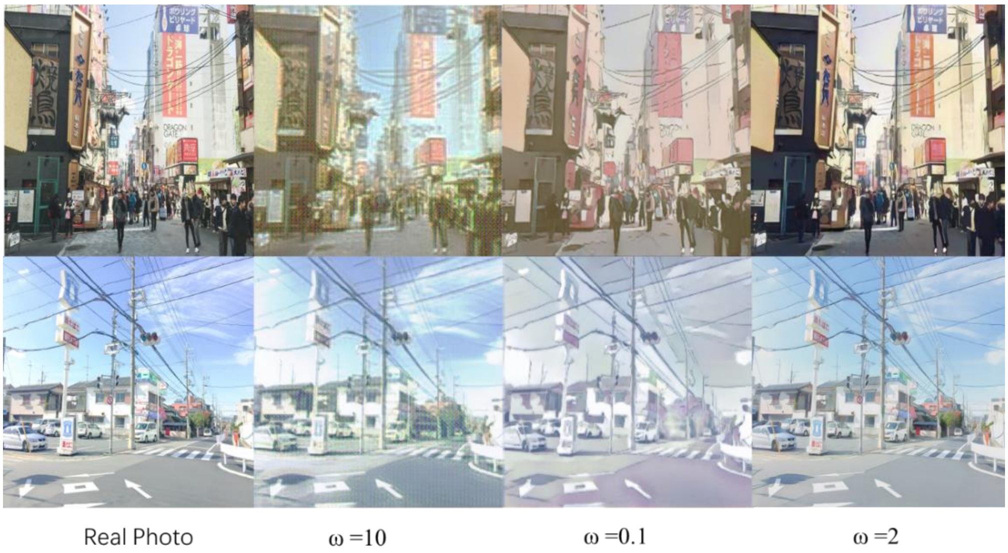


Figure 5 Cartoon images output of Cartoon GAN for different w

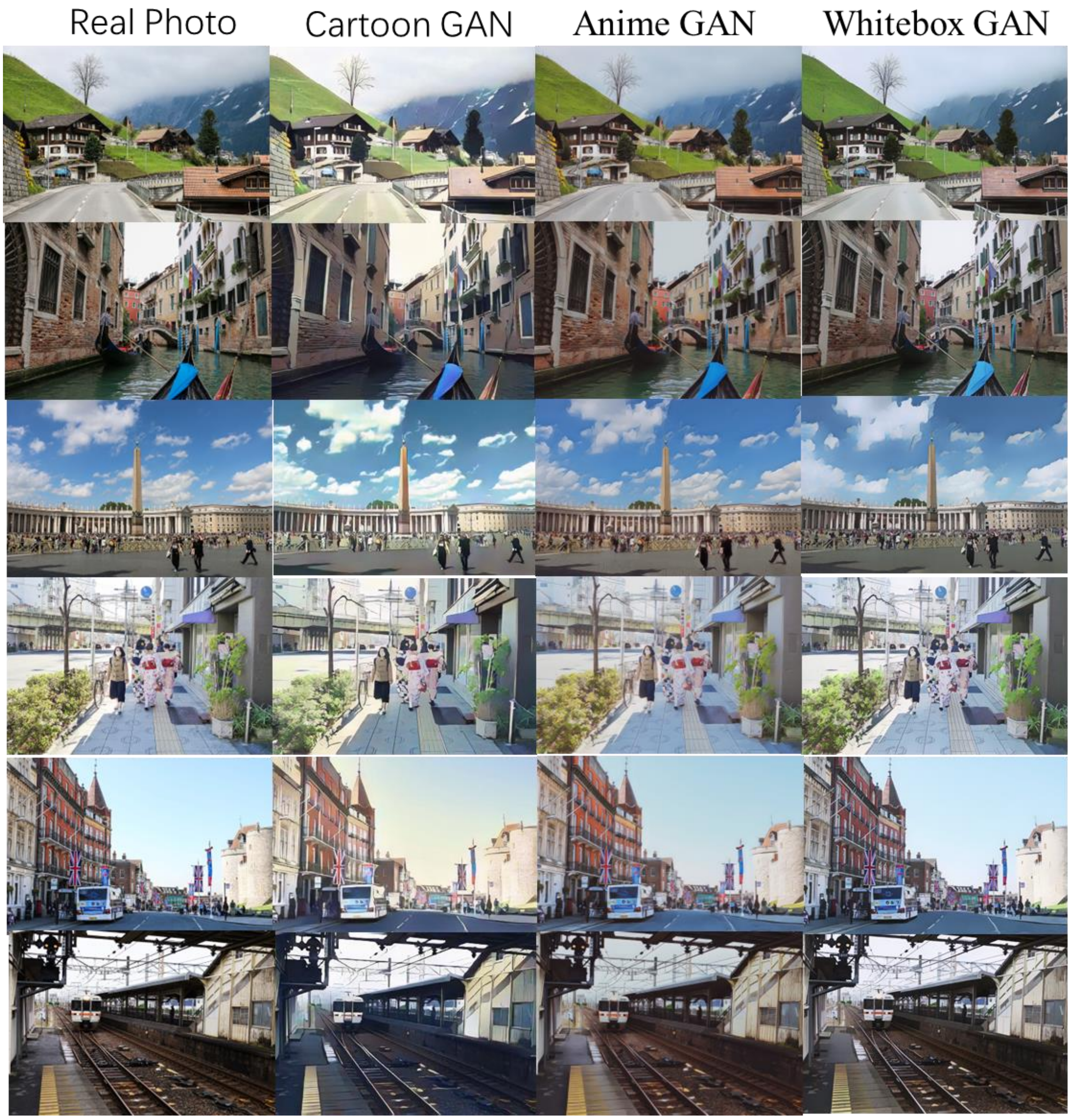


Figure 6 Comparison of Cartoonization performance by different methods

4.2 Comparison between different methods

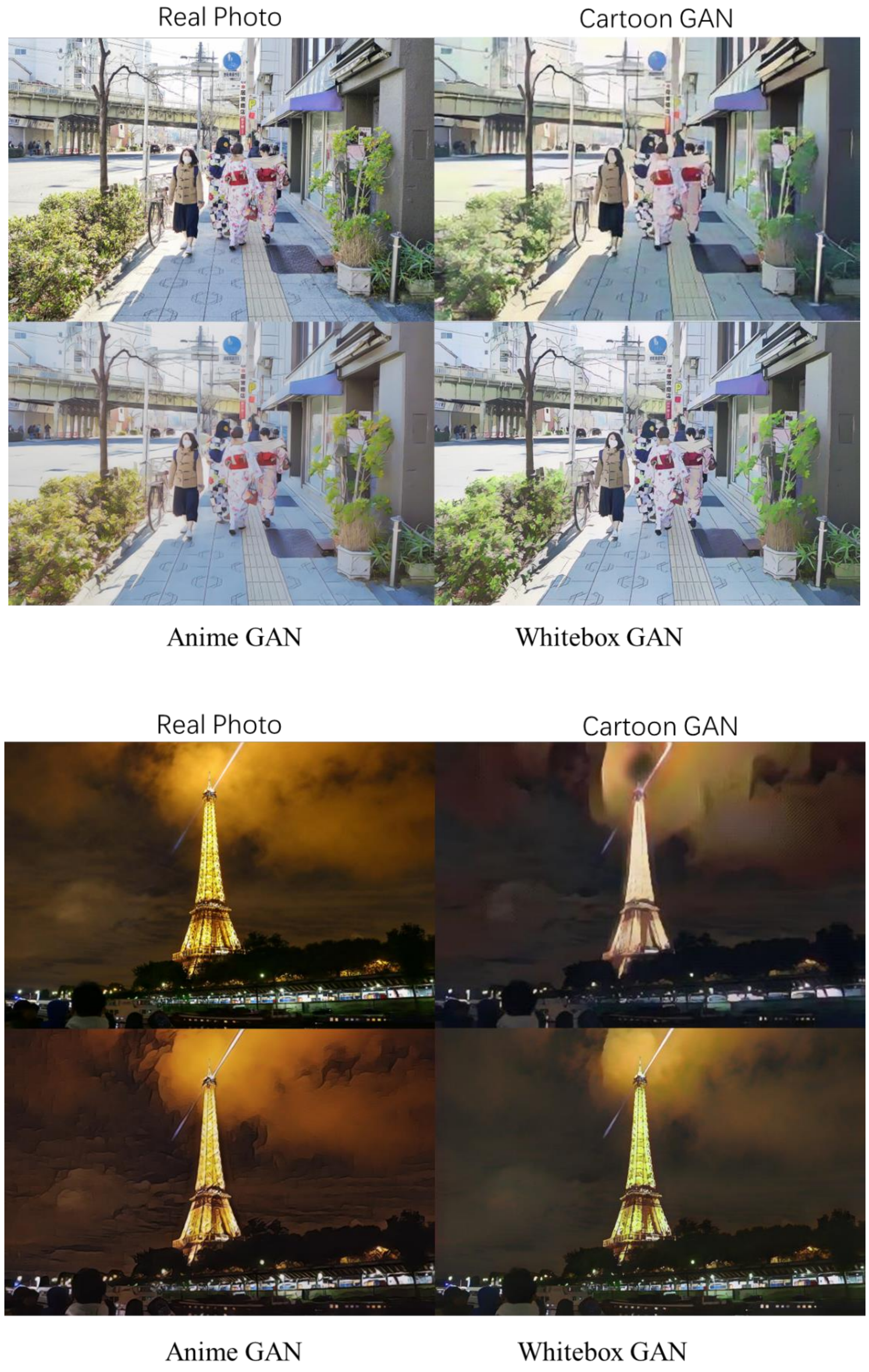


Figure 7 Comparison of transformation performance in details

Comparisons between diffent methods are shown in Figure 7. As we can see, the cartoon images transformed by white-box framework get a better results:

1. Structure: white box frame generates a cleaner contours and borderlines, such as the human face, plants and clouds. The cartoon images generating from CartoonGAN has a good image abstraction effect but also cause noisy and messy contours.

2. Color: white box frame keeps better color harmonious. While Anime GAN generates darkened images and Cartoon GAN weakens the contrast between light and dark of the photo and causes oversmoothed color.

3. Texture: white box frame reduces artifacts while preserves fine details, such as the logo and text in pictures, while the other two methods cause some over-smoothed features.

4.3 Quantitative Evaluation

All of the three methods need a very long training time (the training time for Cartoon GAN>20h for 2 GPU in HPC for 500 epoch and 500batch) and large training data set. But the model of the white-box GAN is very fast and could also be adapted to a variety of painting styles (by different training data). The details of the method performance show in the table1.

Table 1 Comparison of the performance of the different methods (HR means 720*1280 resolutions)

Methods	Cartoon GAN	Anime GAN	Whitebox GAN
Parameter(M)	**12.2**	**3.9**	**1.48**
Inference time (HR, GPU) (ms)	**51 /image**	**43 /image**	**17.23/image**
Model size(M)	**46.74**	**15.09**	**32.59**

Frechet Inception Distance (FID) [15] is wildly used to quantitatively evaluate the quality of synthesized images. The FID distance calculates the real samples and generates the distance between the samples in the feature space. First use the Inception network to extract features, then use the Gaussian model to model the feature space, and then solve the distance between the two features. Lower FID means higher image quality and diversity.

Table 2 Result of User study, higher score means better quality

Methods	Cartoon GAN	Anime GAN	Whitebox GAN
Cartoon quality, mean	**2.940**	**3.220**	**4.017**
Cartoon quality, std	**1.047**	**1.542**	**0.962**

5. Conclusion

Within this report, I implemented the cartoon GAN with TensorFlow, which can generate high-quality cartoonized images from real-world photos, and compare the transformation performance by different methods in paper [1], [6] and [7]. After comparison of the generated cartoon images and the performance and efficiency of 3 different methods, we can conclude that white-box GAN [1] get a better performance. Nevertheless, there are still some limitations for the 3 methods: the algorithms require a lot of data training, and the cartoon images converted from a small amount of data are very close to reality and are not highly generalized. The future work could be: 1) Enhance the brightness contrast of the converted image, classify the converted image according to color, and further summarize the color information of the image to enhance the color contrast of the picture; 2) Conversion from cartoon images to real scene photos.

Reference

[1] Wang, Xinrui, and Jinze Yu. "Learning to cartoonize using white-box cartoon representations." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[2] Shu, Yezhi, et al. "Gan-based multi-style photo cartoonization." *IEEE Transactions on Visualization and Computer Graphics* (2021).

[3] Chong, Min Jin, and David Forsyth. "GANs N'Roses: Stable, Controllable, Diverse Image to Image Translation (works for videos too!)." *arXiv preprint arXiv:2106.06561* (2021).

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," arXiv preprint arXiv:1611.07004, 2016.Image-to-Image Translation with Conditional Adversarial Networks

[5] Wu, Ruizheng, et al. "Landmark assisted cyclegan for cartoon face generation." *arXiv preprint arXiv:1907.01424* (2019).

[6] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9465–9474, 2018.

[7] Chen, Jie, Gang Liu, and Xin Chen. "AnimeGAN: a novel lightweight GAN for photo animation." International Symposium on Intelligence Computation and Applications. Springer, Singapore, 2019.

[8] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *Proceedings of the IEEE international conference on computer vision*. 2017.

[9] Z. Yi, H. Zhang, P. T. Gong et al., "Dualgan: Unsupervised dual learning for image-to-image translation," arXiv preprint arXiv:1704.02510, 2017.DualGAN: Unsupervised Dual Learning for Image-to-Image Translation

[10] Chen, Yang, Yu-Kun Lai, and Yong-Jin Liu. "Cartoongan: Generative adversarial networks for photo cartoonization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[11] Felzenszwalb, Pedro F., and Daniel P. Huttenlocher. "Efficient graph-based image segmentation." *International journal of computer vision* 59.2 (2004): 167-181.

[12] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 172–189, 2018.

[13] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[14] https://github.com/TachibanaYoshino/AnimeGAN/tree/master/dataset

[15] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). Generative Adversarial Nets. Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.