ISSN: 2664-9640

Image Style Migration Based on Improved CycleGAN

Yuqing Zhao *, Guangyuan Fu, Hongqiao Wang and Shaolei Zhang

Xi'an research institute of high-tech, Shaanxi,710025, China.

* Corresponding Author

Abstract

The style migration of images is still of great value in computer vision and graphics, so we propose an improved CycleGAN method to achieve this problem. In order to better maintain the content of the original image and the style of the reference image, content loss and style loss are introduced. Our method is experimented in two areas of Chinese calligraphy and oil painting to verify the versatility and effectiveness of the method. Finally, two methods were used to quantitatively analyze the experimental results, which proved that our method is superior to the modified CycleGAN and neural style migration methods.

Keywords

GANs; style migration.

1. Introduction

Artistic creation is a creative activity in which artists use their own artistic experience, concept and aesthetic experience to transform specific artistic contents and forms into works of art and artistic texts through certain artistic media and language. However, a professional and valuable artistic creation requires a certain artistic foundation of the artist and requires creation time. For an artist who has passed away, it has become impossible to obtain his art work. In recent years, deep learning has been successfully used for image-to-image translation[1-4] as a representative technique learned from examples (especially from big data). However, synthesizing a given photograph into a realistic work of art[5-7] is a challenging problem, and there is much room for improvement with current technology. The Generative adversarial nets (GAN)[8] has shown powerful image generation capabilities. Image-to-image translation is one of the main applications of recently generated models and has shown promising results.

Although the achievements are gratifying, it is well known that GAN faces two major difficulties. First of all, GAN is not easy to train, and the training process is unstable. It is very sensitive to every aspect of its setting (from super parameters to model architecture). Secondly, there are few quantitative evaluation methods for synthetic images at present. In this work, we focus on the above aspects of the problem.

Our main contributions are:

We use the Wasserstein distance to calculate the cyclic consistency loss in CycleGAN. Experiments prove that the content consistency is guaranteed to be more effective.

Content loss and style loss are added to the loss function, which better guarantees the preservation of the original image content and the inheritance of the reference image style.

Use the perceptual hashing algorithm and the cosine similarity method to quantitatively compare the proposed method with other the state-of-the-art methods.

ISSN: 2664-9640

2. Related work

2.1. Generative Adversarial Nets (GAN)

The GAN method consists of a generator G and a discriminator D, which compete in a two-player minimax game: the generator attempts to fool the discriminator by generating realistic images, while the discriminator attempts to distinguish the composite image from the real data. Formally, D and G are two players of minimum-maximum problem to find Nash equilibrium:

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

Goodfellow et al. (2014) demonstrated that this minimax game has global optimal values when $p_g = p_{data}$, and p_g converges to p_{data} under mild conditions (eg, G and D have sufficient capacity).

2.2. Unpaired Image Translation

Pix2Pix[9] can well handle the image transformation of the unpaired data set, but in many cases, there is almost no absolutely paired data set, which is also very difficult to collect. However, the amount of unpaired data in the two fields is still very large. If the image transformation can be realized through the unpaired data, the work becomes very meaningful. In 2017, CycleGAN[10] and DiscoGAN[11] both proposed an image conversion scheme for solving unpaired data sets. The schematic diagram of CycleGAN's structural framework is shown in figure 1, which is essentially a ring network formed by two GAN. The final mean square error loss is expressed as:

$$L_G = L_{G_{XY}} + L_{G_{YX}} = L_{GAN_Y} + L_{CONST_X} + L_{GAN_X} + L_{CONST_Y} \quad (2)$$

$$L_D = L_{D_X} + L_{D_Y} \quad (3)$$

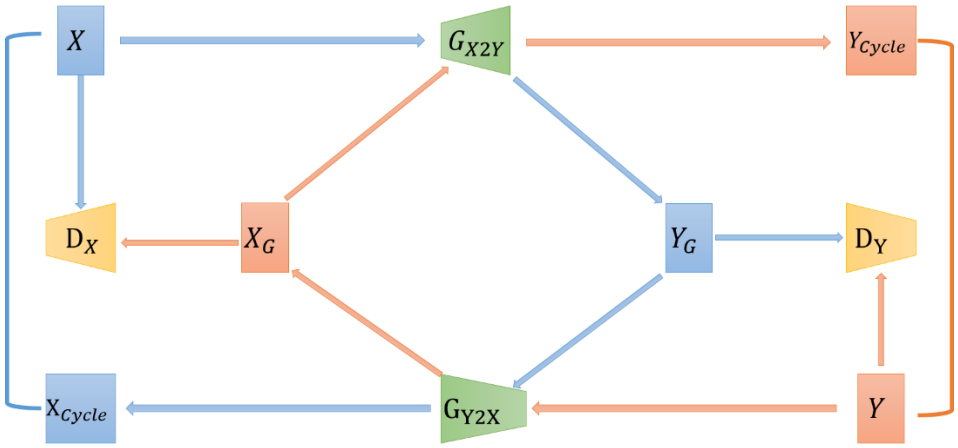

Figure 1: Model structure of CycleGAN

2.3. StackGAN

StackGAN[12] uses two GANs. The first GAN, referred to as the first stage, generates images from subtitles in a manner similar to GAN-CLS[13] at a low resolution of 64×64. The second GAN, called Stage II, has a generator, which takes the images generated by the Stage I generator as input to generate 256×256 images with higher resolution, finer details and better text image matching.

2.4. WassersteinGAN

Researchers found that there are many problems in the training process of GAN, among which the biggest problem is the instability of training. Intuitively, we should first train the discriminator, but in fact, the better the discriminator, the harder it is for the generator to

ISSN: 2664-9640

optimize. If starting from the discriminator loss defined by the original GAN, the form of the optimal discriminator is obtained. In the case of an optimal discriminator, minimizing the generator loss defined by the original GAN is equivalent to minimizing the JS divergence between the real distributions and the generated distribution. However, JS divergence is a constant when the two distributions have no overlaps or the overlaps can be ignored, which makes the gradient of gradient descent method disappear and the generator cannot continue to optimize. From this point, WGAN[14] used the Wasserstein distance instead of the JS distance, and completed the problem of stable training and process indicators. The author uses the existing theorem to transform the Wasserstein distance into the following form:

$$W(P_r, P_g) = \frac{1}{K} \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim P_r[f(x)]} - \mathbb{E}_{x \sim P_g[f(x)]} \tag{4}$$

Where K is the Lipschitz constant of the function f.

3. Method

We describe the network architecture of our method (herein referred to as CycleWGAN) in Section 3.1 and describe the loss functions used in this section in Section 3.2.

3.1. CycleWGAN Network Structure Framework

In terms of network structure design, this work refers to the network research of CycleGAN and StackGAN. Figure 2 shows the network structure of the generator, which is composed of four parts: coding layers, conversion layers, decoding layers and residual layers.

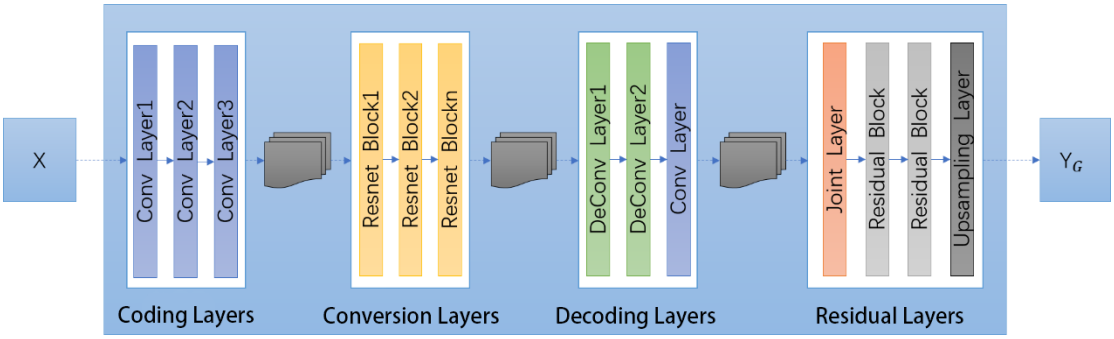

Figure 2: Generator network structure of CycleWGAN

3.2. Loss Function

The conversion between unpaired image domains is not as simple as two GANs. If the generator GX2Y that converts X to Y wants the discriminator DY to judge the generated image as true, then GX2Y had better not extract any information related to X, but independently learn from Y and generate data. At this point, although we got a realistic Y-domain image, this image has nothing to do with the X-domain. Therefore, the concept of Cycle-consistency Loss was introduced in CycleGAN. In Figure 1, we combine the two sets of generated confrontation networks. Image X generates YG through GX2Y, and YG is restored to Xcycle through GY2X. Obviously, the more consistent X and Xcycle are, the closer our entire network is to our desired goal. Therefore, the distance between X and Xcycle is called Cycle-consistency Loss:

$$\begin{aligned} L_{CONST} = (G_{X2Y}, G_{Y2X}) = \mathbb{E}_{x \sim p_{data}}(x)[\|G_{Y2X}(G_{X2Y}(x) - x)\|_1] \\ + \mathbb{E}_{y \sim p_{data}}(y)[\|G_{X2Y}(G_{Y2X}(y) - y)\|_1] \end{aligned} \tag{5}$$

In view of the many advantages of Wasserstein distance discussed in section 2.4, in this work we will use Wasserstein distance to calculate the cyclo-consistency Loss:

ISSN: 2664-9640

$$L_{CONST} = (G_{X2Y}, G_{Y2X}) = W(P_{rx}, P_{gx}) + W(P_{ry}, P_{gy}) \tag{6}$$

Where W(P_rx,P_gx) and W(P_ry,P_gy) are:

$$W(P_{rx}, P_{gx}) = \frac{1}{K_1} \sup_{\|f_1\|_L \le K_1} (\mathbb{E}_{x \sim P_r[f_1(x)]} - \mathbb{E}_{x \sim P_g[f_1(x)]}) \tag{7}$$

$$W(P_{ry}, P_{gy}) = \frac{1}{K_2} \sup_{\|f_2\|_L \le K_2} (\mathbb{E}_{y \sim P_r[f_2(y)]} - \mathbb{E}_{y \sim P_g[f_2(y)]}) \tag{8}$$

On the other hand, the problem we are trying to solve falls into the category of style transfer. Neural style transfer was proposed by Leon Gatys et.al in 2015[15]. We also added the loss function of the content and style in style migration:

$$W(P_{rx}, P_{gx}) = \frac{1}{K_1} \sup_{\|f_1\|_L \le K_1} (\mathbb{E}_{x \sim P_r[f_1(x)]} - \mathbb{E}_{x \sim P_g[f_1(x)]}) \tag{9}$$

$$W(P_{ry}, P_{gy}) = \frac{1}{K_2} \sup_{\|f_2\|_L \le K_2} (\mathbb{E}_{y \sim P_r[f_2(y)]} - \mathbb{E}_{y \sim P_g[f_2(y)]}) \tag{10}$$

Where distance is a norm function; Content is a function that takes an image and computes its representation; Style is a function that inputs an image and computes the representation of its style. Minimizing these two loss functions causes the style of the generated image to be close to the reference image, and the content of the generated image is closer to the original image.

4. Experiments

We performed experimental validation in two application domains: Chinese character fonts and landscape images. In the first field, we try to learn Yan style font generated. In the field of landscape images, we transform a set of landscape photos into oil painting styles. In both domains, the original domain and the target domain have both common and visible differences.

4.1. Chinese Character Font: From SimSum to the Yan Style:

The SimSum is a kind of Chinese character printing font, which imitates the font carved in the book of song dynasty. The strokes are of uniform thickness, which is quite different from the handwriting of the brush. Yan style was created by Yan Zhenqing, a calligrapher in Tang Dynasty. Yan Zhenqing and Liu Gongquan are called "Liu Yan", and there is a saying of "Yan Jin Liu Gu".

The training data included 5,000 SimSum characters from the computer font library and 3,694 Yan characters (including duplicates) from the Multi-pagoda Stele and Yan Qin Ritual Stele. All words are pre-processed to 256 x 256. After 200 rounds of training with our method, the test was done with English letters, and the results shown in Figure 3 were obtained, all of which kept the style of Yan.Acknowledgements

Natural Science Foundation.

ISSN: 2664-9640

Figure 3: Part of the generated characters

4.2. Landscape Image: From Photo to Oil Painting

In another set of experiments, we compared the method in this paper with that in CycleGAN. As a result, it can be seen that the image generated by CycleGAN (as shown in figure 4(b)) is simply pixelated with distortion in color. Our CycleWGAN performs well in style capture and content retention due to the inclusion of Wasserstein distance-based loop consistency loss, content loss and style loss, as shown in figure 4 (b).

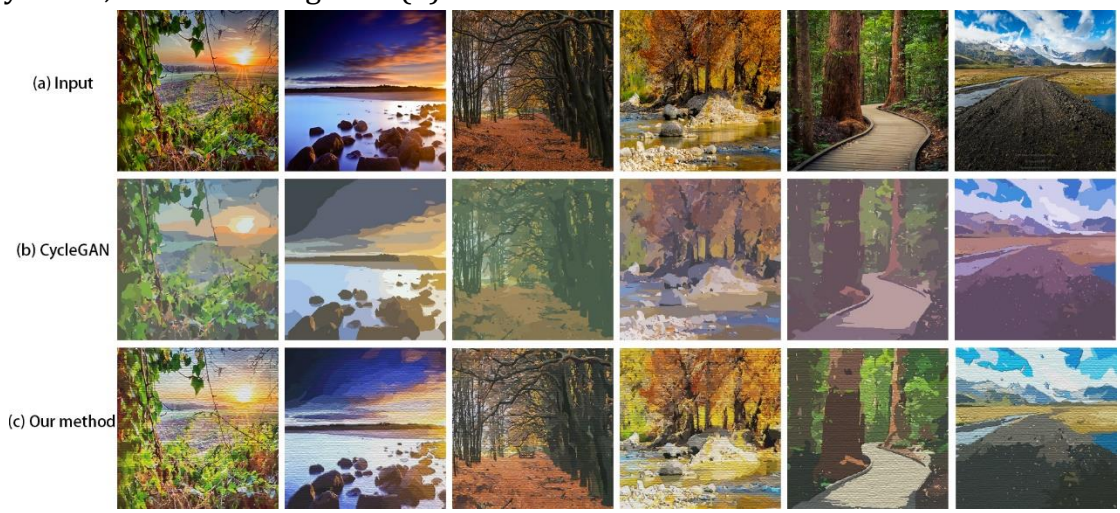

Figure 4: (a) Input image (b) Image generated by CycleGAN (c) Image generated by our method

4.3. Quantitative Evaluation of Experimental Results

The quality evaluation of the composite image after the style migration is mainly reflected in the visual effect of the image. There are relatively few quantitative indicators that can be referred to in the quality evaluation of synthetic images. In this work, perceptive hashing algorithm[16] and cosine similarity method[17] are respectively adopted to evaluate the generated images. Figure 5 shows the calculation of hashing values of 100 oil painting test graphs and 647 oil painting training graphs respectively, and then the hamming distance[18] between each test graph and training graph is calculated and the mean value is calculated. The smaller the distance, the higher the stylistic similarity with the reference atlas. The statistical results show that 82% of the results generated by the method of this work are better than those of CycleGAN.

ISSN: 2664-9640

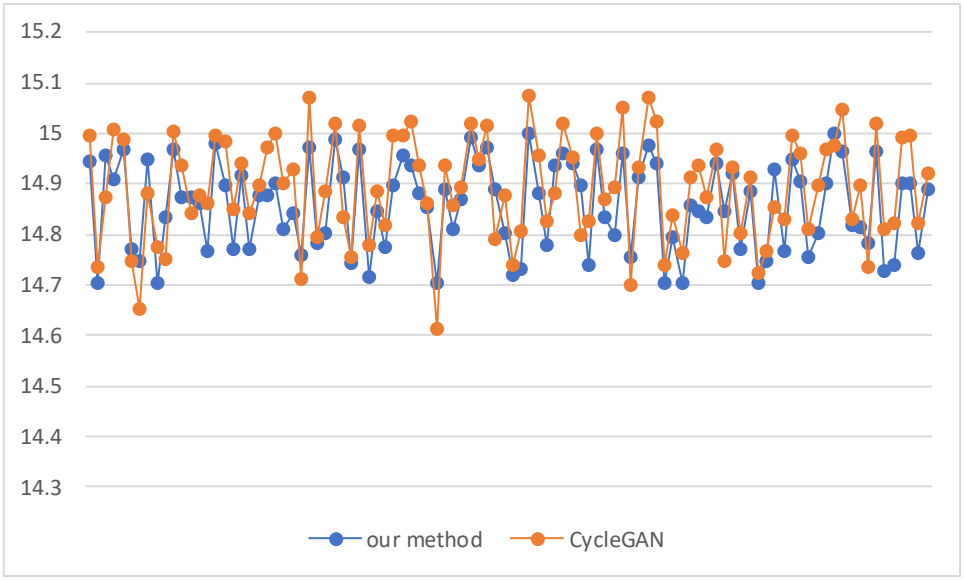

Figure 5: Comparison between CycleWGAN and CycleGAN based on perceptive hash algorithm

Figure 6 shows the results of comparing three algorithms of CycleWGAN, CycleGAN and neural style transfer by cosine similarity method. The closer the calculated value is to 1, the higher the stylistic similarity of the two images is. In the experimental results, our method exceeded 67% of the CycleGAN generation images and 93% of the neural style migration results.

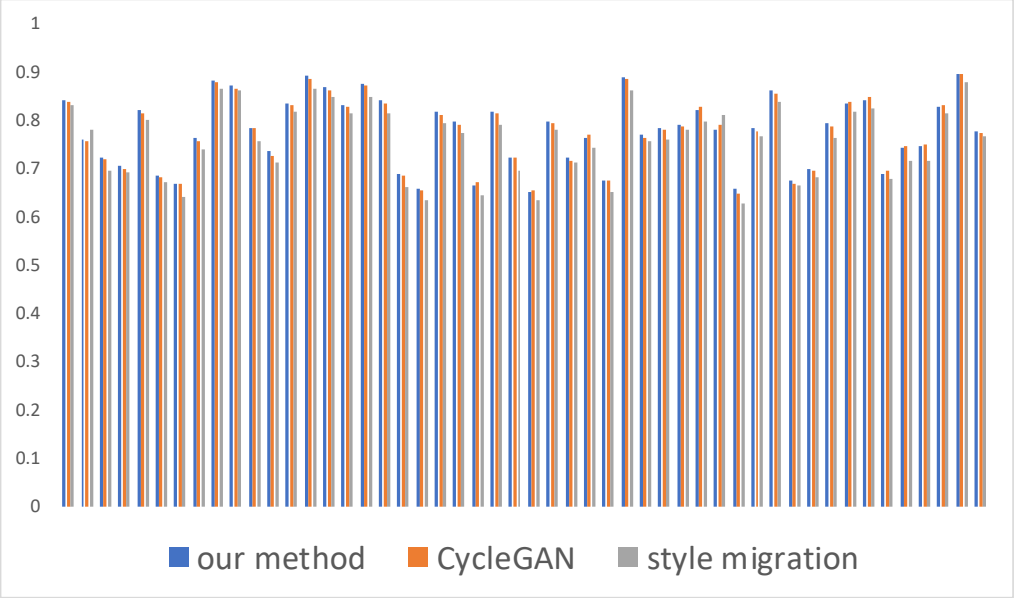

Figure 6: Cosine similarity method comparison CycleWGAN, CycleGAN and neural style migration algorithm

5. Conclusion

The CycleWGAN proposed in this work combines CycleGAN and WGAN to achieve high-quality style transfer of artistic works. In order to achieve higher resolution artwork, we propose: (1) Wasserstein distance is used to calculate the loss of cycle consistency in CycleGAN, and the experiment proves that it provides more effective guarantee for maintaining content consistency. (2) The content loss and style loss commonly used in the style transfer algorithm are added into the loss function to achieve the purpose of changing from the original image to the target image style by continuously minimizing these two losses. (3) The perceptual hash algorithm and cosine similarity method are introduced to try to quantitatively evaluate the

ISSN: 2664-9640

effect of style transfer. In the following work, we try to propose a new evaluation function to quantify the effect of style transfer. On the other hand, we plan to look at ways to generate style transfer video.

References

[1] equation reference goes hereBruna J, Sprechmann P, Lecun Y J a P A. Super-resolution with deep convolutional sufficient statistics[J], 2015.

[2] Hinton G E, Salakhutdinov R R J S. Reducing the dimensionality of data with neural networks[J], 2006, 313(5786): 504-507.

[3] Huang X, Liu M-Y, Belongie S, et al. Multimodal unsupervised image-to-image translation[C]. Proceedings of the European Conference on Computer Vision (ECCV), 2018: 172-189.

[4] Yi Z, Zhang H, Tan P, et al. Dualgan: Unsupervised dual learning for image-to-image translation[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 2849-2857.

[5] Chen Y, Lai Y-K, Liu Y-J. Cartoongan: Generative adversarial networks for photo cartoonization[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 9465-9474.

[6] Li W, Xiong W, Liao H, et al. CariGAN: Caricature Generation through Weakly Paired Adversarial Learning[J], 2018.

[7] Tian Y. zi2zi: Master chinese calligraphy with conditional adversarial networks, 2017.

[8] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]. Advances in neural information processing systems, 2014: 2672-2680.

[9] Isola P, Zhu J-Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017: 1125-1134.

[10] Zhu J-Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 2223-2232.

[11] Kim T, Cha M, Kim H, et al. Learning to discover cross-domain relations with generative adversarial networks[C]. Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017: 1857-1865.

[12] Zhang H, Xu T, Li H, et al. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 5907-5915.

[13] Reed S, Akata Z, Yan X, et al. Generative adversarial text to image synthesis[J], 2016.

[14] Arjovsky M, Chintala S, Bottou L J a P A. Wasserstein gan[J], 2017.

[15] Gatys L A, Ecker A S, Bethge M J a P A. A neural algorithm of artistic style[J], 2015.

[16] Wang J, Zhang T, Sebe N, et al. A survey on learning to hash[J], 2018, 40(4): 769-790.

[17] Nguyen H V, Bai L. Cosine Similarity Metric Learning for Face Verification[C]. Asian Conference on Computer Vision, 2010.

[18] Hamming R W J T B S T J. Error detecting and error correcting codes[J], 1950, 29(2): 147-160.