ISSN: 2688-8645

Multi-style Cartoon Style Migration Study

Song You, Guojun Lin

School of Automation and Information Engineering, Sichuan University of Science and Engineering, Zigong 643000, China

Abstract

Cartoon pictures are a kind of art form that we often contact in our daily life, and generating different styles of cartoon pictures from a given real-life photo is a great promotion for the development of art. Aiming at the current existing methods in generating cartoon images appear in the degree of stylization is insufficient, the generalization ability of the situation is poor, this paper proposes a kind of improvement of the generation of adversarial network: in the generator module to join the adaptive normalization way to improve the generalization ability of the model; at the same time the introduction of the auxiliary discriminator to help the generator style to better present the different styles; in the data processing of this paper on the cartoon image to do guided filtering. The experimental results show that the cartoon image generated by the method of this paper is of higher quality and better effect.

Keywords

Cartoon Animation; Generative Adversarial Network; Style Migration; Adaptive Normalization.

1. Introduction

Animation is a very common form of artistic expression in life, which is loved by people. It has a large number of applications in children's education, video games, film and television, and other industries. So far, animation is mainly created by hand. This not only requires professional drawing skills, but also is a time-consuming job. Many scenes in current animation movies are based on the real world. Therefore, there is a need for on-set footage and authors who have the ability to convert real scenes into anime images. Therefore, the technology to automatically convert pictures into anime images is quite relevant. As far as the movie industry is concerned, it has the potential to free up the creators' time and allow them to focus more on the content of anime movies.

The DCGAN model was trained on a dataset of 143,000 anime character faces to generate new anime faces [1]. However, we can still see some unclean results, partly caused by outliers in the input process.CartoonGAN proposes image translation using unpaired training data, which greatly reduces the amount of work required for data preprocessing [2].The project features a simple patch-level discriminator, edge facilitation against loss, and advanced features for content loss in VGG networks L1 sparse regularization of the graph. Nonetheless, the generalization ability of this black-box model is relatively weak.CycleGAN was one of the first and most enlightening studies that introduced us to cycle-consistent adversarial networks with cycle-consistent loss and full-cycle transformations [3]. Their approach has been expanded and improved in many subsequent studies. A big problem with period-consistent adversarial networks is that they require a large amount of input data.White-boxCartoon model presents a GAN-based white-box controllable image cartoonization framework that can generate high-quality cartoonized images from real-world photographs [4]. The image is decomposed into three cartoon representations: surface representation, structure representation and texture representation. The three representations are extracted for network training using the

ISSN: 2688-8645

corresponding image processing module, and the output style can be controlled by adjusting the weight of each representation in the loss function. This thesis focuses on how to translate real-life footage into anime style. Furthermore, it deals with new algorithms and manually collected datasets to improve the generated animation. Our proposal is an improved generative adversarial model that proposes to incorporate adaptive instance normalization into the generator module along with an auxiliary discriminator, and also proposes guided filtering of cartoon images in terms of data preprocessing. Higher quality and better cartoon images can be generated.

2. Methodology of this Paper

In this paper, we propose adaptive normalized cartoon model for photo cartoonization. The model follows the standard GAN model [5-6].The GAN network model is shown in Fig. 1.The model consists of two CNNs. One is the generative network G, which produces outputs for spoofing the discriminator. The other is the discriminator network D, which is used to determine whether the output image is real or generated by the generator. In order to learn many different cartoon styles and improve model generalization, we decouple the generator G into an encoder and decoder for real-world photos, and for the residual module we introduce an adaptive normalization module. In order to produce significant differences between the output styles, we further introduce an auxiliary classifier and a style loss.

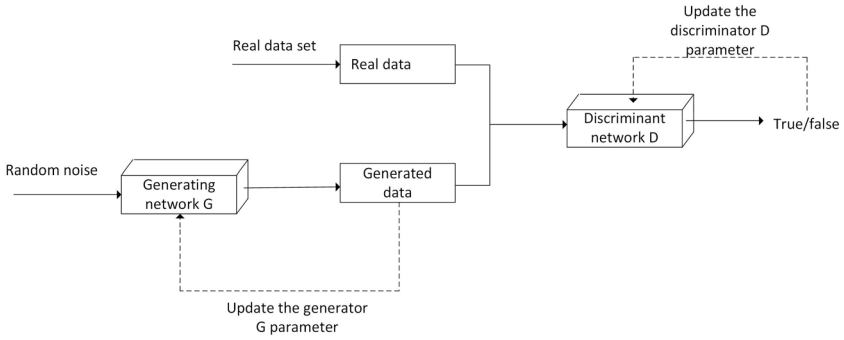

Figure 1. Generator architecture

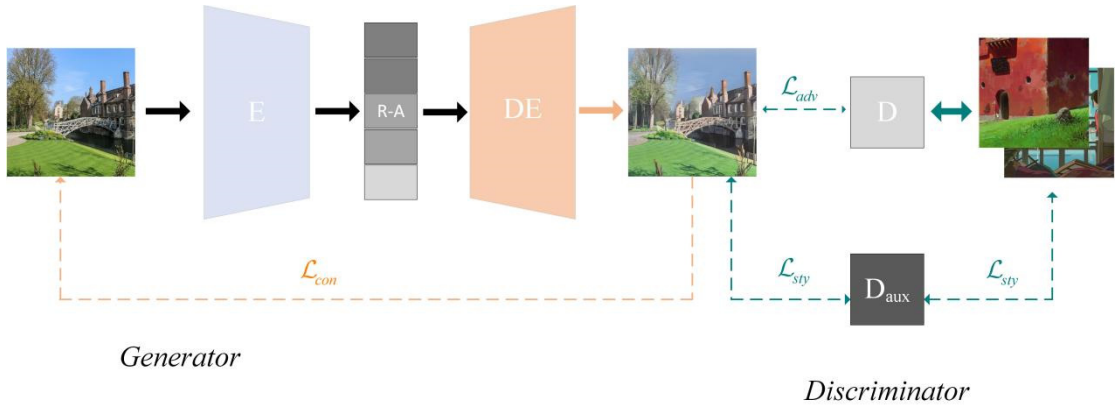

Figure 2. Network architecture

Fig. 2 shows an overview of the method architecture of this paper. In the generator module we denote the shared encoder as E and the shared decoder as DE, and the residual aberration module (R-A) is used between the shared encoder and the shared decoder for image feature extraction. The discriminator module consists of two parts: one is D, which is used to determine whether an image is a real cartoon image or a synthesized image, and the other is the auxiliary

ISSN: 2688-8645

classifier Daux which assigns style labels to real or generated cartoon images. It is specifically described as follows.

2.1. Adaptive Instance Normalization

Cartoon images tend to have rather unique features that are far from real photographs, which means that there is a huge gap between the distributions of the photographs and the cartoon images.AdaIN [7] explicitly replaces the feature statistics of the photographs with the corresponding feature statistics of the cartoon images, which breaks the consistency and continuity of the feature maps. This inconsistency and discontinuity will improve the model generalization ability. Adaptive Instance Normalization (AdaIN) is an extension of Instance Normalization (IN), where AdaIN will receive content inputs and style inputs, and then the mean and variance of the channels corresponding to the content inputs will be used to match the mean and variance of the style inputs. Unlike batch norm normalization (BN) [8],IN [9], AdaIN to-be-learned affine parameters [10]. Instead, it computes the affine parameters adaptively from the stylized inputs:

Inspired by kim [11], we introduce adaptive layer instance normalization (AdaLIN) into the residual block. The choice of the normalization function has an impact on the transformed structure, especially for various datasets with different shapes and amount of texture variation.

2.2. Generator Architecture

For a given image p, the encoder E learns a mapping that transforms P into a shared latent space Q, which spatially encodes the image content of P and common features shared by all cartoon styles. Decoder DE learns a mapping from the space Q to the target cartoon manifold C. $p \rightarrow c$ as G(p), corresponding to the cartoon styles in the training data. Encoder E starts with a planar convolution stage followed by two downstream convolution blocks for spatial compression and coding of the image. This stage extracts local signals for downstream transformation. Then, five identical residual blocks are used to construct content and streaming features. The decoder DE reconstructs the output cartoon image of the style from four layout identical residual blocks and then by two upper convolutional blocks, which contain stepwise convolutional layers. Finally, an additional layer is added which consists of two other planar convolutional blocks of 32 and 16 channels, respectively, to enhance the details of the output (e.g., texture), followed by a convolutional layer with a 7 × 7 kernel.

2.3. Architecture of the Discriminator

We divide the architecture of the composite discriminator D into two parts: the classification network D and the auxiliary classifier Daux. we use a patch-level architecture [12] for determining whether the output image has a specific cartoon style. Unlike common classification tasks, the discrimination of cartoon style is based on local features of the image. So D is designed to be relatively concise. After the planar convolution stage, the network employs two stepwise convolution blocks to reduce the resolution and encode key local features for classification. The classification response is then obtained using a feature construction block and a 3 × 3 convolutional layer. LeakyReLU (LReLU) with $\alpha = 0.2$ is used after each classification layer [13-14]. To help the generator better represent the differences between cartoon styles. An auxiliary classifier, Daux [15], which assigns a style label to the input cartoon image that defines the style loss of the current style corresponding to the final objective function. daux contains four lower convolutional blocks followed by a planar convolutional order and a softmax activation function.

In the framework given in Fig. 2, the task of the discriminator D is to determine whether the input image is synthesized by the generator or by the real target manifold. However, we observe that simply training the discriminator D to separate the generated and real cartoon images is not sufficient to transform a photo into a cartoon. This is because the original cartoon image

ISSN: 2688-8645

has clear edges, but these edges are usually a small percentage of the overall image. If the output image does not clearly reproduce the edges, but has the correct shading, it becomes difficult to train the discriminator.

To circumvent this problem, in this paper, we use guided filtering for edge smoothing on the trained cartoon image [16], where general filters are isotropic filters, such as Gaussian filters, which smooth the image while erasing some of the edge frame details. Whereas guided filtering smooths the graph without blurring out the image edges and contours. A special case of the guided filter is that the guiding image is the input image itself, in which case the guided filter plays the same role as an edge-preserving filter, which is also the scenario used in this paper. The formula for the guiding filter is as follows, where the input image to be processed p, the guiding image I, and the filtered output result q:

$$q_i=\Sigma w_{ij}(I)pj \quad (1)$$

2.4. Loss Functions

(1) Adversarial loss

Because there is no ideal output result map in the task of cartoon style migration, that is, the model does not get the real output label. The idea of adversarial training of adversarial generative network can be a good solution to this problem, because it is not possible to give a specific limit in the pixel space, then the use of adversarial learning strategy, so that the model in the training process to learn the appropriate method of calculating the loss of the pixel level. In this paper, the specific formula for calculating the adversarial loss is as follows:

$$L_{adv(G,D)}=E_{ci\sim Sdata(c)}[\log D(c_i)]+E_{ej\sim Sdata(e)}[\log(1-D(e_j))] \quad (2)$$

(2) Style loss function

In order to make the style that generator G can produce significant, we further introduce a style loss. This loss utilizes an auxiliary classifier Daux. for a given input cartoon image C, which can be either real or generated, Daux outputs the likelihood of C belonging to the input cartoon style. Where se denotes the edge-processed cartoon image style and s denotes the style of the input cartoon image.

$$L_{ssty}(G_i,D_{aux})=E_{c\sim Sdata(c),s}[-\log Daux(s|c)]+E_{e\sim Sdata(E),se}[-\log D_{aux}(se|e)]+E_{pt\sim Sdata(p),s}[-\log Daux(s|G(pt))] \quad (3)$$

(3) Content loss

Content loss can be regarded as how well the content structure of the source image is preserved as follows

$$L_{con}(Gi)=\Sigma E_{p\sim Sdata}(p)[Re_l(H_h(G_i(p)))-Re_l(H_h(p))_1] \quad (4)$$

3. Experiments

This GAN model we implemented in pytorch. All experiments were performed on an NVIDIARTX3060 graphics card. In order to compare the method of this paper with the current state of the art, the data used for training and testing of the experiments are shown in Section 2.1. A comparison with existing cartoon style migration methods is shown in Section 4.2.

3.1. Dataset

The training dataset contains, real and cartoon images. The cartoon data was collected manually by us, we cropped from real animation videos, mainly from Makoto Shinkai's movies and Hayao Miyazaki's movies. For Makoto Shinkai's work we cropped 7718 images and for Miyazaki's we cropped 8000 photos. For the real data side, we used 4800 photos collected from Flickr, out of which 4000 photos were used for training and 800 for testing.

ISSN: 2688-8645

3.2. Comparison with Currently Available Techniques

The comparison of our approach with some related work is shown in Fig. CycleGAN results do not capture the cartoon style well. While cyclic consistent loss helps to preserve the content better, and while the results tend to reproduce some of the key textures of the cartoon images they seem to ignore the semantics of the cartoon images and are far from satisfactory.CartoonGAN generates high quality results with good textures and crisp edges but lacks abstraction and tends to distort the colors. The model in this paper prevents inappropriate color modifications.The White-Box model has very clear boundaries but focuses too much on smoothing the image using color blocks and enhancing edges, which results in the original image lacking small details and generating noise. Finally, our model is not perfect, but it provides balanced results in terms of color, texture, and still generates images with a cartoonish feel. In summary, our method outperforms previous methods in generating images with harmonious colors, clean edges, fine details, and less noise.

Figure 3. Comparison with each method

3.3. Evaluation Criteria

Since people's evaluation of style migration results is more subjective, in this paper we use the FID score to evaluate our method along with the subjective evaluation score.FID is used to evaluate the similarity between two drawing set [17] distributions, such as real images and generated cartoon images. The lower the score, the closer the two distributions are. Regarding the subjective evaluation scoring, thirty volunteers were selected to score the effectiveness and quality of the cartoon images for the given method (the lowest score was 1 and the highest score was 5).

Table 1 shows the FID scores and subjective scores of CycleGAN, CartoonGAN, White-boxGAN and our method, and the comparison of the values in the table shows that the image generated by our method has the smallest FID score and the highest subjective score, which proves that it has the largest number of cartoon/anime style results, which is better than other methods.

ISSN: 2688-8645

Table 1. Evaluation metrics

methods	FID Hayao style	FID Shinkai Style	Subjective scoring (Hayao style)	Subjective scoring (Shinkai style)
CycleGAN	160	167	3.1	3.3
CartoonGAN	140	125	2.8	3.7
White-Box	118	110	3.5	4.0
ours	85	78	4.4	4.3

3.4. Ablation Experiment

Figure 4. Ablation experiment

First, we investigate the role of adaptive instance normalization, auxiliary discriminator and the corresponding style loss function in the network through the ablation experiment. Fig. 4 shows the ablation experiments for two different style models, the first line is in Makoto Shinkai style and the second line is in Miyazaki style, where (a)(b) shows the results of our ablation analysis with complete loss of functionality, (a) for the lack of instance normalization to get the result, (b) for the lack of auxiliary classifiers and style loss, and (c) for the result of fusing all the functions. Observations show that each component plays an important role in this network. (1) Instance normalization guides the generator G to produce clear edge effects, which results in better cartoon-style images and avoids causing irregular textures. (2) The auxiliary classifiers coupled with the style loss allow the network in this paper to produce significant differences between styles. As a comparison, the results of our full model shown in (c), the results of this paper have smoother features, sharper boundaries, and less noise. Both contribute to the results of our method.

4. Conclusion

In this paper, we have proposed a method to convert real-world photos into cartoonization in multiple styles. Based on CartoonGAN, improvements were made in the following aspects. (1) Designing the generator as an encoder and decoder structure reduces the complexity of the framework. (2) Adaptive instance normalization is added to the generator module to improve the model generalization ability (3) Auxiliary classifiers are introduced to help the generators produce obvious style differences. (4) Add guided filtering to improve the discriminator's discriminative ability. Experimental results and survey studies show that the method proposed in this paper can generate better quality and more expressive cartoon images than current methods.

ISSN: 2688-8645

Acknowledgments

Supported by The Innovation Fund of Postgraduate, Sichuan University of Science&Engineering, the project number is (Y2022169).

References

[1] Fang W, Zhang F, Sheng V S, et al. A Method for Improving CNN-Based Image Recognition Using DCGAN [J]. Computers, Materials & Continua, 2018, 57(1).

[2] 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.

[3] 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.

[4] X. W,J. Y. Learning to Cartoonize Using White-Box Cartoon Representations[J]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2020.

[5] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[6] Liang Yue, Xu Linfeng, Liu Daiyao et al. A face caricature generation method based on attention mechanism[J]. Chinese Science and Technology Paper,2023,18(03):304-309+316.

[7] LIANG Yue, Xu Linfeng, Liu Daiyao et al. Face cartoon generation method based on attention mechanism [J]. Chinese Journal of Science and Technology,2023,18(03):304-309+316.

[8] X. W,J. Y. Learning to Cartoonize Using White-Box Cartoon Representations[J]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2020.

[9] Gao S H, Han Q, Li D, et al. Representative batch normalization with feature calibration [C]// Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 8669-8679.

[10] Kolarik M, Burget R, Riha K. Comparing normalization methods for limited batch size segmentation neural networks[C]//2020 43rd international conference on telecommunications and signal processing (TSP). IEEE, 2020: 677-680.

[11] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization [C] // Proceedings of the IEEE international conference on computer vision. 2017: 1501-1510.

[12] Kim J, Kim M, Kang H, et al. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation[J]. arXiv preprint arXiv:1907.10830, 2019.

[13] Yi J, Yoon S. Patch svdd: Patch-level svdd for anomaly detection and segmentation[C]//Proceedings of the Asian conference on computer vision. 2020.

[14] Xu J, Li Z, Du B, et al. Reluplex made more practical: Leaky ReLU[C]//2020 IEEE Symposium on Computers and communications (ISCC). IEEE, 2020: 1-7.

[15] Dubey A K, Jain V. Comparative study of convolution neural network's relu and leaky-relu activation functions [C]//Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018. Springer Singapore, 2019: 873-880.

[16] Shu Y, Yi R, Xia M, et al. Gan-based multi-style photo cartoonization[J]. IEEE Transactions on Visualization and computer graphics, 2021, 28(10): 3376-3390.

[17] Ochotorena C N, Yamashita Y. Anisotropic guided filtering[J]. IEEE Transactions on Image Processing, 2019, 29: 1397-1412.

[18] Obukhov A, Krasnyanskiy M. Quality assessment method for GAN based on modified metrics inception score and Fréchet inception distance[C]//Software Engineering Perspectives in Intelligent Systems: Proceedings of 4th Computational Methods in Systems and Software 2020, Vol. 1 4. Springer International Publishing, 2020: 102-114.