Anime Style Transfer using GAN Model

Arlan Kalin
UNIST
arlan914unist@unist.ac.kr

Talgatkyzy Akniyet
UNIST
akniyet@unist.ac.kr

Abstract

This paper describes the working process of rendering a content image into anime style image, achieving so-called style transfer. Progress in this kind of research is growing rapidly and achieving amazing results. The main influence to artistic style transfer was made by Gatys et. al. However, this method gives unsatisfying results in transforming images into an anime style. In addition, the existing anime style transfer methods are mostly designed to translate a portrait photo-face into an anime appearance. Our aim is to present a model that can turn real-world scene photo images into images with Japanese animation. Our model is based on a generative adversarial network (GAN), to make the generated images have better animation visual effects. Furthermore, we train this model using a dataset consisting of anime scenes. Based on experimental results, this method can rapidly convert real-world photos into high-quality anime images that outperform existing methods.

1 Introduction

1.1 Motivation

In the contemporary world, cartoons scenes play a huge role in modern society. In particular, series such as Anime have a great influence on young people. Undoubtedly, anime displays the beauty of the modern world. Perhaps many of us have seen picturesque landscapes and scenes in such series and films as Naruto, Death Note and Your Name. From such frames it is breathtaking and it becomes interesting how the author was able to convey a beautiful picture in such a detailed quality. However, if we look behind the curtain, we can see the tremendous amount of work that the author puts into every line, every building, every frame in the anime. It may be months or even years before an anime is released and viewers can enjoy it. Of course, there are video editors that help animators draw and render sections and scenes of a video faster. However, this still requires manual rendering of the base model and the rendering process requires more automation in today's realities.

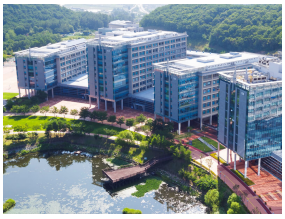


Figure 1: Content image.

Figure 2: Our result.

1.2 Real-world to Anime

Attempts to reproduce the landscapes of the modern world using different styles of arts began back in 2016 using Convolutional Neural Networks [2] and a 19-layer architectural model for this network called VGG19 [9]. Later, the ability to transfer art styles led to the transfer of Disney character styles to people's faces using the cartoon dataset. This development was made possible thanks to StyleGAN [7] - a Generative Adversarial Nets that allows users to generate images from an entirely novel domain and do this with a degree of control over the nature of the output. After that, this framework nevertheless led us to anime, or rather, the use of relationships between different sources and the use of styles on pictures. This development was called MangaGAN [10] and AniGAN - it was based on the recognition of facial features. Although, these applications could be trained on the dataset of anime fragments, it was not applicable on the real-world landscape objects. Since they were trained on facial features, output of landscapes contained high blurs and linear errors. But, the problem is solvable using Neural Network techniques and right training datasets.

1.3 Contribution

This paper is aimed to propose Cartoon Style Transfer on the real-world landscapes using GAN Model using Generator and Discriminator approach leading to less noise fragments on the image. Furthermore, in this research we:

1. Implented Generator and patch-level discriminator in order to teach model to convert photo into cartoon.
2. Compared efficiencies of existent model and ours.
3. Showed how parameters and datasets can affect the output of the model.

2 Related work

Neural Style Transfer. In 2015 Gatys et al. published their work "A Neural Algorithm of Artistic Style", which was a key moment in discovering the ability of neural networks to split the style of image from its content. Gatys et al. applied the gram matrix to different local feature maps extracted by the VGG-19 network and calculated the correlation between features to form a statistical model. This method performed impressive results, but its optimization is prohibitively slow and its main focus is on artistic style transfering.

Li and Wand [5] introduced a framework based on markov random field (MRF) combined with discriminatively trained deep convolutional neural networks (dCNNs). The category of MRF considers NST at a local level, i.e. works with patches to match the style and because of that, fine structure and arrangement are better preserved. However, this method is more effective for photorealistic styles and usually fails when the content and style of the images have strong differences in perspective and structure. Although these neural style methods are good in translating images into some artistic work, they are still not good at producing animation styled images.

Generative Adversarial Networks. In recent years, there has been an increased interest in the field of neural network research related to image generation. First of all this is due to the fact that the model of generative-adversarial neural networks has been presented, with the help of which it has been possible to achieve significant advances in this area. This model consists of generator, which learns to produce the target output, and a discriminator, which learns to distinguish true data from the output of the generator.

A generative adversarial learning technique was used by Isola et al. [3] and Li et al. [6] to transform two domains. A further method for training unpaired image datasets was introduced by Zhu et al.[13] Choi et al.[1] aimed at obtaining one single model to transfer multiple artistic styles. A few improvements were made by Zhang et al.[12], using attention-based style transfer techniques. The results of these techniques, however, are unsatisfactory when it comes to transferring animation styles.

3 Method

Our approach is based on the usage of two Convolutional Neural Networks - A Generator and a Discriminator. The role of the Generator is to transform the photos of real-world scenes into anime manifold and the role of the Discriminator is to discriminate produced Generator output from the real-world picture. The process of learning of transforming real images into a cartoon is formulated as follows. We take the representation of real-world image P and map it with corresponding cartoon representation C. Thereby,

$$S_{data}(p) = \{p_i | i = 1...N\} \subset \mathcal{P}$$

$$S_{data}(c) = \{c_i | i = 1...M\} \subset \mathcal{C}$$

where N and M are the numbers of photo and cartoon images in the training set, respectively.

3.1 Architecture

Generator. For the Model Training, we used Generator Architecture which mainly consisted of standard convolutions, upsampling and downsampling model and inverted residual blocks(IRB) [8] (**See figure 3**). During the stage of down-convolution, we extract valuable local signals. Then, using eight identical residual blocks, content and manifold are being formed. Eventually, the cartoon manifold of image is reconstructed by up-convolution stage with 1/2 stride and final 7x7 kernel layer.

Discriminator. Additional to the generator network, the discriminator network D is used to decide whether the input image is a true cartoon image. The simple patch-level discriminator with fewer parameters in D is used since task for the discriminator is not complex. In the model, cartoon style discrimination relies on local features of the image substituting image classification. Generally, we used straight convolutional layers. After the stage

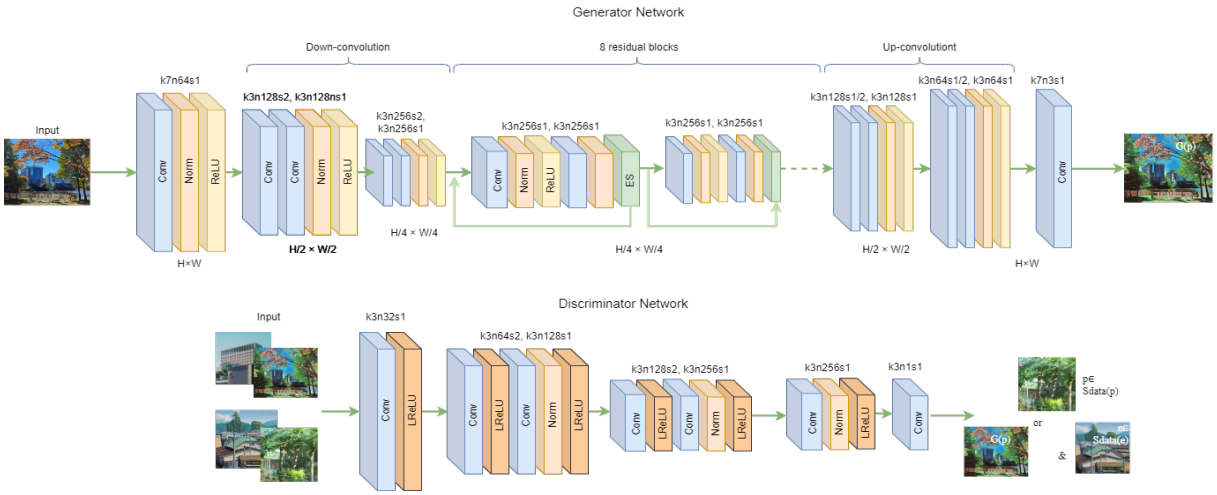


Figure 3: The general Architecture of Generator and Discriminator model with the block type depicted.

with flat layers, the network operates two stridden convolutional blocks to reduce the resolution and encode actual local features for classification. Thereafter, a feature construction block and a 3 × 3 convolutional layer are used to obtain the classification response. Moreover, Leaky ReLU (LReLU) [11] with $\alpha = 0.2$ is used after each normalization layer to avoid the "died" ReLU problem.

3.2 Loss Function

Naturally, let $\mathcal{L}$ be the loss function and G^* and D^* be weights of networks. Therefore, our task will be represented as followed function

$$(G^*, D^*) = arg \min_G \max_D \mathcal{L}(G, D).$$

From here, Loss function can be rewrited as composition of separated losses

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

$$w_{gra}\mathcal{L}_{gra}(G, D) + w_{col}\mathcal{L}_{col}(G, D)$$

Here, $\mathcal{L}_{adv}$ - is adversial loss that directs generator network to achieve accurate manifold transformation with appropriate weight w_{adv}. On the other hand, $\mathcal{L}_{con}$ opposes big change by preserving image content during style transformation. Also, w_{con} adds balance and relation to the image retention. The larger the w_{con} - the more content information from the input photos is retained, and thus, more textured details are saved on the image. The w_{gra} and w_{col} as well as $\mathcal{L}_{gra}$ and $\mathcal{L}_{col}$ gives the control over texture and color reconstruction of the manifold.

4 Experiments

4.1 Datasets

First of all, our dataset consists of two types of photo: real-world landscapes photo as the content image and images from anime movies as the style image. During training all images are cropped and resized in a region of size 256×256 pixels. Furthermore, we don't need paired data, our model can work with unpaired images. For the training we use both real-photo images and anime images, but for the testing we only need real-world photos.

For the content image, we have a total 4318 images, in which 3455 photos are for training and 863 are used for testing. This data is taken from kaggle.

For the style images, since every artist has his own style in drawing, we are using a collection of key frames from the popular anime movies drawn and directed by different artists. Overall, we have 1752 images from the movie called "The Wind Rises", 1445 animation images from Makoto Shinkai's movies "Your Name" and "Weathering with you" and 1284 images that are taken from "Paprika" movie.

We can observe that distinct style datasets produce distinct output. For example, if we train our model on "Your Name", the result would be represented in muted tones. Otherwise, if we train the model on dataset from anime "Paprika", we will see output with red tones which are related to the mood of series.

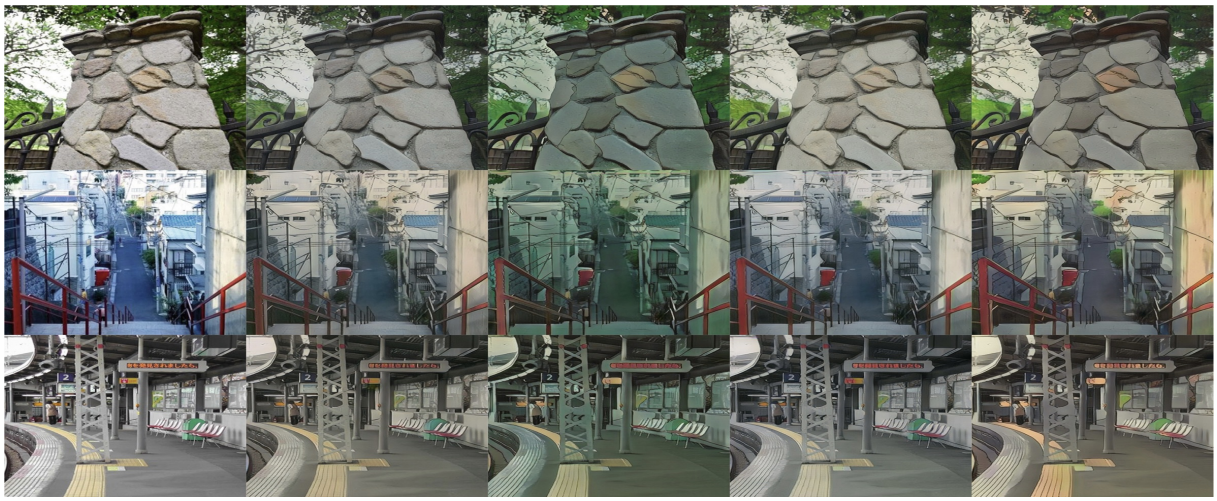
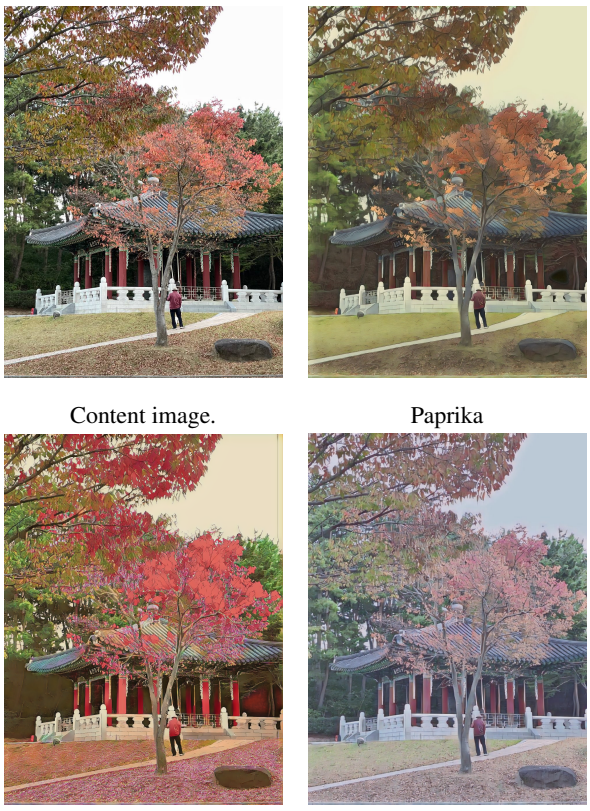


Figure 4: $(w_{adv}, w_{col}) = (0, 0), (250, 10), (350, 10), (300, 50), (300, 10)$



Content image. Paprika

The Wind Rises Your Name

Figure 5: Comparison of different datasets

4.2 Training

Pretraining. Since we use GAN, which is highly non-linear, the random initialization would make optimization tangled in the suboptimal local minimum. Therefore, to boost convergence of GAN, we pretrained the Generator G only with semantic $\mathcal{L}_{con}$ loss. This initialization training was executed on one epoch with the learning rate equal to 0.0001. Similar approach was used in [2].

Training. For the general training, we chose the learning rates of the generator and discriminator are 0.00008 and 0.00016, respectively. The number of epochs is 100, which is necessary for a reasonable result and the batch size is equal to 4. Adam optimizer [4] is applied to minimize the total loss. AnimeGAN is trained on a Google Colab Environment with Tesla T4 GPU and Pytorch.

Tuning. We are able to choose which w_{adv}, w_{con}, w_{gra} and w_{col} for the training. Below are shown results for different w_i configurations. As you can see on figure 4, the more we adjust w_{col}, the more detailed image it remains. For balanced result, we chose $w_{col} = 10$ and $w_{adv} = 300$.

5 Results/Analysis

5.1 Comparison with other models

To evaluate our method, we compare it with two types of style transfer methods: neural style transfer(NST) and style-attentional network (SANet).

Qualitive examples. In Fig. 6 examples of style transfer results generated by compared methods are shown. We can clearly see, the that these two methods are completely not suitable for anime style transformation. They loses the color of the original

Input Gatys SANet ours

Figure 6: Comparison with other models

content images and edges are not clear. In comparison, our result has higher smoothness and colors of the original photo are preserved, therefore our approach can produce higher quality animated visual effects.

6 Conclusion and Future Work

In this paper, we demonstrated Generative Adversarial Network to transform real-world photos into high-quality cartoon-style images. Besides, we observed how parameters of loss functions and datasets can affect the final output.

Since we made an exemplary step for Anime scene creation, still anime is being loved for dynamic scenes and stories. We would like to investigate how to make fluent scenes combined in video taking real-world video data and converting it into an anime episode. Also, we do not exclude adding sequential constraints to the training process to extend our method.

References

[1] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, pages 702–716. Springer, 2016.

[6] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. Gan compression: Efficient architectures for interactive conditional gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5284–5294, 2020.

[7] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020.

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

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

[10] Hao Su, Jianwei Niu, Xuefeng Liu, Qingfeng Li, Jiahe Cui, and Ji Wan. Mangagan: Unpaired photo-to-manga translation based on the methodology of manga drawing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2611–2619, 2021.

[11] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[12] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[13] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.