Style Transfer for Anime Sketches with Enhanced Residual U-net and Auxiliary Classifier GAN

Lvmin Zhang, Yi Ji and Xin Lin
School of Computer Science and Technology, Soochow University
Suzhou, China
lmzhang9@stu.suda.edu.cn, jiyi@suda.edu.cn

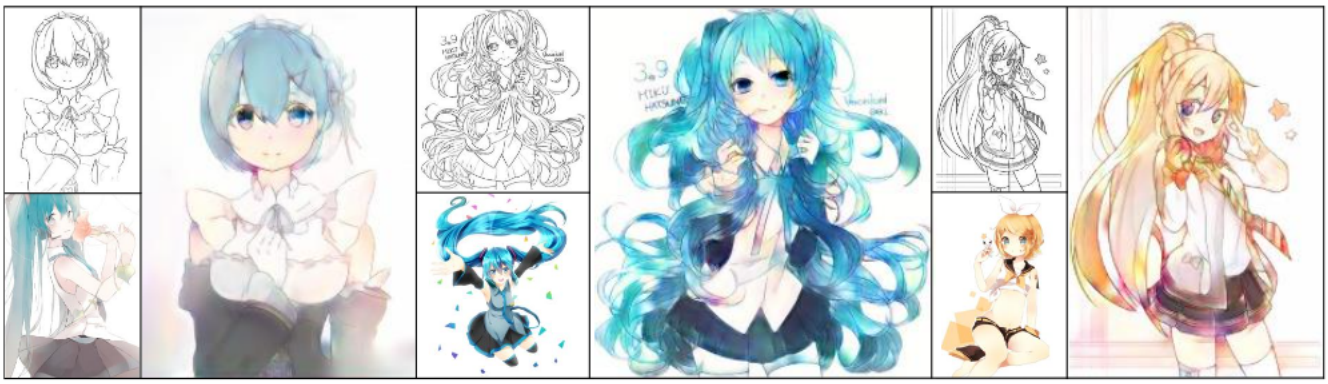

Figure 1: Examples of combination results on sketch images (top-left) and style images (bottom-left). Our approach automatically applies the semantic features of an existing painting to an unfinished sketch. Our network has learned to classify the hair, eyes, skin and clothes, and has the ability to paint these features according to a sketch. More results can be seen at the end of paper.

Abstract

Recently, with the revolutionary neural style transferring methods [1, 3, 4, 8, 17], creditable paintings can be synthesized automatically from content images and style images. However, when it comes to the task of applying a painting's style to a anime sketch, these methods will just randomly colorize sketch lines as outputs (fig. 7) and fail in the main task: specific style tranfer. In this paper, we integrated residual U-net to apply the style to the grayscale sketch with auxiliary classifier generative adversarial network (AC-GAN) [12]. The whole process is automatic and fast, and the results are creditable in the quality of art style as well as colorization.

1. Introduction

In the community of animation, comics and games, the majority of artistic works can be created from sketches, which consumed a lot of time and effort. **If there is a method to apply the style of a painting to a half-finished work in form of sketches, many redundant work will be saved**, e.g. using an existing picture of a specific character as a style map and then apply the style to a sketch of the character. The neural algorithm of artistic style [3] can produce amazing and perfect images combining of content maps and style maps, but it lacks the ability to deal with sketches. The paintschainer [18], as well as pix2pix [7], can turn sketches into paintings directly and the result can be even perfect with pointed hints added, but it cannot take the advantage of existing paintings. We investigated residual U-net and auxiliary classifier GAN (AC-GAN) as a solution and our network can directly generate a combination of sketch and style image. Our model is fully feed-forward so as to synthetize paintings at a high speed. Furthermore, we find out that U-net and conditional GAN (cGAN) relatively declines in performance with absence of a balanced quantity of information of paired input and output, and we propose using residual U-net with two guide decoders. In addition, we compare many kinds of GANs and find that conditional GAN is not suitable for this task, resorting to the AC-GAN finally.

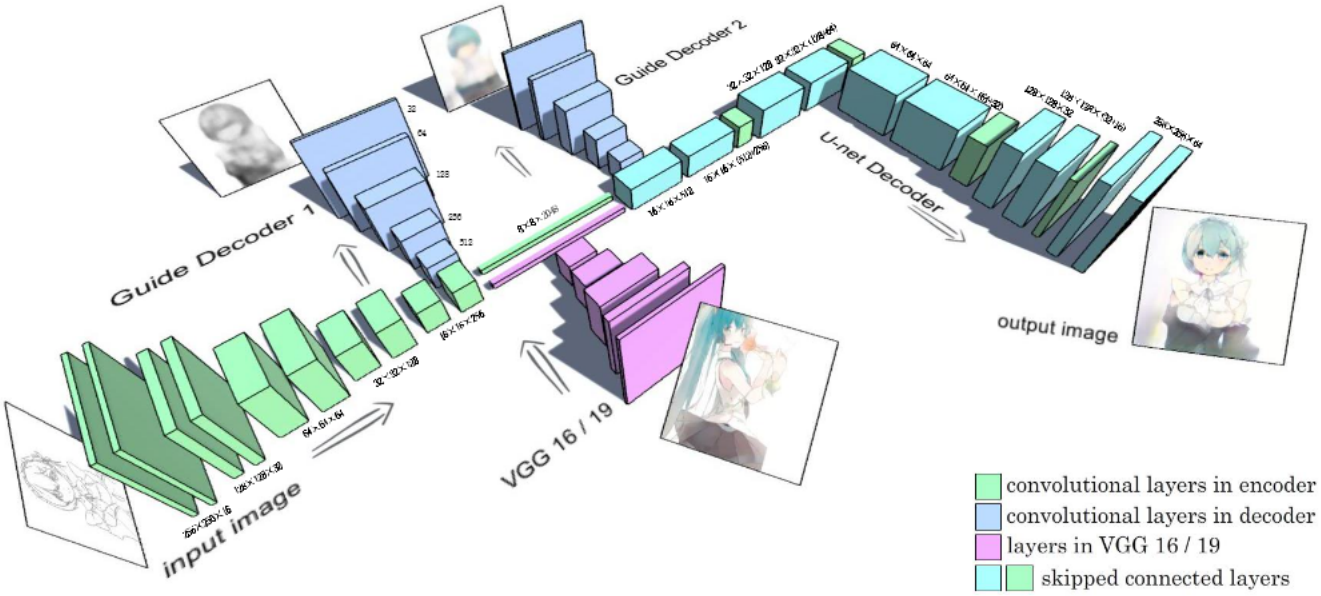

Figure 2: The architecture of the generator. The generator is based on U-net with skip connected layers. The weights of the VGG 16 or 19 [15] is locked in order to avoid being disturbed while training. The two "Guide Decoder" are located at the entry and exit of mid-level layers. The 4096 outputs of VGG's *fc1* are regarded as global style hint and added to the mid-level layers after a global normalization. For better performance in training, we add a dense 2048 layer above the *fc1*.

Our contributions are:

- A feed-forward network to apply the style of a painting to a sketch.
- An enhanced residual U-net capable of handling paired images with unbalanced information quantity.
- An effective way to train residual U-net with two additional loss.
- A discriminator modified from AC-GAN suitable to deal with paintings of different style.

2. Related Works

Neural Style Transfer [1, 3, 4, 8, 17] can synthesize admirable image with art style from an image and content from another, based on an algorithm that minimize the difference in gram matrixes of deep convolution layers. Nevertheless, our objective is to combine a style image and a sketch. Unfortunately, the neural style transfer is not capable of this kind of task. In fact, the results of neural style transfer on sketches from style images can be really strange, far from a proper painting.

Pix2Pix [7] and some other paired image2image transfers based on conditional GAN [11] are accomplished in transformation between paired images. In our research, the quality of the outputs of networks based on cGAN depends on the information gap degree between the inputs and outputs. That is to say, if the input and output are similar in the quantity of information, the result can be reliable. In the experiment of Pix2Pix's edge2cat, based on users' input edges, small circles always means eyes, triangles regarded as ears, and closed figures should always be filled with cat hair. If we shuffle the datasets of cat2edge, bag2edge, and even more like house, bicycle, tree, dog and so on, the quality of outputs will decline accordingly. Our task is to transfer sketches correspondingly to paintings, which is far more complicated than cats or bags, and the generator of cGAN needs to learn semantic features and low-level features simultaneously. Actually, a conditional discriminator can easily leads the generator to focus too much on the relationship between sketches and paintings, thus, to some extent, ignore the composition of a painting, leading to ineluctable overfitting.

Paintschainer [18] has abandoned cGAN and resorted to an unconditional discriminator due to the same reason as above, and obtained remarkable and impressive achievements. It becomes a powerful and popular tool for painters. Users only need to input a sketch to get a colorized painting. They can add pointed color hints for better results. With massive existing finished paintings, though the demand for a method to colorize a sketch according to a specific style is extremly high, there is no reliable and mature solution for it.

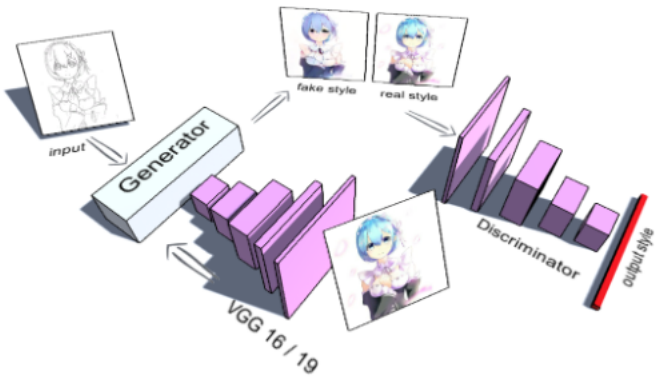

Figure 3: The architecture of our adversarial network. The discriminator is modified from AC-GAN, which not only has the ability to reveal whether the map is real or fake, but also tell the classification. We notice that the global style hint can be regarded as a low-level classification result with 2048 or 4096 classes.

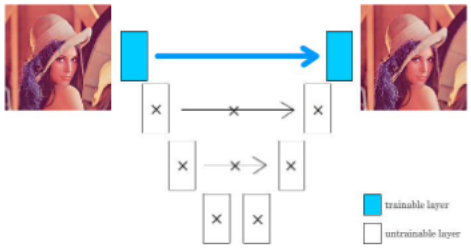

Figure 4: In the experiment of copying images, the mid-level layers of U-net receives no gradients.

3. Methods

We combine an enhanced residual U-net generator and an auxiliary classifier discriminator as our adversarial network. We feed the generator with sketch maps and style maps. Our discriminator can tell whether its input is real or fake, and classify corresponding style simultaneously.

3.1. Architecture and Objective of Generator

The detailed structure is in fig. 2. We are not the first to add the global style hint to mid-level layers of a encoder-decoder or a U-net [14]. In *Iizuka et.al's* [6] and *Richard Zhang et.al's* [20] the global hint is extracted from a high-level layer of a classification network as a high dimension vector and then added to the colorization network. For photograph colorization, the shadow, material and texture is known variables in inputs and the network only need a spot of information to analyse the color distribution. In *Iizuka et.al's* [6], they only use vectors of $1\times1\times256$ as global hints. However, it is far from enough for the network to paint sketches into paintings with various unique styles. Therefore, We integrate style hint at size of $1\times1\times4096$ or

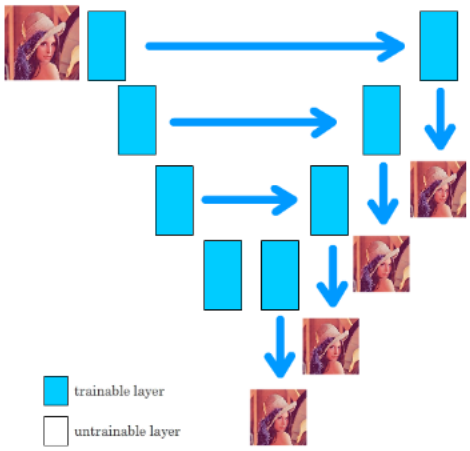

Figure 5: The ideal architecture of a residual U-net. Additional losses are applied to all mid-level layers so as to provide gradients for these layers.

$1\times1\times2048$. But we failed to train such a generator directly and finally found some "nature" of the well-known structure U-net. The failure result is in fig. 7.

The U-net is "lazy". That is to say if the U-net find itself able to handle a problem in low-level layers, the high-level layers will not bother to learn anything. If we train a U-net to do a very simple work "copying image" as in fig. 4, where the inputs and outputs are same, the loss value will drop to 0 immediately. Because the first layer of encoder discovers that it can simply transmit all features directly to the last layer of the decoder by skiping connection to minimize the loss. In this case, no matter how many times we train the U-net, the mid-level layers will not get any gradient.

For each layer in U-net's decoder, features can be acquired from higher layers or from skip connected layers. In each iteration of a training process, these layers select other layers' outputs with nonlinear activations in order to minimize the loss. In the experiment of copying image, when the U-net is initialized with Gaussian random numbers, the output of the first layer in encoder is informative enough to express the full input map while the output of second-to-last layer in decoder seems very noisy. Thus the "lazy" U-net gives up the relatively noisy features.

Because a hint of $1\times1\times256$ is far from enough for sketch painting, we resorted to a hint of $1\times1\times4096$, from the output of VGG 19's *fc1*, without the *Relu* activation. The 4096 hint vector is very informative and powerful. However, for a newly initialized U-net, the output of the mid-level layers can be extremely noisy if we add the vector of 4096 directly to the layer. As mentioned above, the noisy mid-level layer is given up by U-net and these layers cannot receive any gradient as a consequence.

Inspired by LeNet [9] and GooLeNet [16], We resort to residual networks in fig. 5. If we attach additional loss to

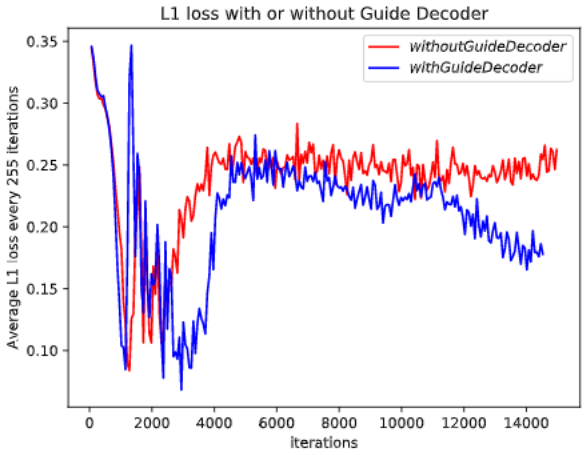

Figure 6: Loss with or without the Guide Decoder

Figure 7: The style map and sketch map are same with fig. 1. The neural style algorithm fails to transfer a sketch to a painting. Though normal U-net can predict meaningful paintings when the global hint is $1\times1\times256$, the result can be rather disappointing when the hint of $1\times1\times4096$ is applied. Our residual U-net with two Guide Decoders is very capable of handling such an informative global style hint.

each layer which is possible to be "lazy", no matter how noisy the output of a mid-level layer is, the layer will never be given up by the U-net and all layers will get stable gradient in the whole training process. Thus, it is possible to add a very informative and to some extent noisy hint to the mid-level layers. We implemented two additional loss, in the "Guide decoder 1" and "Guide decoder 2", to avoid the gradient disappearance in mid-level layers. The loss trend with or without the two Guide Decoders can been seen in fig. 6. Significant difference of the networks' prediction can be seen in fig. 7.

The loss is defined as:

$$L_{l1}(V, G_{f,g_1,g_2}) = \mathbb{E}_{x,y\sim P_{data}(x,y)}[||y - G_f(x, V(x))||_1 + \alpha||y - G_{g_1}(x)||_1 + \beta||y - G_{g_2}(x, V(x))||_1] \quad (1)$$

Where the x, y is the paired domain of sketches and paintings, and $V(x)$ is the output of VGG 19's $fc1$ without $Relu$, and $G_f(x, V(x))$ is the final output of U-net, the $G_{g_1}(x)$ and $G_{g_2}(x, V(x))$ are outputs of the two guide decoders at the entry and the exit of mid-level layers accordingly. The recommended value of α and β is *0.3* and *0.9*. We also find out that the distribution of color can be improved by feeding the Guide Decoder located at the entry of mid-level layers with grayscale maps, so the final loss is as below, where $T(y)$ can transform y into grayscale image.

$$L_{l1}(V, G_{f,g_1,g_2}) = \mathbb{E}_{x,y\sim P_{data}(x,y)}[||y - G_f(x, V(x))||_1 + \alpha||T(y) - G_{g_1}(x)||_1 + \beta||y - G_{g_2}(x, V(x))||_1] \quad (2)$$

3.2. Architecture and Objective of Discriminator

As we mentioned, the traditional cGAN is not suitable for our project. Painting is a complicated work and needs human artists to take color-selection, composition and fine-tuning into consideration, and all these need an artist to focus on the global manner of the painting. However, a conditional discriminator has always a tendency to focus much more on the relationship between sketch line and color than the global information. In our experiments with Pix2Pix, if a conditional discriminator is applied, the generator will resist fiercely and the color surrounding the line can be extremely over-colorized. Tuning parameters is not enough to reach a balance.

Furthermore, it can be appreciated if the discriminator has the ability to tell the style and provide gradient accordingly, in order to fit the main task: style transfering. We finally integrate AC-GAN and our discriminator has exactly 4096 outputs, which will all be minimized to 0 when the input is fake and approach to the same value of VGG 19's *fc1* when the input is real in fig. 3.

The final loss is defined as:

$$L_{GAN}(V, G_f, D) = \mathbb{E}_{y\sim P_{data}(y)}[Log(D(y) + (1 - V(y)))] + \mathbb{E}_{x\sim P_{data}(x)}[Log(1 - D(G_f(x, V(x))))] \quad (3)$$

Unfortunately, to minimize the loss, it requires a large memory size in GPU. We also employed the traditional discriminator of DCGAN for fast speed and large batch size. We fine-tune our networks by shifting the two loss functions.

$$L_{GAN}(V, G_f, D) = \mathbb{E}_{y\sim P_{data}(y)}[Log(D(y))] + \mathbb{E}_{x\sim P_{data}(x)}[Log(1 - D(G_f(x, V(x))))] \quad (4)$$

The final objective is:

$$G^* = \arg\min_{G_f}\max_{D} L_{GAN}(V, G_f, D) + \lambda L_{l1}(V, G_{f,g_1,g_2}) \quad (5)$$

Figure 8: Additional results of our network. All the reference paintings are from Pixiv. The creditions for all artists are available in the github page "style2paints".

4. Limitations and Discussions

Our network is capable of drawing on sketches according to style maps given by users, depending on the classification ability of the VGG. However, the pretrained VGG is for ImageNet photograph classification, but not for paintings. In the future, we will train a classification network only for paintings to achieve better results. Furthermore, due to the large quantity of layers in our residual network, the batch size during training is limited to no more than 4. It remains for future study to reach a balance between the batch size and quantity of layers.

References

[1] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Computer Vision and Pattern Recognition*, page To appear, 2017. 1, 2

[2] Emily L Denton, Soumith Chintala, arthur szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. pages 1486–1494, 2015.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge. A Neural Algorithm of Artistic Style. *ArXiv e-prints*, August 2015. 1, 2

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 1, 2

[5] Ian Goodfellow, Jean Pougetabadie, Mehdi Mirza, Bing Xu, David Wardefarley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[6] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)*, 35(4), 2016. 3

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition*, page To appear, 2017. 1, 2

[8] Justin Johnson, Alexandre Alahi, and Fei Fei Li. Perceptual losses for real-time style transfer and super-resolution. *Computer Vision ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*, 9906:694–711, 2016. 1, 2

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3

[10] Y Li, E Adelson, and A Agarwala. Scribbleboost: adding classification to edge-aware interpolation of local image and video adjustments. In *Nineteenth Eurographics Conference on Rendering*, pages 1255–1264, 2008.

[11] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 2

[12] A. Odena, C. Olah, and J. Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs. *ArXiv e-prints*, October 2016. 1

[13] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 3

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[16] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. pages 1–9, June 2015. 3

[17] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *CoRR*, abs/1603.03417, 2016. 1, 2

[18] Taizan Yonetsuji. Paintschainer. *github.com/pfnet/PaintsChainer*, 2017. 1, 2

[19] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666, 2016.

[20] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017. 3