Style Transfer for Avatar Sketches with Enhanced Residual U-net and Deep Convolutional GAN

Dixing Xu, Chenhui Wang, Yiting Wang **Supervised by Prof. Steven Guan**

Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University

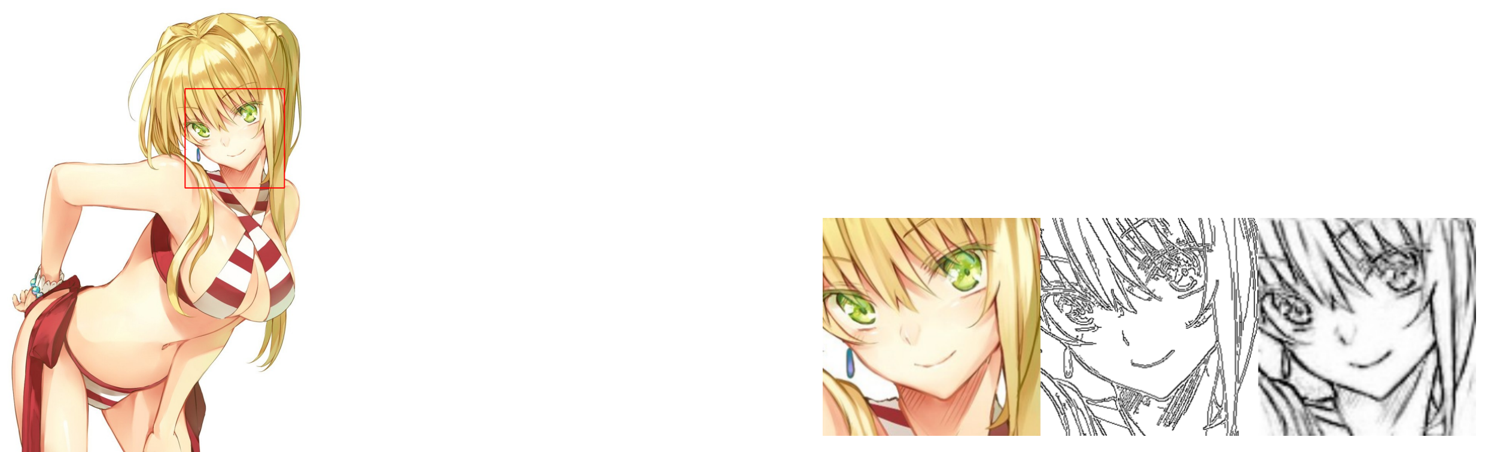
Code: 201733

Abstract

Convolutional Neural Networks(CNN) has been widely used in style transfer problems by separating and recombing image content and style. However, for painting a sketch according to a specific color style, traditional methods fail to learn the color distribution with respect to the sketch. In this project, we propose a novel structure based on conditional Generative Adversarial Networks(cGAN) combing U-net generator and DCGAN discriminator for the task of specific style transfer. Besides our main task, we also achieve better results on edge detection and object segmentation combing traditional methods with CNN.

Data Preparation

In this project, we have collected 51223 anime images in total from *Safebooru*. The image data are pre-processed using *lbpcasacde animeface* to retrieve only the facial part of the image. Figure 1(a) shows the process of object detection for face using CNN while Figure 1(b) shows the results of edge detection using Canny and Neural Network. (Note: This network is trained with paired images of human sketches and paintings and use U-net generator to reverse the process of colorization)



(a) edge detection

(b) From left to right: original, Canny, Neural Network

Figure 1: Preprocessing the image

Related Work

Traditionally, an encoder-decoder network servers as the solution to image-to-image translation problem. Such network passes information only from the previous layers. However, for style transfer problem, we need not only consider high-level structure, but also low-level information shared between input and output. With this in mind, we adopt U-net as our generator. However, merely using a vanilla U-net will just have the same effect as autoencoder and lead to gradient vanishing. Thus, we added two decoders which will generate additional output and prevent the plummet of the loss.

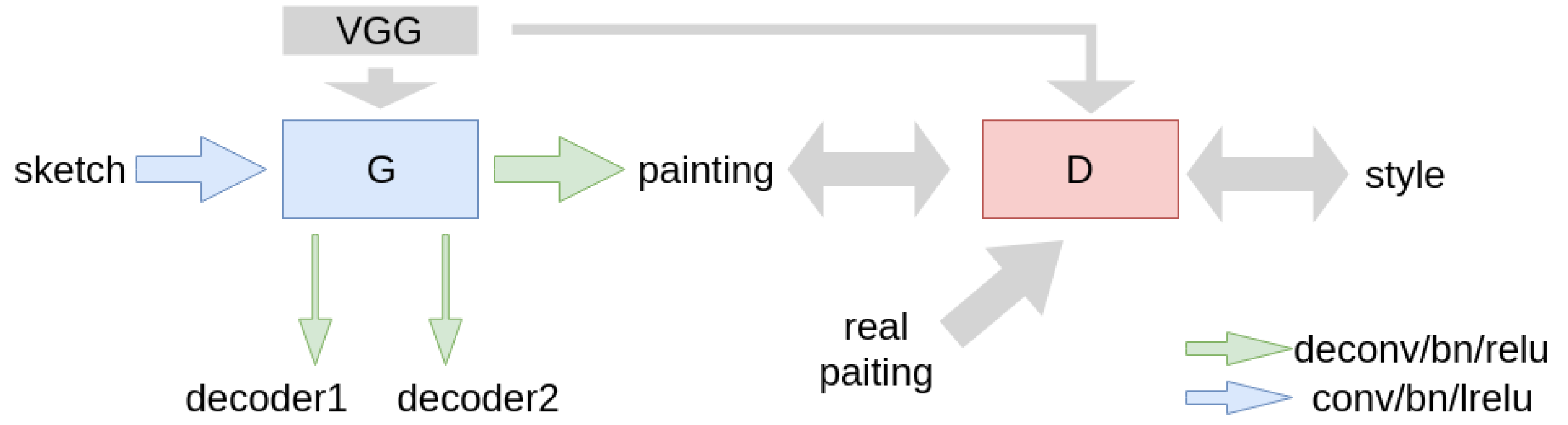


Figure 2: network architecture

Methods & Network architecture

As shown in Figure 2, we employed an enhanced residual U-net generator and a deep convolutional discriminator as our adversarial network. The sketch is feeded as input with a style hint learned from vgg fc1 without ReLU. The discriminator is used to classify the style and distinguish the fake images from the real ones. Our final objective is:

$$G^* = \arg\min_G \max_D L_{GAN}(V, G_f, D) + L_{l1}(V, G_{f,g_1,g_2}) \tag{1}$$

Enhanced Residual U-net Generator

Assuming we have n layer network. The ith $(i > 1)$ layer in U-net is able to learn not only from the $i-1th$ layer but also $n-ith$ layer. Previous work has shown that if the task is "easy", the middle layers in U-net will not receive any gradients. Therefore, we use a enhanced residual U-net shown at Figure 2. Two additional decoders are implemented to prevent gradient vanishing in the middle layers. The loss of our generator is defined as:

$$\begin{aligned} L_{l1}(V, G_{f,g_1,g_2}) = \mathbb{E}_{x,y\sim P_{\text{data}}(x,y)}[\gamma||y - G_f(x, V(x))||_1 + \\ \alpha||y - G_{g1}(x)||_1 + \beta||y - G_{g2}(x, V(x))||_1] \end{aligned} \tag{2}$$

Where the x is the sketch input and y is the corresponding painting. $V(x)$ is the style hint learned by VGG network. $G_{g1}(x)$,$G_{g2}(x, V(x))$, $G_f(x, V(x))$ are the results generated by decoder1, decoder2, U-net respectively. The L1 loss is used for pixel level feature alignment. From our experiment, without a coefficient assigning to the function, the loss will either go vanishing or exploding. The recommend values for γ, α, β are 1.8, 0.3, 0.9 respectively.

DCGAN Discriminator

We want the discriminator being able to collect the style information and provide corresponding gradients, thus we adopted the DCGAN discriminator and added additional features as Figure 2 shows. The shape of discriminator output is $1 \times 1 \times 2048$ which is of the same shape as learned result from VGG network. The discriminator will minimize of fake input to 0 and maximize the true input to $V(x)$. The loss of discriminator is defined as:

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

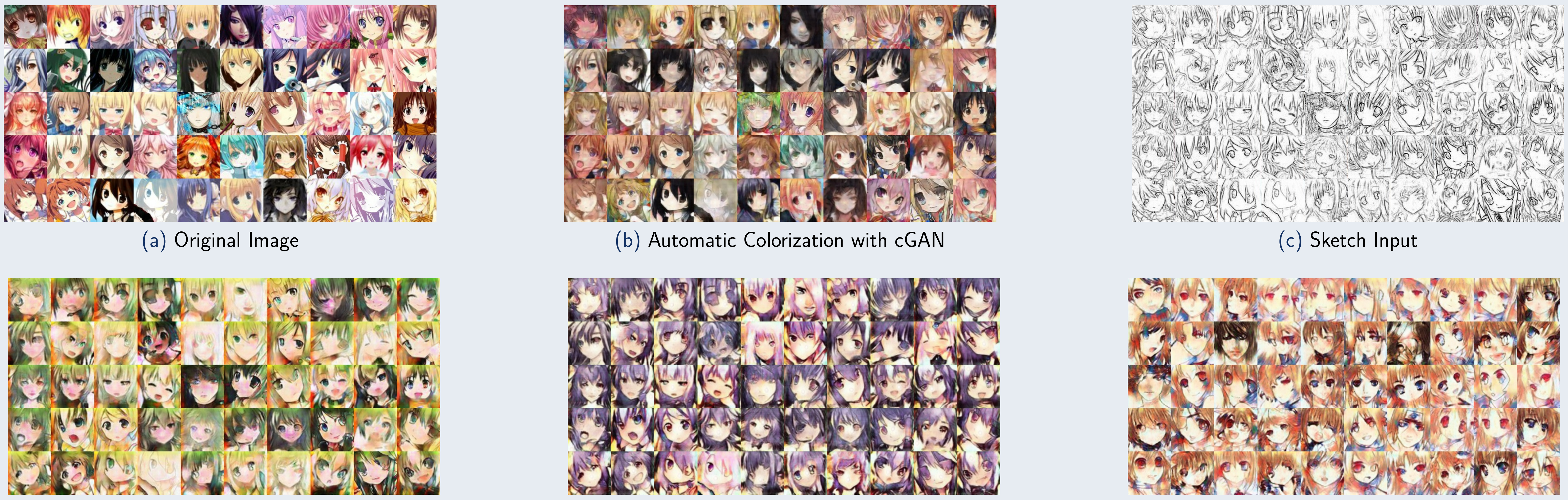
Although L2 Loss tends to output blurry results, in our experiment, we just calculate pixel level loss without the need to classify the result. Besides, we remove the log function from the traditional ones due to the instability of log function in Pytorch package.

Further Research

The next step is using ResNet as generator and improve the discriminator. Since paired data is rarely available in real-world settings, a novel architecture which is able to learn from separate domains is currently under development.

More results of this project is available at *http://stellarcoder.com/* and we will opensource our code soon.

Visualization: An image is worth a thousand words



(a) Original Image

(b) Automatic Colorization with cGAN

(c) Sketch Input

(d) Style Transfer Result (Style 1)

(e) Style Transfer Result (Style 2)

(f) Style Transfer Result (Style 3)

Figure 3: Result Visualization: (b) is the result generated by conditional GAN, (d)(e)(f) are the style transfer results generated by our network (trained 6 epoch)