DOM-E5129 Intelligent Computational Media, Project Report

Peizhe Hou, Student number: 765772

February 6, 2020

1 Introduction and motivation of the project

Nowadays the prosperity of machine learning increases the productivity of many areas, for example, computer vision and natural language processing, by substituting traditional hard-coded methods. While for industrial designers, they still put lots of efforts on coloring and putting visual effects on their sketches. In essence, the most important part of design is to have a good sketch, and coloring is just a kind of ornament. Thus, in the ideal case, after they've done their sketches, industrial designers could have some specialized tools that can turn sketches into concept art according to their demands, such as using blue as the main color or embellishing the product with red strips.

Starting from this point, I decided to investigate methods of translate sketches into colored ones via machine learning. To be honest I'm not familiar with machine learning methods, so I had to even learn basics of machine learning from the very beginning. However, I think it's worthwhile to at least know what's happening in the research community and perhaps dedicate to it in the near future.

There are mainly 2 methods that I learned and tried for, Style Transfer and CycleGAN. My learning outcome and some results of the experiment will be introduced in the following sections. **Source code, dataset, learning outcome and the report are available at the project repository [1].**

2 Learning process reporting

I've spent around 120-150 hours to first learn the basics of machine learning and neural networks (including 20 hours on the material provided by the course), and then specifically learn and implement Style Transfer and CycleGAN. As stated before, my learning goal is to know what's happening in the research community and hopefully catch it up in the level of practice. I think I've reached my learning goal and I'm satisfied with the learning outcome. The details of what I've learned in the learning process is available in **Learning Outcome.pdf** in my project Github repository.

3 Style Transfer

3.1 Problem Formulation

The problem setting of translating sketches into colored ones and transferring the style of images are similar in the sense that they cannot be done in supervised manner because the content image in most cases is not available in the target style. And this makes the problem tricky. Fortunately [2] proposed a method that could solve this problem via only a pre-trained feature extractor.

3.2 Model Set-up

This part is omitted and available in **Learning Outcome.pdf** in my project Github repository.

3.3 Experiment results

In my tiny experiment I used the pre-trained SqueezeNet model as the feature extractor to ease the computational load. Two of the results are shown in Figure 3.1, 3.2 and 3.3, 3.4.

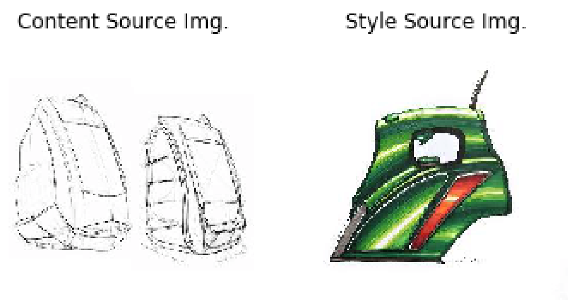

Figure 3.1: Content image and style reference image

Figure 3.2: Result

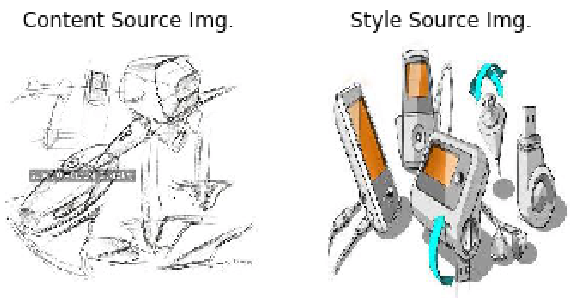

Figure 3.3: Content image and style reference image

Figure 3.4: Result

We can see that the result is not so good since the loss function apparently didn't capture the essence of the style reference. One of the possible reason is that the style difference between sketches and colored sketches are too subtle and hard to capture, which is already beyond the capability of Style Transfer method. Therefore what we've got is just a kind of overfitted result of noises in the style reference image. In this case it's wiser to make use of the advantage that we have a relatively dataset of colored sketches instead of having just one style reference image. Therefore I tried with another model, CycleGAN, which is a newer and better model to do unpaired image-to-image translation.

4 CycleGAN

4.1 Why use CycleGAN?

Style Transfer is good if we only have one single style reference image. But in my case actually the problem is more similar to what-so-called image-to-image translation, where we would like to learn a translation from one domain to another, for instance, translating summer landscapes to winter landscapes (or the reverse) or translating paintings to photographs (or the reverse). In addition, we have a bunch of images available both from domain A (sketches) and domain B (colored sketches) but most of them are not paired, which is exactly what CycleGAN is trying to solve. Therefore using CycleGAN may be more reasonable to color sketches for industrial designers.

4.2 What is a GAN?

This part is omitted and available in **Learning Outcome.pdf** in my project Github repository.

4.3 Structure of CycleGAN

This part is omitted and available in **Learning Outcome.pdf** in my project Github repository.

4.4 Experiment results

Although there's a dataset available for learning anime sketch colorization in [7], there seems no available dataset for industrial design sketches. I manually downloaded 100 images for both sketches and colored sketches, resize them into 256×256 jpg images and use them as the training set for the CycleGAN. Since for sure due to the small number of training set images the model will overfit, I didn't run any validation (the validation loss for sure wil be high), instead of logging the validation loss, I randomly downloaded additional images and use the model to color them in order to see the performance of CycleGAN.

Firstly, the loss histories of four individual models (Generator 1, Generator 2, Discriminator 1 and Discriminator 2) are shown in Figure 4.1, 4.2, 4.3 and 4.4.

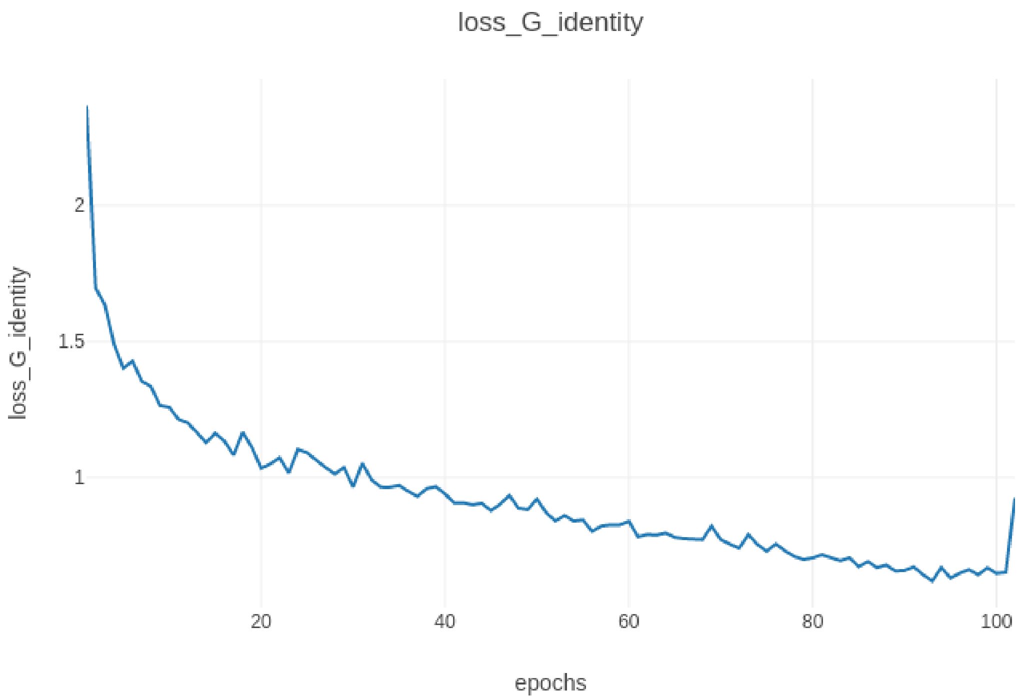

Figure 4.1: Identity loss history of the generator

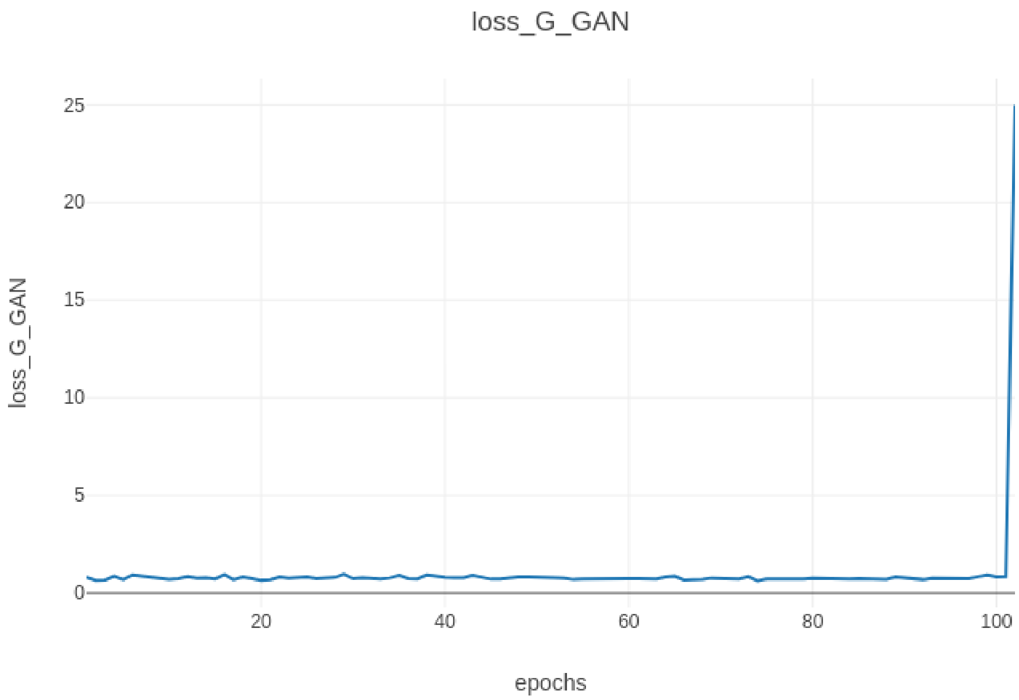

Figure 4.2: GAN loss history of the generator

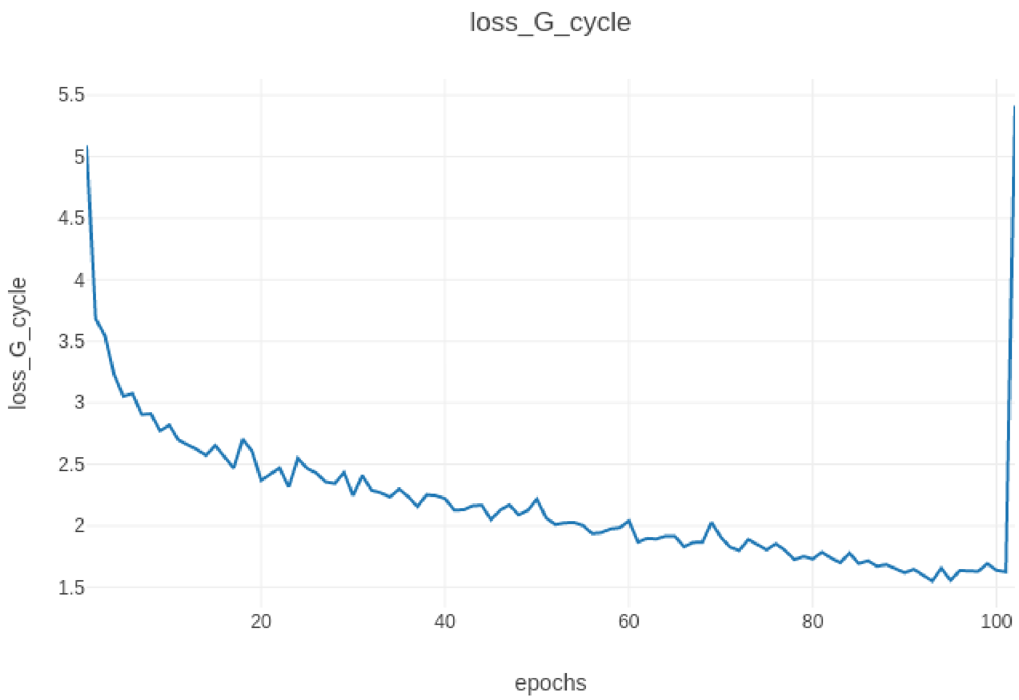

Figure 4.3: Cycle consistency loss history of the generator

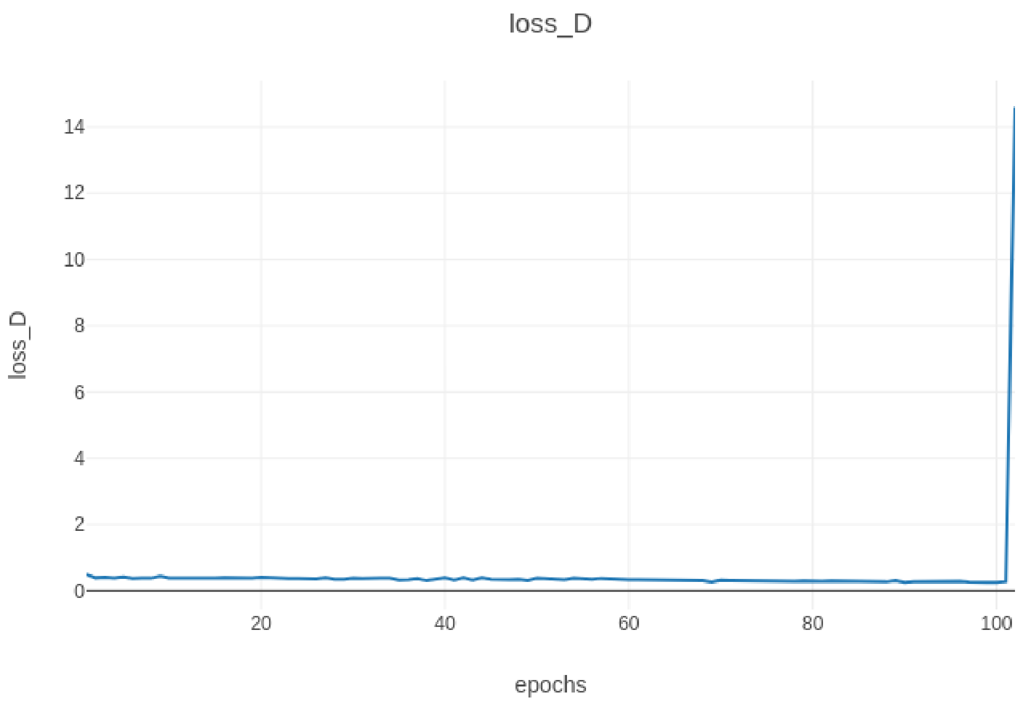

Figure 4.4: Loss history of the discriminator

Some validation results are shown in Figure 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11.

Figure 4.5: Fake sketch

Figure 4.6: Fake colored sketch

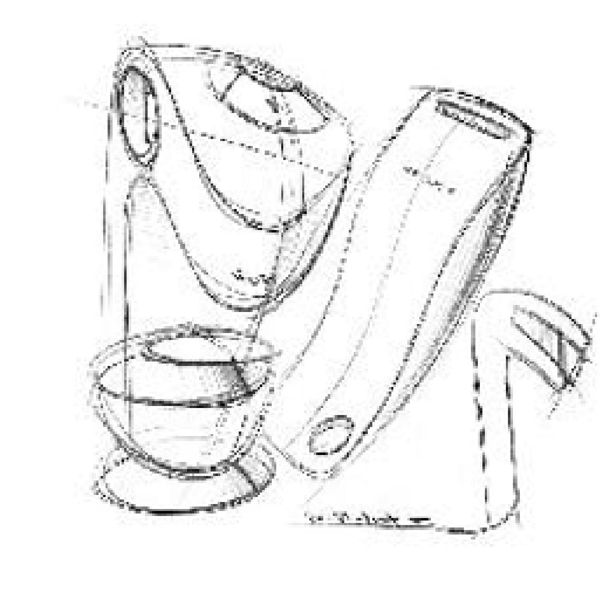

Figure 4.7: real sketch

Figure 4.8: real colored sketch

We can see that although the generated images obtained by CycleGAN are much better than those via Style Transfer, the model still generates noises on the background of the output image. In addition, the color pattern in the generated images are still not so diverse

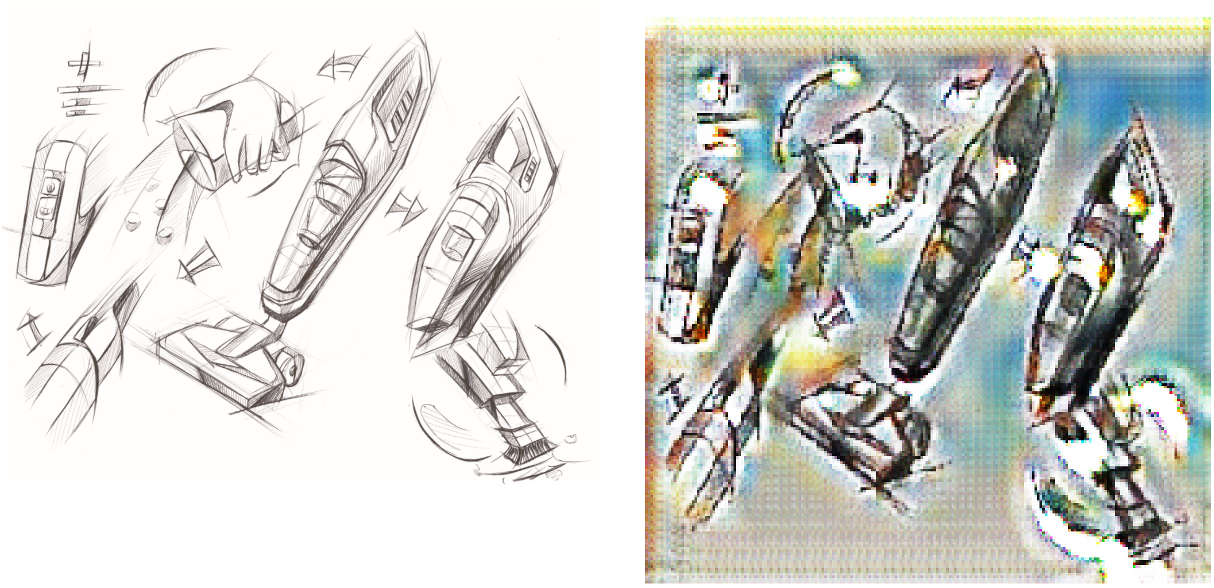

Figure 4.9: My own sketch 1 and Generated colored sketch 1

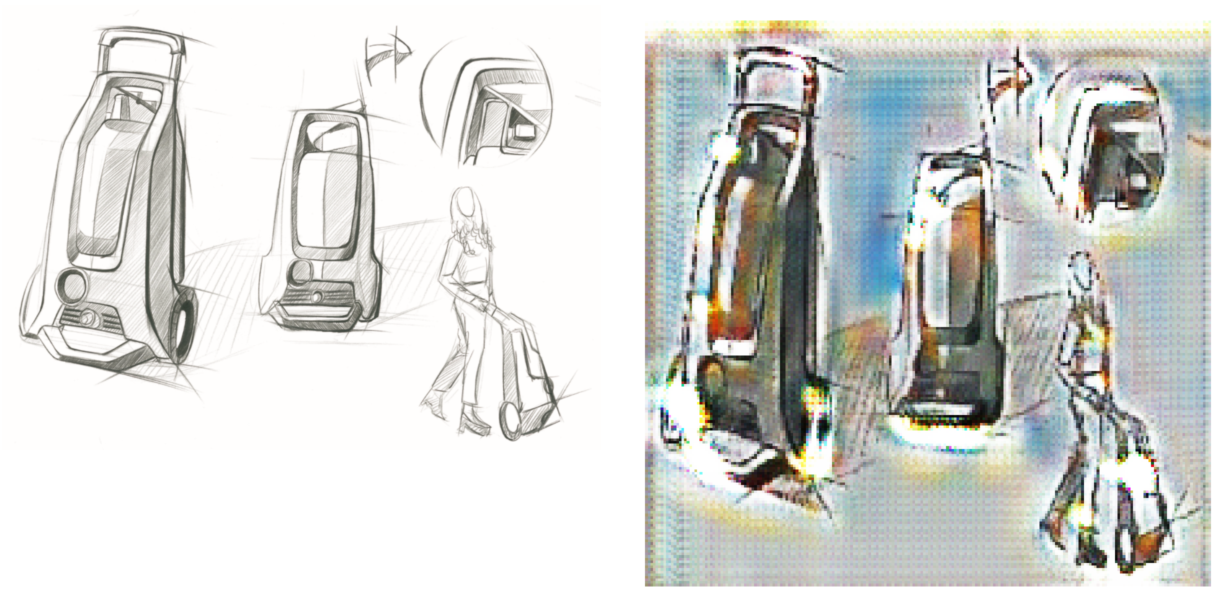

Figure 4.10: My own sketch 2 and Generated colored sketch 2

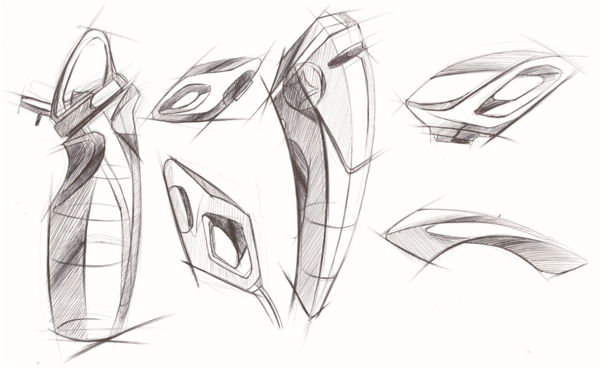 

Figure 4.11: My own sketch 3 and Generated colored sketch 3

compared to the color pattern in images of the training set. Perhaps enlarging the dataset could help mitigate this problem.

5 Conclusion

In this project I tried with Style Transfer first, but it didn't give any meaningful results. One possible reason is that the style (colorization) of the reference image is too unconspicuous that the style of the generated image has overfitted to the noises in the reference image, like the shadowing effect or any specific color. As the second method I've tried, CycleGAN has apparently better results even with extremely limited dataset (only 100 images for sketches and 100 images for colored sketches). I feel that there's still a lot of space to improve the performance, for example, collect more images, better data-preprocessing (as you may see, watermarks are not moved in many of the images in the dataset), or even come up with a better model. But generally I'm very satisfied with my learning outcome.

References

[1] Peizhe Hou. Project of Intelligent Computational Media. Github Repository. https://github.com/peizhehou/Project-of-Intelligent-Computational-Media

[2] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *CVPR*. 2016.

[3] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*. 2017.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, Yoshua Bengio. Generative Adversarial Nets. arXiv preprint arXiv:1406.2661, 2014.

[5] Aitor Ruano. Pytorch-CycleGAN. Github Repository. 2017. https://github.com/aitorzip/PyTorch-CycleGAN

[6] Chris Nicholson. A Beginner's Guide to Generative Adversarial Networks (GANs). https://pathmind.com/wiki/generative-adversarial-network-gan

[7] Taebum Kim. Anime Sketch Colorization Pair. Kaggle Dataset. https://www.kaggle.com/ktaebum/anime-sketch-colorization-pair

[8] Winnie Lin. Notes and assignments for Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition. Github Repository. https://github.com/cs231n/cs231n.github.io