Style Transfer Between Human and Anime Faces

Jianzhi Long
ECE
A14000205

Zi Zhang
ECE
A590102021

Abstract

We plan to tackle the problem of style transfer in computer vision with generative adversarial models. Specifically, we would like to transform human faces into anime-style faces. We train a CycleGAN model on the Selfie2Anime dataset, and obtained decent results. The code for this project can be found at `https://github.com/JamesLong199/ECE285_selfie2anime`.

1 Problem Definition

In this project, our goal is to generate anime-style human faces from real human selfies. We want to build a model that can produce outputs with an aesthetic anime-like style, while still preserving the unique facial features of the individuals in the selfie. Anime nowadays has become increasingly present in the pop culture today, and the purpose of this project is to explore the underlying aesthetic possibility for a creative interconnection between anime and our daily life.

Our project belongs to the category of style transfer in computer vision. Style transfer is the task of changing the style of an image in one domain to the style of an image in another domain. This is a more general version of image-to-image translation, where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs[8].

However, for most of the style transfer tasks that we are interested in, there is a lack of dataset that provides paired training data, or it is impossible to pair data at all. This is also true in our case, as there is no one-to-one correspondence between people in real life and characters in anime. As a result, we would like to use an algorithm that do not need paired input-output examples learn the mapping between images from two different domains.

2 Dataset

To train the model, we are going to use the Selfie2Anime dataset. The training set consists of 3400 women selfie images and 3400 anime face images, while the test set consists of 100 women selfie images and 100 anime face images. All images in the dataset are sized to 256x256 pixels.

3 Related Work

The advent of Generative Adversarial Network(GAN) has revolutionized unsupervised learning in the field computer vision[1]. Since then, it became possible to generate images of a particular style from our choosing. However, the input of GAN is only a single Gaussian random vector, therefore we do not have control over the content of the generated image except its visual style.

To make GAN generate images with certain desired content, Isola et al.[2] introduced Conditional GAN as a general purpose solution to image-to-image translation problems. Here the model takes an image from source domain along with a random noise vector as inputs, and then outputs the corresponding image that possesses the style of target domain while retaining its semantic content.

The image from source domain could be seen as a prompt that provide the model with the knowledge about what content to generate.

The main problem of this work is that the model needs paired images from the source and target domains to train on. Collecting and labeling such datasets with decent amount of paired data could be very labor-intensive and time consuming, even more so when we have to manually create data in one of the domains. As a result, despite being a general solution to image-to-image translation problems, the scope of usage of Conditional GAN is severely limited by the lack of paired data.

This limitation from lack of paired data is overcome by CycleGAN [8], which uses a pair of GANs to ensure cycle consistency between two domains. This model does not need random vector as input, and can perform style transfer from any one domain to the other. CycleGAN performs really well on tasks that involve color and texture changes, but often fails on those that require geometric changes.

To address the shortcoming on translating holistic and large scale geometric changes, Kim et al. proposed the U-GAT-IT model, which incorporates a new attention module and a new learnable normalization function [4]. The attention module let the model focus on more important regions distinguishing between source and target domains, while the novel normalization technique help to control the amount of change in shape and texture flexibly.

In this project, we are going to implement CycleGAN for our task, due to limited amount of time available and knowledge on the related subjects.

4 Method

4.1 CycleGAN

The CycleGAN model consists of two pairs of generator and discriminator. As a concept first proposed by Goodfellow et al.[1], the generator attempts to generate a realistic image in the target domain, while the discriminator tries to distinguish between a real image and a generated image. In the setting of image style transfer, given an input image from source domain X, we could like to match its visual style to that of target domain Y, while still preserving its semantic information. For example, converting a selfie of a woman with short hair and petite nose into an anime profile which also features short hair and petite nose.

Among the total of four networks, generator G transfer the style from source domain X to target domain Y, and discriminator D_Y tries to distinguish between the real and generated image in target domain Y. The other pair of networks, namely generator F and discriminator D_X does the opposite.

In the absence of paired training images, CycleGAN relies on ensuring cycle consistency to deliver satisfactory result. The generated image in target domain Y is passed through generator F again and vice versa, then the cycle consistency is optimized through a distance metric loss function.

4.2 Model Architecture

We follow the model architecture used in the CycleGAN paper[8], which was adopted from the work of Johnson et al[3]. For each generator, the input image is downsampled by a factor of 2 for two times and then passed through a series of residual blocks, and finally upsampled back to its original size in the same fashion.

The discriminator is a 70×70 PatchGAN, the input image is downsampled by a factor of 2 for three times. The output scores will have one channel and size 30×30, and the name PatchGAN suggests that each neuron in the output has a receptive field of size 70×70. Each score indicates the probability of its corresponding 70×70 image patch being real.

Generator	
Layer	Activation Size
Input	3x256x256
64x7x7 conv, stride 1, pad 3	64x256x256
128x3x3 conv, stride 2, pad 1	128x128x128
256x3x3 conv, stride 2, pad 1	256x64x64
9 x Residual Blocks	256x64x64
128x3x3 convTranspose, stride 2, pad 1, out_pad 1	128x128x128
64x3x3 convTranspose, stride 2, pad 1, out_pad 1	64x256x256
3x7x7 conv, stride 1, pad 3	3x256x256

Table 1: The model architecture of the two generators.

Discriminator	
Layer	Activation Size
Input	3x256x256
64x4x4 conv, stride 2, pad 1	64x128x128
128x4x4 conv, stride 2, pad 1	128x64x64
256x4x4 conv, stride 2, pad 1	256x32x32
512x4x4 conv, stride 1, pad 1	512x31x31
1x4x4 conv, stride 1, pad1	1x30x30

Table 2: The model architecture of the two discriminators.

4.3 Objective Functions

The two generator is jointly optimized by a combination of three objectives, namely the adversarial loss, the cycle consistency loss and the identity loss.

4.3.1 Adversarial Loss

The adversarial loss is the same as the one formulated by Goodfellow et al[1]. Given a generator mapping function G and its discriminator D_Y, we express the objective as:

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))] \quad (1)$$

In practice, the negative log-likelihood is replaced with least-square loss. As a result, the adversarial loss is expressed as:

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{x \sim p_{data}(x)}[D_Y(G(x))^2] + \mathbb{E}_{y \sim p_{data}(y)}[(1 - D_Y(y))^2] \quad (2)$$

4.3.2 Cycle Consistency Loss

According to Zhu el al.[8], Adversarial loss alone is not enough to guarantee that the learned function can map an individual input x_i to a desired output y_i. A network with large enough capacity could map the same set of input images to any random permutation of images in the target domain. Cycle consistency loss is introduced to reduce the space of possible outcomes. More specifically, after an input image in domain X goes through the two generators respectively, the output image should be nearly identical to the original input, i.e., $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$.

Mathematically, cycle consistency is optimized through $L1$ metric between the reconstructed image and the original image, and is expressed as

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y \sim p_{data}(y)}[||G(F(y)) - y||_1] \quad (3)$$

4.3.3 Identity Loss

For applications which we would like to preserve the color composition between the input and output, we apply an additional identity loss. Essentially, the generator should produce an identical output when real samples of the target domain are provided as the input: i.e.,

$$\mathcal{L}_{\text{identity}}(G, F) = \mathbb{E}_{y \sim p_{data}(y)}[||G(y) - y||_1] + \mathbb{E}_{x \sim p_{data}(x)}[||F(x) - x||_1] \tag{4}$$

4.3.4 Generator Loss

Overall, the full objective of the generator is a weighted combination of the above losses. λ is a weight coefficient hyperparameter, and we set it to 10 in our experiments, same as the default value used in the paper.

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) \\ + \mathcal{L}_{GAN}(F, D_X, Y, X) \\ + \lambda \cdot \mathcal{L}_{\text{cyc}}(G, F) \\ + 0.5 \cdot \lambda \cdot \mathcal{L}_{\text{identity}}(G, F)\end{aligned}$$

4.3.5 Discriminator Loss

The discriminator tries to maximize the likelihood of correctly distinguishing generated image from real image, and its objective can be expressed as

$$\mathcal{L}_{D_Y}(G, X, Y) = \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(X)))] + \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] \tag{5}$$

Again, the negative log-likelihood is replaced with least-square loss, and consequently

$$\mathcal{L}_{D_Y}(G, X, Y) = \mathbb{E}_{x \sim p_{data}(x)}[D_Y(G(x))^2] + \mathbb{E}_{y \sim p_{data}(y)}[(1 - D_Y(y))^2] \tag{6}$$

5 Training Details

We use Adam[5] optimizer with learning rate 0.0002 for all of our networks. Specifically, the two generators share one optimizer, and each of the two discriminators has its own optimizer. Also, we use a batch size of 1 for our experiments. We set the generator loss weight coefficient $\lambda = 10$, thus placing a high importance on the cycle consistency objective. Besides that, the discriminator loss is halved, equivalent to reducing the frequency of updating discriminator by a factor of 2.

To reduce model oscillation, we adopt the strategy of experience replay for adversarial training[7]. We use an buffer to store a history of 100 previously generated images. For each iteration, there is a 50% chance that the current generated image will become the input of the discriminator, while there is another 50% chance that an image randomly sampled from the buffer will serve as the input.

We initialize the weights for all of our network layers from zero-centered Normal distribution with standard deviation 0.02, same as in the DCGAN paper[6].

6 Results

6.1 Analysis of Visual Outputs

In general, the our task of transforming human faces into anime-style faces involves both color and texture changes and geometric changes. We found that the performance of CycleGAN is significantly affected by some visual properties of the input images, and our analysis on the underlying cause matches the conclusions that Zhu et al.[8] have discussed in their paper. From the test set images, the model produced satisfiable result on some samples while failed badly in some cases.

(a) Sample of anime images.

(b) Success cases of CycleGAN model.

(c) Failure cases of CycleGAN model.

Figure 1

6.1.1 Selfie vs Anime

While both the selfie images and the anime faces in the training data are depictions of female human images, there are several textual and structural differences between them. First of all, the training data contains selfie images with a wide variability in facial features, head poses and expressions. In contrast, anime images mostly features straightforward head pose, large cartoon eyes, a tiny dot for a nose and closed mouth with very thin lips. Humans have 3D facial structures while anime characters only have planar ones. Also, lighting condition varies significantly across selfie images, sometimes causing inconsistency of color composition on faces. In comparison, anime faces almost always have consistent color compositions.

6.1.2 Success Cases

Fig 1b displays the most successful transformations done by the model. In these samples, the head poses are all facing straightforward. There is large color contrast between human faces and the surroundings, and between skin and facial features like eyes, nose and mouth. Also, there is no complex facial expression involved and the facial texture is very consistent, i.e. no obvious shades or lines on the skin. These characteristics are similar to the visual properties of the anime faces. Therefore, the CycleGAN model mainly processes changes in color and texture, which generally leads to good result.

6.1.3 Failure Cases

Fig 1c displays the typical failure cases of the model. These samples often involves a high degree of head pose deviation, complex facial expressions, 3D facial textures and low color contrast. Also, as anime hairstyle often features a bang i.e. a curtain of hair on the forehead, the model is prone to generate a bang for the human faces, but often with different color and texture.

In some rare cases, the output image looks almost exactly the same as the input, having the same geometric structure and minor difference in color composition. We suspect that this tendency to produce similar output in the target domain is caused by the training objective and the convergence of adversarial training. When the equilibrium of adversarial training breaks and tilt toward the discriminator, the discriminators became increasingly good at distinguishing generated images while the generators struggle to improve. Also, as we place high importance on the cycle consistency objective, the model could cheat by letting both generators produce identical mapping to achieve a high cycle consistency.

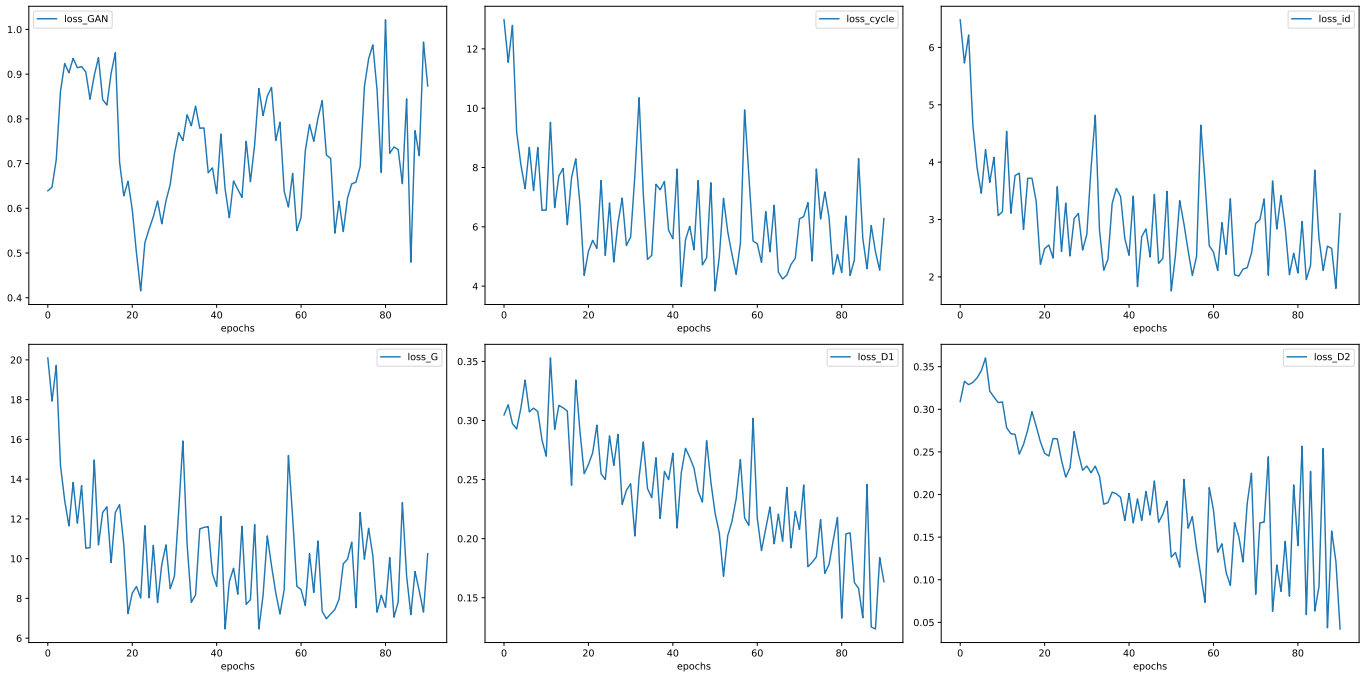

Figure 2: Training losses. First row from left to right: adversarial loss, cycle consistency loss and identity loss. Second row from left to right: total generator loss, discriminator loss for selfie, discriminator loss for anime.

6.2 Analysis of Training Losses

As we have placed a high importance on cycle consistency(factor of 10) and identity loss(factor of 5) in the training objective, the magnitude of cycle consistency loss and identity loss is much larger than that of the adversarial loss. Consequently, cycle consistency loss and identity loss follow a overall decreasing trend as the training goes on, as they are the priority of optimization. In contrast, the adversarial loss keeps fluctuating around the same level and sometimes almost reaching the maximum value of 1.

In the meantime, the discriminator losses also decreases steadily. Overall, as discussed previously, the range of possibly mappings that the generators could learn is narrowed down with low cycle consistency and identity losses; whereas the generators struggle to produce images that could pass the test of the discriminators.

Due to the limited capability of the model, the generator could never completely grasp the gist of style transfer for this particular task, especially with the geometric disparities between the source and target domains. On the other hand, the discriminators have learned how the generators would typically behave and consequently became increasingly better at distinguishing the generated images.

6.3 Discussion and Future Work

In conclusion, we have achieved decent outcomes on style transfer between human and anime faces with the classic CycleGAN model. However, the geometric disparities between the two image domains and the inherent weaknesses of the CycleGAN model have set limit on the quality of the generated images.

Given more time, we would like to try different weights on the generator loss terms. Specifically, we want to lower the weight of cycle consistency and identity losses, in order to see whether the adversarial loss would start decreasing after given more importance. We expect the best scenario to be seeing some marginal improvement on the quality of output images.

Secondly, since the CycleGAN model has limited power for handling geometric changes, we want to experiment with models that utilize attention mechanisms. Attention in computer vision helps to find correlation between different layers and within the same layer, therefore, we expect to achieve results with finer and more anime-like details with less artifacts.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[2] 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.

[3] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[4] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019.

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

[6] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[7] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[8] 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.