Avatar Artist Using GAN

Hui Su *(huisu@)*, Jin Fang *(jinf9812@)*

CS230 FINAL PROJECT PRESENTATION, STANFORD UNIVERSITY

Motivation & Objectives

Human stick figures are always very intuitive and abstract. We found solving the problem of transforming human face sketches into anime avatars would be an interesting area. One application could be developing a kid friendly app that records and converts a kid's hand draw sketches into anime or cartoon images. In this project, the input to our algorithm is an image of a human face sketch, and the output would be an anime picture of the face. We use GAN as our algorithms to generate the output. The ultimate goal is that output avatars should look similar with input features and meanwhile look authentic.

Datasets

- Input domain: Danbooru Sketch dataset (Figure 1), Quick Draw (Figure 2)
- Outout domain: Danbooru dataset 2019 (Figure 3), Cartoon Set (Figure 4)

Figure 1

Figure 2

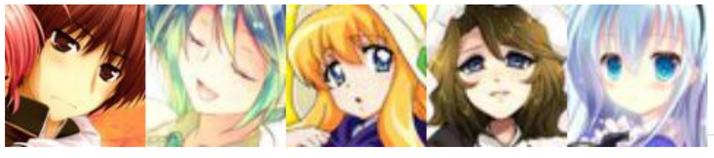


Figure 3

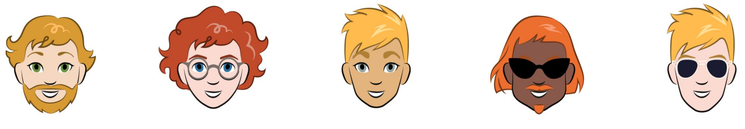
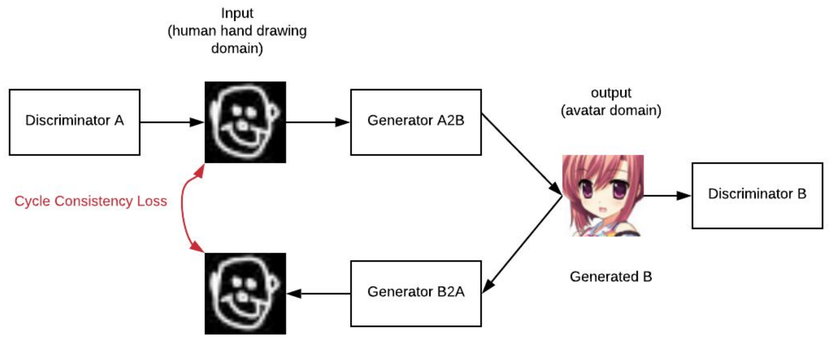


Figure 4

Models

we have explored CycleGAN[1], DiscoGAN[2], and Unit. We will just demo CycleGAN here



In the CycleGAN, we have the Adversarial Loss from X to Y domain. We also have the same loss from another direction.

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

We also use Cycle Consistency Loss to implies that generators should be able to bring x or y back to the original input,

$$L_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} [\| F(G(x)) - x \|_1] + E_{y \sim p_{data}(y)} [\| G(F(y)) - y \|_1]$$

Then we can get the full loss function,

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F)$$

The optimization goal is,

$$G^*, F^* = \min_{G,F} \max_{D_X, D_Y} L(G, F, D_X, D_Y)$$

We use the implementation provided by Erik Linder-Norén. In this implementation, the generators are ResNet. PatchGAN is used in Discriminators.

Expriment 1

- **Cycle GAN** [1] **(**lr =0.0002 , adam_optimizer, n_residual_blocks=9, batch_size = 64, Figure 5 at 20 epochs(about 14700 iteration)

Figure 5

Expriment 2

- **Cycle GAN (**lr =0.0002 , adam_optimizer, n_residual_blocks=9, batch_size = 64, Figure 6 and Figure 7 at around ~40 epochs

Figure 6

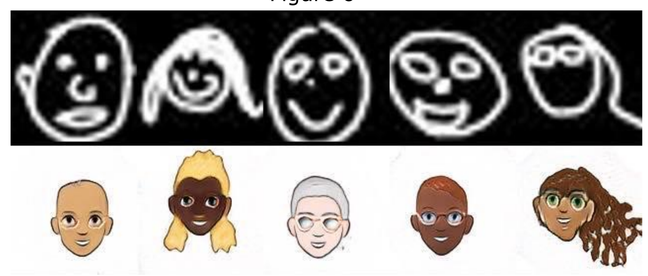


Figure 7

Discussion & Analysis

- Quick draw data's facial features of eyes, nose, etc are underrepresented for majority of the data. Even for human beings, it would be a hard task to image the transformation to cartoon. Find a way to address the problem would improve the output a lot as well

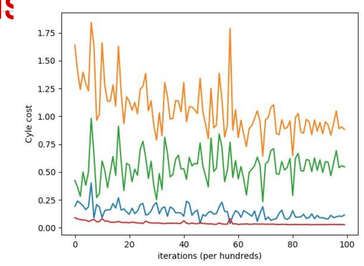


Figure 8 (Orange - average Adversarial Loss, Green - cost of generator, Blue - cost of Discriminator, Red - cost of identity)

Conclusion &Future Work

- We used CycleGAN, DiscoGAN, and Unit. In general, we found CycleGAN is the highest performing
- In both experiments, mode collapse/dropping is a very common issue we found when using GAN to generate domain B images. We addressed mode drop by using mini-batch when updating the discrinatinator. We could continue improving it in the future

References

[1] Almahairi, Amjad, et al. "Augmented cyclegan: Learning many-to-many mappings from unpaired data." *arXiv preprint arXiv:1802.10151* (2018).

[2] Kim, Taeksoo, et al. "Learning to discover cross-domain relations with generative adversarial networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

Video Link: