11-785 @ CMU
Introduction to Deep Learning

CartoonFaceGAN

Changlin Jiang, Xin Shen, Yuyang Wang, Jundong Yi
Carnegie Mellon University

Introduction

In this project, We focus on transferring real-human faces to cartoon style. The application of texture rendering/transfer is currently widely applied to filming industry and picture-editing apps. However, instead of traditional methods in terms of either computer graphics or computer vision algorithms, we approached to this problem with a generative adversarial model based method.

Dataset

To train the model, we use human face https://github.com/NVlabs/ffhq-dataset used in NVIDIA StyleGAN paper as shown in Fig 1. And cartoon face from Cartoon Set: https://google.github.io/cartoonset/ as shown in Fig 2. All images are resized to 286 × 286 and then randomly cropped to 256 × 256, so that all the images have the same size and in a way mitigate overfitting with training data.

Fig 1. Training Data: Human Face

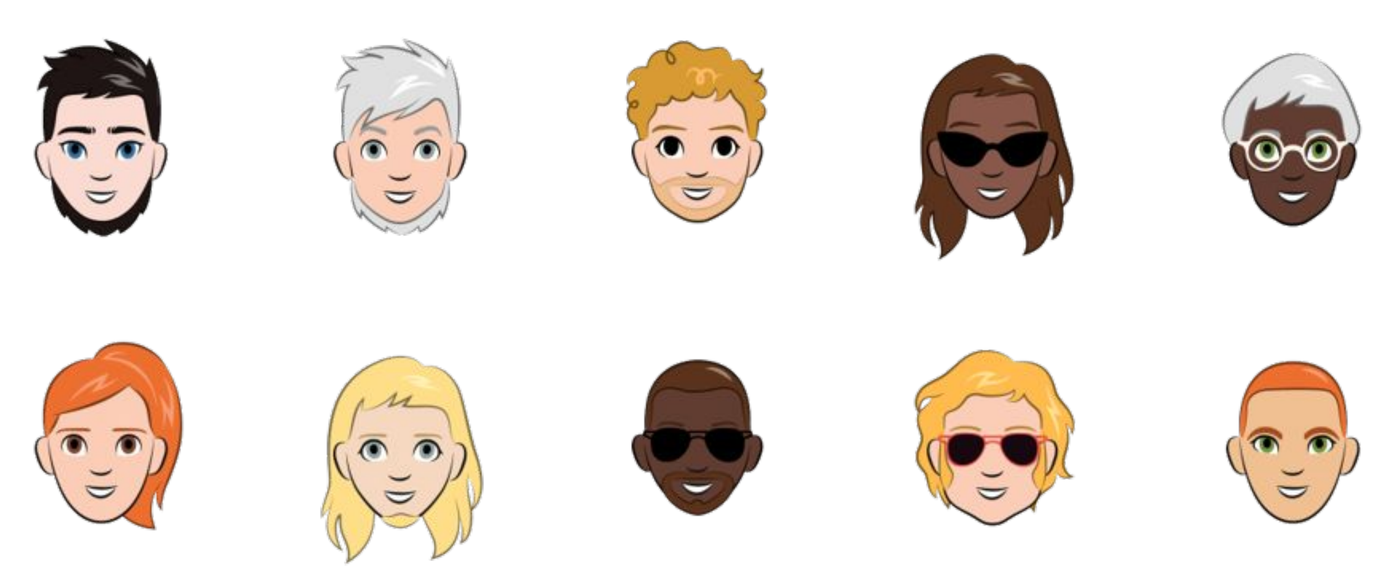

Fig 2. Training Data: Cartoon Face

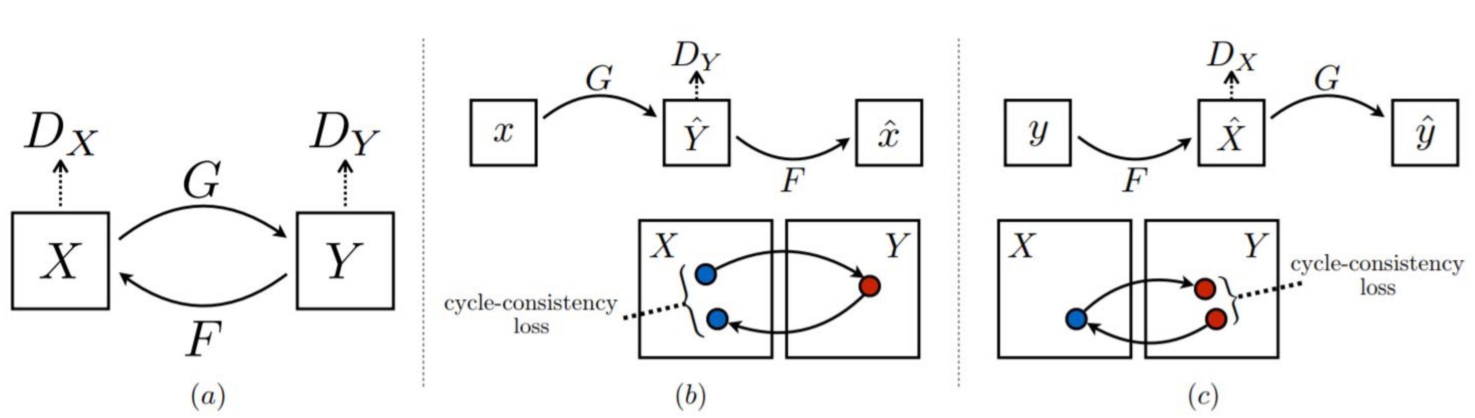

Fig 3. CycleGAN Mechanism

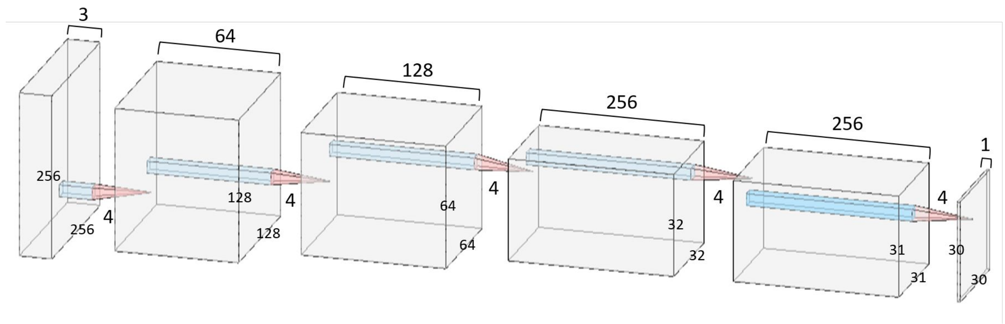

Fig 4. CartoonFaceGAN Discriminator

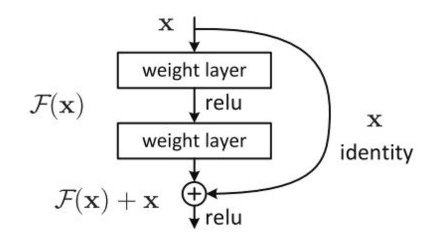

Fig 5. ResNet Block

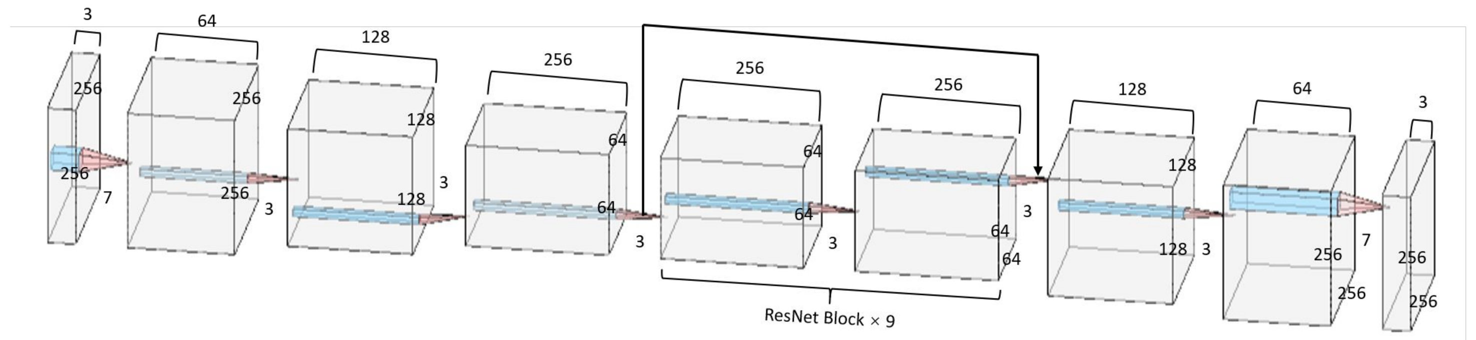

Fig 6. CartoonFaceGAN Generator

CartoonFaceGAN Model & Mechanism

Instead of adopting image-to-image feature transferring, this model is able to apply feature-transferring to the original source twice, similarly to the architecture of CycleGAN as shown in Fig 3. The model consists of two components:

1. Map the original feature to the target feature field;
2. Map it back to the original feature field so that it eliminates the requirement of label- requirements.

The discriminator and generator is shown in Fig 4 and Fig 6 respectively, where 9 ResNet blocks as Fig 5 indicates are used in generator.

The loss function consist of two adversarial losses and one cycle consistency loss. The generators (G, F) try to minimize the adversarial loss while discriminators (Dx, Dy) try to maximize it. The cycle consistency loss is defined to bring generated image back to the original image. The generators and discriminators are optimized alternately using Adam optimizer with learning rate decay and weight decay.

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

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

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

a: Overall loss function
b: Adversarial loss function
c: cycle consistency loss

Fig 8. Image Render Test using CycleGAN

Results

In baseline model with batch size 1 and learning rate of 2e-4 on both G and D, we found it suffered from mode collapse (Fig. 7b). Trying to fix this problem, we tried to increase batch size into 2 and it didn't help (Fig. 7c). Another technique we used was update G and D asynchronized by changing learning rate of D into 9e-5 and this partially helps to get rid of mode collapse (Fig. 7d).

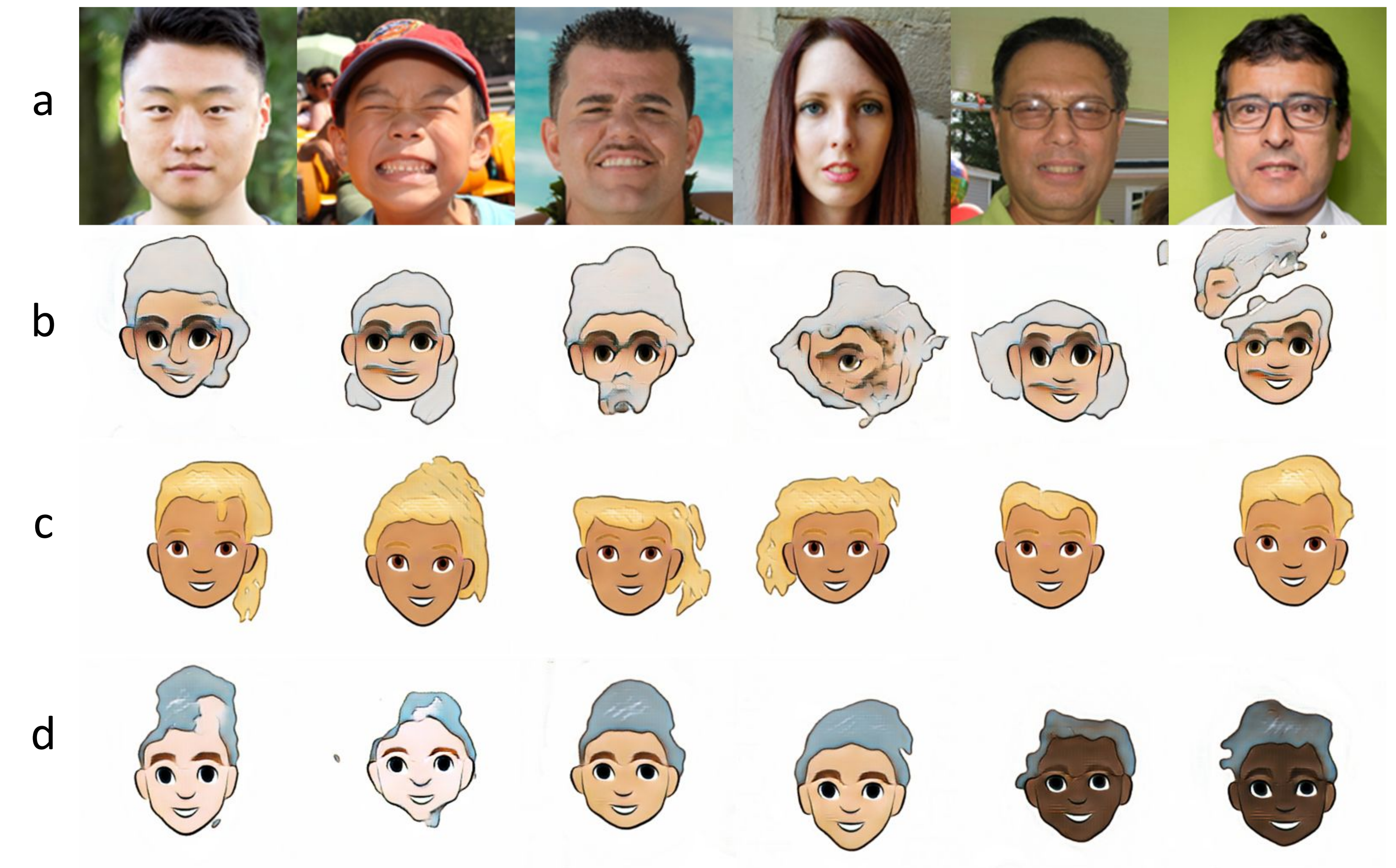

Fig 7. Test Results (a) Test Input (b) Baseline Model Output (c) Batchsize 2 (d) Asynchronous Update

Discussion

- Why modification C, & D
  - From reference [6], choosing mini- batch helps to average out gradients over the batch; individuals will not move along its own internal curve but the overall
  - The key of core cause of collapsing is from Generator being over limited to produces constrained varieties. Thus, we can pre-boost the Generator.
- What we can do in future for improvement?
  - Wasserstein GAN model is quite effective for unsupervised learning for GAN model
- Image Rendering
  - Unsupervised GAN learning is effective for simple rendering

Acknowledgement

- Professor Bhiksha Raj's generous and inspiration teaching
- Mentor William Hu's in-time guidance and help
- Generous knowledge and thoughts from all TAs and CAs

Reference
[1] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." Proceedings of the IEEE International Conference on Computer Vision (2017).
[2] Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." arXiv preprint arXiv:1812.04948 (2018).
[3] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
[4] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
[6] Hoang Thanh-Tung, Truyen Tran. "On Catastrophic Forgetting in Generative Adversarial Networks". 2018