CS230

Generating Artistic Portraits with Different GANs

Kexin Weng
Department of Mechanical Engineering
Stanford University
kexinw@stanford.edu

Zhuzhu Wang
Department of Mechanical Engineering
Stanford University
wangzz@stanford.edu

Abstract

Recent years, many powerful photo/video apps obtain various filters for people to generate different styles of portraits. Yet many of them are too expensive or only provided temporarily. The goal of this project is to learn how the portraits can be translated into cartoons with pix2pix and CycleGAN methods. Modifications are made to the model structures and the generators to see the behavior of each part of the models. We use FID to evaluate the generated results.

1 Introduction

Recent years have witnessed powerful photo/video tools on the popular social media apps (Instagram / Tic Toc) that help people to generate different styles of portraits as a way of expressing their personalities and lifestyles. Yet some of them are pop-ups tools that are temporarily provided or would require additional expensive purchases for permanent usage.

Therefore, in this project we would like to replicate popular commercial filters by training them through powerful GAN models, which enable people to achieve the same goal. The project is divided into the three parts - dataset construction/preprocessing, baseline models exploration, and innovation & improvement of baseline by comparing different model structures. Specifically, we use (1) pix2pix, a conditional generative adversarial network, whose training needs sets of image pairs that are already aligned. (2) CycleGAN, which breaks the requirement of exact pairing and is able to map between input and output datasets that are not aligned.

2 Related work

Zhu et al. [5] proposed an unpaired image-to-image translation model referred as CycleGAN. The model additionally uses an inverse mapping F:Y$\rightarrow$X and uses a cycle consistent loss to push the inverse result F(G(X)) towards X. As a result, the model is able to translate an image from X to Y without paired training data. This method is implemented as our baseline model and will be further discussed in terms of its generator and discriminator structures.

Liu et al. [2] proposed CoGAN, which is another method for unpaired image generation. CoGAN consists of two GAN generators: one for domain X and one for domain Y; the two generators share the latent representations. CoGAN learns a joint distribution for the latent representations to transform X to style Y. This method is concurrent with the CycleGAN method mentioned above and is often used for comparison. One advantage of CoGAN is that it uses less parameters as it partially shares the weights. Similarly, Yi et al. [4] proposed DualGAN, using the same idea for unpaired image-to-image translation. For DualGAN, it uses a primal-dual relation to make images translated from either domain and then reconstructed.

Different from the above two methods, the work of Isola et al. [1] provides a pix2pix method, which learns the mapping from input images to output images, and learns a loss function to train the mapping as well. Pix2pix requires paired images for training. This method is also implemented in our project as a comparison to the CycleGAN model.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

There are also other implementations of the GANs besides image generation and translation. Mukherjee et al. [3] proposed ClusterGAN for clustering using GANs, by sampling latent variables and using an inverse network. With a clustering specific loss, the model achieves clustering in the latent space.

3 Dataset and Features

The Generative Adversarial Network needs two datasets: a content dataset and a style dataset. For the content dataset, we use real face portraits from the "Real and Fake Face Detection" dataset created by Yonsei University. For the style dataset, we manually generate 500 cartoon faces using online cartoon filter platform "Photo To Cartoon", and feed into the real face photos at hand, and manually generate cartoon faces as our style dataset, as illustrated in Figure 10.

3.1 Dataset preprocessing

Since we adopt the pix2pix model and cycleGan model, we then construct the train and test sets respectively, as described in Figure 1.

Pix2pix model requires paired real and styled data, and thus we stitched the styled and real photos together as general format of one input. We downscale the original images (600*600) into (128*128) and stitch the paired image. Hence, the constructed model input has size 256*128 pixels. We then separate all photos into train, validation and test sets following a percentage around 6:1:1.

CycleGan requires no alignment between content and style photos, and thus we randomly select 80% of the images for training, and 20% of the images for testing. We tag real photo as “A”, and styled image as “B”. For each image, we crop and downscale it into 200*200 pixels before feeding it to the model.

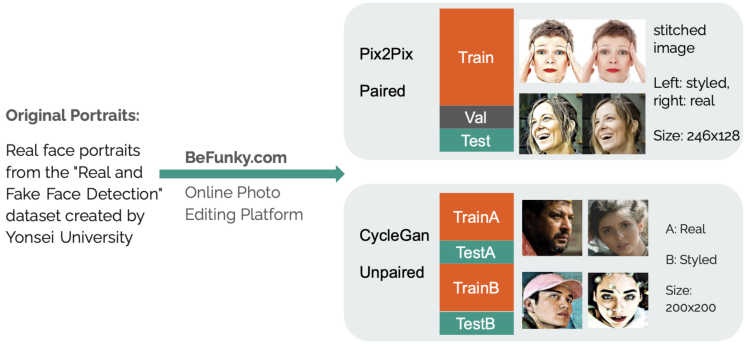


Figure 1: Dataset Construction Process.

3.2 Data Augmentation

We also applied data augmentation techniques, adding flipping and rotation layer, but no obvious improvement seen. We attribute this to the fact that portraits being frontal and align horizontally and remove those layers for faster computation.

4 Methods

4.1 Pix2Pix

The pix2pix model requires aligned data inputs. Pix2pix is a Generative adversarial Network that uses a UNet256 generator and a PatchGAN descriminator. It optimizes over the Gan loss function as described in Equation 1, which incorporates both the conditional GAN loss and a traditional L_1 regularization. Previous work[1] suggests that L_1 performs better in removing the blurring.

$$L_{GAN} = L_{cGAN}(G, D) + \alpha L_{L1}(G) \tag{1}$$

The conditional GAN loss is calculated as in Equation 2, where the x denotes the input image to generator, y denotes the output image of the generator, and z denotes a random noise vector. Engineering the Gan Loss by conditioning the discriminator would help the discriminator more accurately distinguish the matching and alignment of two images. The L_{L1} norm is calculated following Equation 3.

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x, z))] \tag{2}$$

$$L_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x,z)||_1] \tag{3}$$

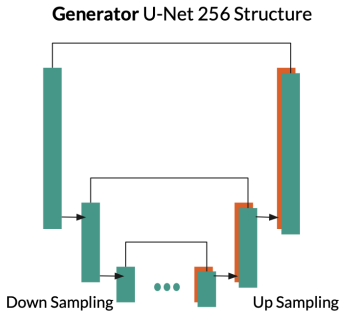
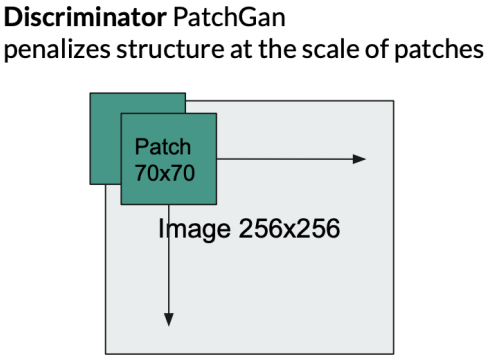
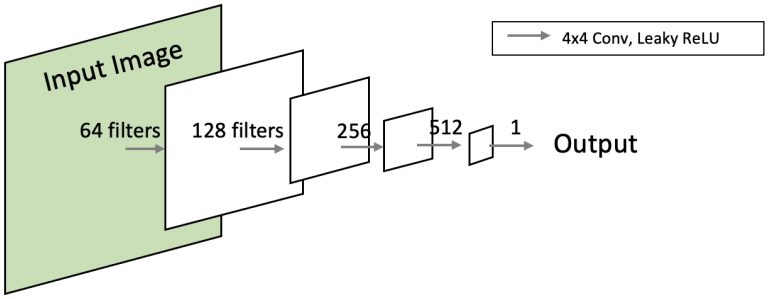


Figure 2: Network architecture for Pix2Pix Model



(a) The effect of the discriminator (PatchGAN)



(b) The layer size and the structure of the PatchGAN

4.2 CycleGan

The CycleGAN model uses two generators and two discriminators to learn the image translation and reconstruction. In our baseline model, we use the 9-block ResNet for the generator, and convolutional networks for the discriminator.

4.2.1 Model Structure

For the baseline model, the architecture of the generator and the discriminator can be seen in Fig. 4 and Fig. **??**, respectively. For the generator, we use a 7x7, followed by 2 3x3 convolutional layers to downsample the input. We use 9 Res-blocks which allow skip connections between the input and output. The ResNet is followed by the up-sampling layers to generate the fake images. Between the layers we use ReLU for activation. For the discriminator, we also use the PatchGAN, which is the same as in the Pix2pix model.

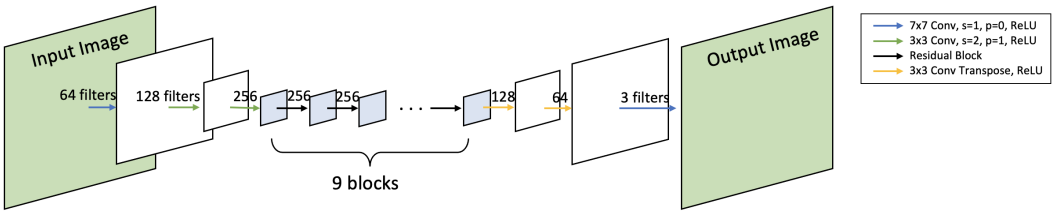


Figure 4: Network architecture for the generator of the CycleGAN model

4.2.2 Loss function

For the loss function, we use the adversarial loss for the two mappings: from portraits to cartoons and from cartoons to portraits. We use the cycle consistency loss to ensure the mapping from an input to a corresponding output with L1-norm. The overall loss is the summation of the two types of losses:

$$L_{GAN} = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \alpha L_{L1}(G, F) \tag{4}$$

where X and Y are the two domains and our goal is to learn the mapping function between them, G is the mapping function from X to Y, F is the mapping function from Y to X, and D_X and D_Y are the two adversarial discriminators aiming to distinguish x from $F(y)$, and y from $G(x)$ respectively. The adversarial loss and the L1 loss are calculated as:

$$L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_y[\log D_Y(y)] + \mathbb{E}_x[\log(1 - D_Y(G(x))] \tag{5}$$

$$L_{L1}(G, F) = \mathbb{E}_x[||F(G(x)) - x||_1] + \mathbb{E}_y[||G(F(y)) - y||_1] \tag{6}$$

We also use the Adam optimization for the model.

5 Experiments & Results

5.1 Evaluation Metrics

We adopt qualitative and quantitative methods to analyze the results of the model. Qualitatively, we inspect the generated "fake_B" image vs. the desired "real_B" image on aspects of color, texture, etc. To reduce bias, we use the Frechet inception distance (FID), a typical metric to assess the generated images of GANs. FID compares the difference between the generated cartoons and the groundtruth.

5.2 Pix2Pix Model Experiment Results

5.2.1 Experiments on Network Structures - UNet256, UNet 128, ResNet

We experiment on the network structures of Pix2Pix Gan models. We replace the orginal UNet256 with UNet128 and ResNet-6-Blocks. A typical set of output images are displayed in Figure 5.

FID for UNets have smaller values than that of ResNet-6-block model. Observing from the generated stylish portraits with the underlying ground truth Real B, UNets are more powerful in predicting features like smoothed texture, and the image is a bit whitened. ResNet generates images with more noises with a yellowish image. Results for UNet256 and UNet128 are very similar in terms of image quality, and FID difference is very small and negligible compared with random noises in training.

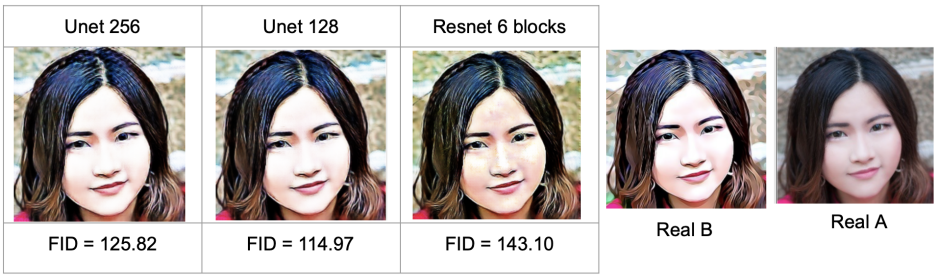


Figure 5: Results for Pix2Pix network structure experiments

5.2.2 Experiments on Initialization Methods - Normal, Xavier, Kaiming, Orthogonal

We experiment on different initialization methods in running Pix2Pix Gan models. Specifically, we replace the original random normal distributions with Xavier, Kaiming, and orthogonal initialization. A typical set of output images are displayed in Figure 6. Observing from generated images, the qualities of normal, Xavier and orthogonal initialization are similar, with smooth texture and color patterns very close to the ground truth style image. Kaiming initialization has seen a mesh texture with noise, and the color pattern is biased towards the content image. PID for Kaiming is largest among the four initialization methods. Random Normal method overall has the most favorable performance.

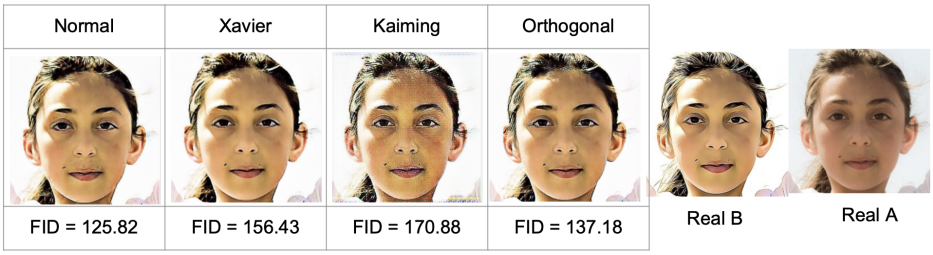


Figure 6: Results for Pix2Pix network initialization experiments

5.3 CycleGan Model Experiment Results

We train the baseline CycleGAN model for 20 epochs with learning rate equal to 0.0002. We then add a batch normalization and used Kaiming initialization; and finally we change the ResNet to UNet-256 to observe the behavior of the model. The results are shown and compared in 9.

From the result, we see that Kaiming initialization and batch normalization does not make much difference from the baseline, but both of them performs better than the Unet in terms of FID. Though quantitatively we can observe that UNet generates very cartoon-like portraits, qualitatively, the FID is lower. This may be because the Cycle GAN model generates closer light conditions as in the ground-truth cartoon images.

For now, we have trained our model for 20 epochs with learning rate equal to 0.0002 to observe the behavior. Figure 10 shows the loss and the weighted moving average loss for the generators and the discriminators. The losses for the discriminators have already shown a decreasing trend, while we may need more epochs or parameter tuning for the generator in the future to show a better trend.

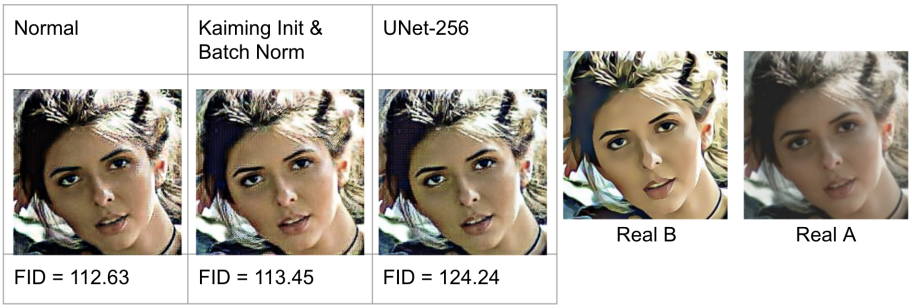


Figure 7: Results for CycleGAN network

6 Discussion

In this project, we study the baseline model of the Pix2Pix and CycleGan, and we develop different networks structures in combination with different initialization methods as an improvement for the baseline model. Comparing the two models, pix2pix trains much faster than the CycleGan. We attribute this to the fact that the model absorbs more information in the paired datasets. With a similar number of epochs, CycleGan generates images with lower FID, yet most of the them display grid textures and noises.

In improvement stage, we found that different initialization methods influence the quality of images, and different model structures have distinct advantages. UNet is good at learning features, and ResNet is good at predicting lights condition, making ResNet have lower FID.

In this project, we enjoy the process of exploring different techniques and network structures. We also meet our initiative of replicating commercial artist filter - generation of Kexin and Zhuzhu's Portrait is available in Appendix.

7 Contributions

Team member inlcudes Zhuzhu Wang (wangzz@stanford.edu) and Kexin Weng (kexinw@stanford.edu). Their contributions are stated as follows.

- Zhuzhu Wang:
  1. Searched and selected suitable content dataset for the project.
  2. Applied data preprocessing / Augmentation to the datasets for CycleGan model.
  3. Implemented baseline for CycleGan.
  4. Explored improvement for CycleGan on structures and initialization methods.
  5. Coded for evaluation metrics for final results analysis.
- Kexin Weng:
  1. Searched plausible tools for artistic portraits, and generated stylish datasets.
  2. Applied data preprocessing to stitch up the images for the pix2pix model datasets.
  3. Implemented baseline for pix2pix.
  4. Explored improvement for pix2pix on structures and initialization methods.
  5. Coded for loss visualization, etc. for deliverables.

References

[1] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: *CVPR* (2017).

[2] Ming-Yu Liu and Oncel Tuzel. "Coupled Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/502e4a16930e414107ee22b6198c578f-Paper.pdf.

[3] Sudipto Mukherjee et al. *ClusterGAN : Latent Space Clustering in Generative Adversarial Networks*. 2019. arXiv: 1809.03627 [cs.LG].

[4] Zili Yi et al. "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2868–2876. DOI: 10.1109/ICCV.2017.310.

[5] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss". In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.

Appendix

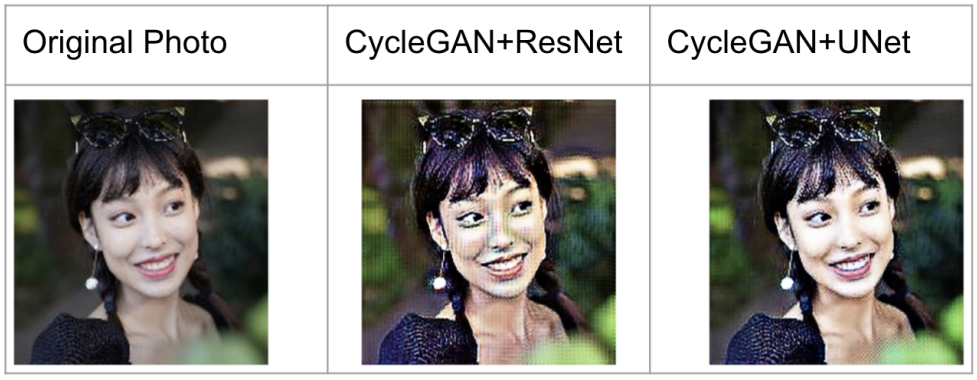
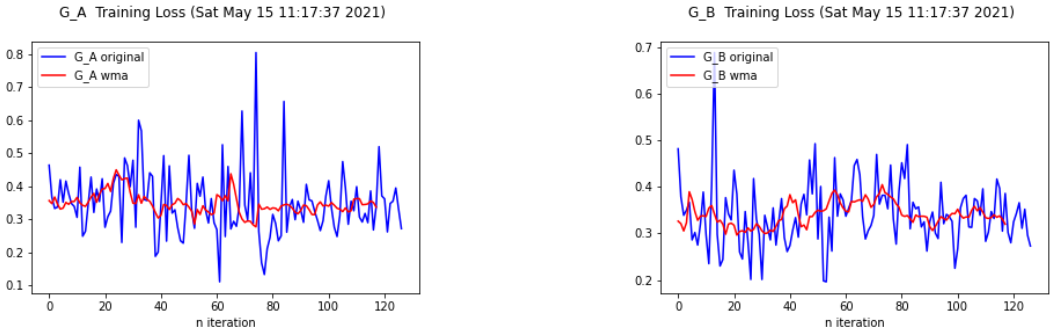


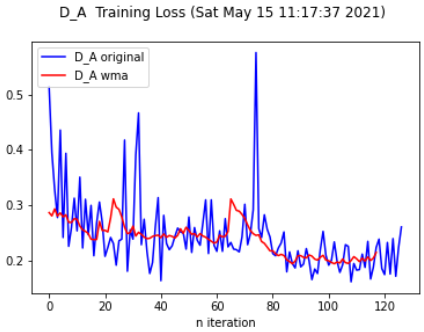
Figure 8: Test for our own photos on CycleGan

Figure 9: Test for our own photos on Pix2Pix

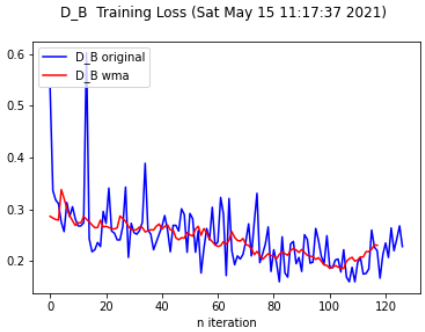


(a) G_A training loss

(b) G_B training loss



(c) D_A training loss



(d) D_B training loss

Figure 10: Training loss for baseline model.