Generative Adversarial Networks for Van gogh-style Image Generation

Fengze Zhang, Yihou, Chenhao Zhang

(yh4858, fz2244, cz2632 NYU Tandon)

Abstract

We use different methods to implement the CycleGAN model and train them on our dataset to generate Van gogh-style images. Then we compared the result between different model frames and From a theoretical perspective, an explanation and analysis were provided for the observed issues and phenomena.

Problem Description

In our project, our objective is to train a deep learning model that can transform an input photograph into the style of Vincent Van Gogh. Additionally, we require objective quantitative evaluation metrics to assess the output results of our model.

Literature Survey

In the face of a typical unsupervised learning problem as image style transfer, we directed our focus towards the widely acclaimed and extensively applied family of generative models known as Generative Adversarial Networks (GANs). We aimed to investigate multiple networks within the GAN family and ultimately select one as the foundational architecture for our project. There are various models within the GAN series that are applied to image generation. These models range from the initial application of GANs to image processing with DCGAN (Deep Convolutional GAN), to Pix2pix, which is based on CGAN (Conditional GAN), and the combination of GANs with diffusion models in Diffusion GAN, among others. Among the various models utilized for image generation within the GAN series, several caught our attention due to their simultaneous emergence and consistent conceptual ideas. These models include CycleGAN, DualGAN[1], and DiscoGAN.

These three models independently employ the idea of duality, effectively addressing the common issue of mode collapse prevalent in traditional GANs by establishing bidirectional mappings. This aligns perfectly with our project's objectives. The images in our dataset originate from two distinct image domains that lack a one-to-one correspondence. The aforementioned models excel at addressing this problem, overcoming the limitations of Pix2pix, which requires explicit correspondence between image pairs. Through comparative analysis and studying relevant papers, we discovered that CycleGAN achieved the most promising results in image style transfer, as evidenced by both quantitative metrics and subjective evaluations by volunteers. Consequently, we chose CycleGAN as the foundational framework for our model.

CycleGAN

1. Model Architecture:

CycleGAN leverages a generator-discriminator setup typical in GAN architectures. There are two sets of generators and discriminators: one for mapping from domain X to domain Y (G: X → Y and Dy), and one for mapping from Y to X (F : Y → X and Dx). The generators Gx_y and Gy_x are responsible for translating images from one domain to another, while the discriminators Dy and Dx aim to differentiate between translated and real images in their respective domains.

2. Loss Functions:

CycleGAN uses a combination of adversarial losses and cycle consistency losses. Th4e adversarial loss ensures that for each domain, the generator fools the discriminator into thinking the generated images are real. The detector considers the generated picture to be real. This is true for every domain For the mapping from X to Y, the generator G tries to generate fake y that can fool the discriminator Dy. Similarly, for the mapping from Y to X, the generator F tries to generate a fake x that can fool the discriminator Dx.

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

The cycle consistency loss ensures that an image, when translated from one domain to another and then translated back, should remain the same. This enforces the notion that

the learned mappings G and F should be consistent with each other. The cycle consistency loss is defined as:

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

loss encourages F(G(x)) ≈ x and G(F(y)) ≈ y.

3. Hyperparameter Selection:

The selection of hyperparameters in CycleGAN requires careful tuning. Typically, the learning rate is set to 0.0003, and the Adam optimizer. The hyperparameter λ is used to control the trade-off between adversarial loss and cycle consistency loss. If λ is set too large, the cycle consistency loss will dominate the optimization process, which may result in images that cannot "fool" the discriminator sufficiently. On the contrary, if λ is set too small, the adversarial loss will dominate the optimization process, which may cause the generated image not to be preserved after round-trip transformation. In the current training, we choose the λ=10, but in the actual process, we found that it may be more natural to transfer the picture to Van Gogh's painting style by reducing λ and increasing Adversarial Loss.

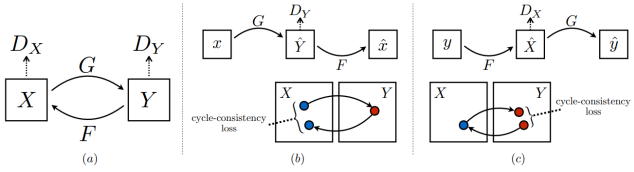

Figure 1: (a) shows the overall architecture, (b) and (c) show the unidirectional architectures)

Implementation

In our project, we implemented the Cycle GAN structure according to [2] and its official codebase[2]. Described in [2], generators are of ResNet structures with residual blocks. Therefore, in our first version of implementation (**model 1**), we reused the ResNet we implemented in our mid-project (with code refactors). The net of generators first extracts features by increasing channels, then goes through several residual blocks and eventually gathers information when decreasing channels to the normal photo level to reconstruct new photos. During the whole generator process, the size of images remains the same. Noticing that U-net is known as a special kind of ResNet for its power to extract features and used as generators in Dual GAN, we also implemented a U-net version of generators (**model 2**).

For the discriminator itself, it always works as a classifier telling between the real pictures and fake ones, so its mission is just a 2-classification problem, which is enough for the traditional CNN to work on. Therefore, we didn't modify discriminators.

In addition, when we tried to speed up the training process, we came to the idea of cutting the bi-directional structure to a single-directional structure. Since our mission here is just a single direction transform from real photos to Van Gogh-style pictures, we may reduce a discriminator, training only the discriminator for the X to Y transforms as shown in Figure 1 (b). So our third version of the implementation is the single-directional GAN with ResNet-based generators (**model 3**).

We trained our models on a vangogh2photo dataset provided by [2]. This dataset consists of 401 pictures of Van Gogh's downloaded from Wikiart. In this dataset, real photos are 731 unrelated random pictures. All of these pictures are 256*256 pixels. After data argumentation, we have deviated and noised data using various transforms.

Due to the vast cost of calculation resources, we chose to train each of the three models for 20 epochs. In each epoch, We first use the Y-style image on G (X -> Y) and the X-style image on F (Y -> X) to generate Y_fake and X_fake respectively. In our mission, X equals the domain of real photos and Y the domain of Van Gogh's pictures. Loss between them and the original images is calculated so that we can conclude that it is possible that Y and Y_fake are similar in style. Next, we use the X image to generate Y_fake, and Y to generate X_fake, use their respective discriminators to identify the fake image, and get the confrontation loss. Then reconstructed images are built using the contrary generators, and the consistency loss is calculated to ensure that the gap between X and restored X should be very small. After that, we add the loss obtained by adding the weights, and perform backpropagation to follow the new parameters.

During the training process, we sample and save pictures at every 100 intervals, which are used for later comparison and determination of model qualities. The sample result and loss changes are shown below.

Figure 2: Sample results after epoch 5 of model 1. The first row is real pictures and the second row is the transform result to a Van Gogh-style picture. The third row is a Van Gogh picture and the last row is its transformation to real picture

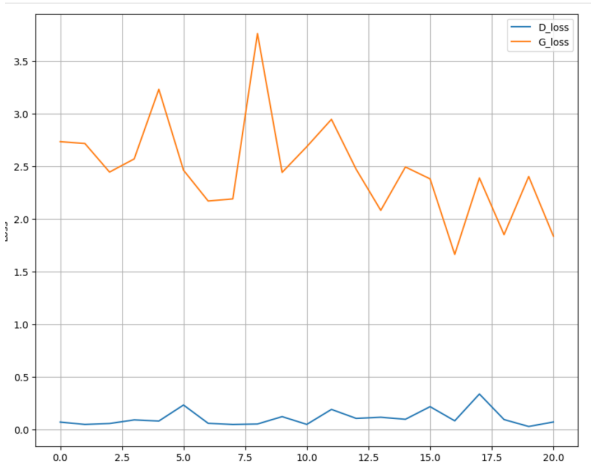

Figure 3: loss plot for model 1

Figure 4: Sample results after epoch 5 of model 2. The first row is real pictures and the second row is the transform result to a Van Gogh-style picture. The third row is a Van Gogh picture and the last row is its transformation to real picture

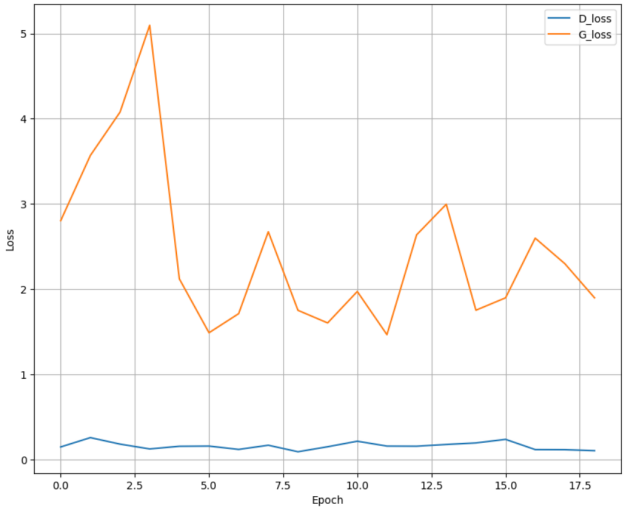

Figure 5: loss plot for model 2

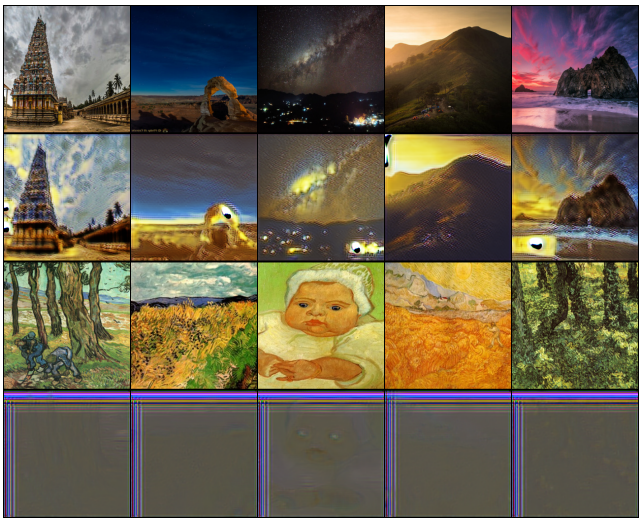

Figure 6: Sample results after epoch 5 of model 3. The first row is real pictures and the second row is the transform result to a Van Gogh-style picture. The third row is a Van Gogh picture. Due to the deletion of the second discriminator, the last row outputs nothing.

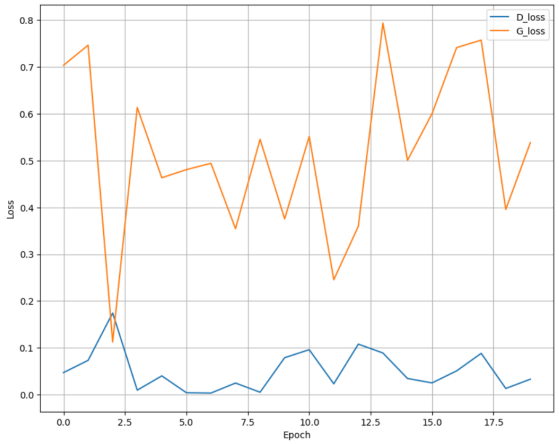

Figure 7: loss plot for model 3

Result

In this section, we employed several objective quantitative evaluation metrics to compare our three different models.

1 IS(Inception Score)

For image generation models, we first need to assess the generated images based on two aspects: 1) image clarity and completeness of details, and 2) overall diversity of the generated images. To address these points, we selected the Inception Score (IS) as the first evaluation criterion. In the IS metric, the generated images are fed into the Inception V3 network to obtain a 1000-dimensional vector.

$$\mathbf{IS}(G) = \exp\Big(\frac{1}{N}\sum_{i=1}^{N} D_{KL}\left(p(y|\mathbf{x}^{(i)})||\hat{p}(y)\right)\Big)$$

However, IS only considers the generated images themselves and does not establish a connection with real data. In other words, IS alone cannot reflect the authenticity of the data.

2 FID (Fréchet Inception Distance)[5]

To address this issue, we chose to use the Fréchet Inception Distance (FID) as the second evaluation criterion. FID calculates the distance between feature representations of two sets of images or two image domains. The calculation formula for FID is as follows:

$$\mathrm{FID} = ||\mu_r - \mu_g||^2 + Tr\big(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\big)$$

When the generated images closely resemble the real images in terms of their features, the mean and covariance will be smaller, resulting in a lower FID value. A smaller FID indicates a better capability of the model to generate images that are similar to the real images.

The following table shows the IS and FID metrics of the test result of three models we implemented :

	IS_mean	IS_STD	FID
model 1	1.6536	0.1516	118.58
model 2	1.7316	0.1555	168.52
model 3	1.7408	0.1343	134.41

Table 1: Table for the result metrics

The following images show the test result of the three models we implemented:

a. model 1:

b. model 2:

c. model 3:

As we can see, the images generated by model 1 are the most vibrant as well as lifelike. Model 1 also performes best on all the metrics. Model 2 demonstrates the lowest level of learning in generating images, as visually observed, it does not quite align with the style of Van

Gogh's paintings.The FID score for Model 2 also supports this observation, indicating a larger discrepancy between the generated images and the real images in terms of their style representation. Model 3 also produces images with good quality and visual appeal. However, in some images, there are circular blurred patches resembling mosaic-like artifacts.

Analysis and Conclusion

After replacing the generator part of the model with Unet, we experienced a significant decline in performance, which has left us puzzled. This is unexpected because Unet is an improvement based on fully convolutional networks and is known for its strong ability to extract image features. Upon reflection, we believe that the possible reason behind this phenomenon lies in the fact that image style transfer and similar image generation problems are not primarily concerned with local features. As mentioned in the original paper of the CycleGAN model, these models do not perform pixel-level processing on the images. Therefore, we have reason to believe that when dealing with such problems related to image style transfer, it is crucial to emphasize the common global features of image domains and the relationship between the images, rather than solely focusing on local features.

As for the appearance of blurry patches after removing the reverse cycle from the original CycleGAN model, our initial hypothesis was based on the limited computational power of our personal computer's GPU and the Colab cloud platform, which resulted in long training times. We hoped that removing the reverse cycle would expedite the training process. Initially, we believed that since we only desired one-way generation from photos to Van Gogh-style images, the reverse cycle process would not contribute significantly to improving the quality of our target images. However, the emergence of blurry patches prompted us to reconsider the issue.

In the original paper, it is mentioned that one of the key innovations of CycleGAN, compared to traditional GANs, is the introduction of cycle consistency loss[2]. Therefore, we believe that the bidirectional cycle structure should serve as a manifestation of the model's dual consistency. By removing half of the structure, the ability to achieve bidirectional mapping is lost, resulting in local image patterns experiencing mode collapse.

The presence of blurry patches in the resulting images might be attributed to approximate fixed points in the transformation due to mode collapse caused by the removal of the reverse cycle.

Codebase

Our codes are on the github link:
https://github.com/DemoySegment/CycleGAN

Reference

[1] Yi, Zili, Hao Zhang, Ping Tan, and Minglun Gong. "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation." In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2868-2876. 2017.

[2] Zhu, Jun-Yan, et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks." Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.

[3] Zhu, Jun-Yan. "PyTorch-CycleGAN-and-pix2pix." GitHub, 2021, github.com/junyanz/pytorch-CycleGAN-and-pix2pix.

[4] Goodfellow, Ian, et al. "Generative Adversarial Networks." Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS). 2014.

[5]Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium." Advances in Neural Information Processing Systems (NeurIPS). 2017.

[6]Nowozin, Sebastian, Botond Cseke, and Ryota Tomioka. "The GAN Landscape: Losses, Architectures, Regularization, and Normalization." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8971-8980. 2018.