Improved CycleGAN for Image-to-Image Translation

Weining Hu[*]
weining.hu@stat.ubc.ca

Meng Li [†]
meng.li@alumni.ubc.ca

Xiaomeng Ju[‡]
xiaomeng.ju@stat.ubc.ca

Abstract

CycleGAN [1] is one recent successful approach to learn a mapping from one image domain to another with unpaired data. We investigated CycleGAN as a solution to artistic style transfer, in particular, translating photographs to Chinese paintings. To improve the stability of training, we improved CycleGAN based on Wasserstein generative adversarial network (WGAN) and further improved WGAN with gradient penalty. The performance of CycleGAN, CycleWGAN, and Improved CycleWGAN are compared on a self-collected dataset CNPaintings both quantitatively and qualitatively.

1. Introduction

Image-to-image translation is the task of learning a mapping from images of one domain to another domain. Such translation is useful for multiple purposes: e.g. converting gray scale images to colorful images [1–4], transforming sketches to realistic images [1, 3, 5], and mapping the face of one person to another [6] etc. In particular, we are interested in *artistic style transfer*, recomposing images in the artistic style of some other images.

We are interested in transforming natural images into Chinese paintings with a self-collected dataset. This task belongs to artistic style transfer applications, but has its own challenges. Generating Chinese painting style images can be difficult due to their semi-transparent layers, lack of background color, and irregular shaped objects. We aim to build a model that produces images that look like real Chinese paintings given inputs of natural images. Our model can also be applied to video translation, making Chinese paintings come to life.

For the training architecture, we first implemented CycleGAN which is the state-of-the-art method for image-to-image translation with unpaired data. However, it is known that *General Adversarial Network* (GAN) methods usually suffer from the problem of unstable training. Typically, a GAN consists of two networks: a generator network and a discriminator network. The generator network produces synthesized samples given some input noise, and the discriminator network will try to distinguish between the real data and the generated data. The discriminator network distinguishes the real data distribution from the generated data distribution with some distance measures such as *Jensen-Shannon* (JS) divergence. However, if the two distributions do not have substantial overlap, the gradients can point to random directions, resulting in unstable training [7].

To improve the stability of training, we develop CycleWGAN based on a recent work of Wasserstein generative adversarial network (WGAN) [8]. WGAN is constructed with Wasserstein distance. Compared to other distance measures, Wasserstein distance provides a meaningful and smooth representation of the distance between two distributions, even when they are located in lower dimensional manifolds with little overlaps [8]. We further consider an Improved CycleWGAN that replaces weight clipping in WGAN with gradient penalty. This improvement is motivated by the work of Gulrajani et al, who pointed out that gradient penalty further eliminates mode collapse and avoids optimization difficulties. [9].

In this project, we present the Chinese painting style transfer as a problem that learns the mapping from existing images of natural flowers to Chinese paintings. We propose different methods to solve the problem. Overall, we plan to contribute in the following aspects:

- Implement CycleGAN to transform natural images into Chinese painting style images.
- Develop CycleWGAN and Improved CycleWGAN to address the problems of unstable training and mode collapse in CycleGAN.
- Demonstrate the strengths of our proposed meth-

[*] Department of Statistics, University of British Columbia

[†] Department of Electrical and Computer Engineering, University of British Columbia

[‡] Department of Statistics, University of British Columbia

ods on our self-collected CNPaintings dataset for both model stability and image quality.

2. Related Work

2.1. Image-to-image translation

Image-to-image translation has recently become a trending topic in academic researches and industrial applications. Early image-to-image translation problems were tackled with separate context-specific approaches [10–14] despite of the similarity in their settings. *Pix2pix* [3] was developed as a general framework to solve the problem of image-to-image translation. It relies on the paired structure of data to form a "U-net" architecture based on *conditional GAN* (cGAN). However, obtaining paired training data is difficult or impossible in many image-to-image translation tasks. Shortly after, CycleGAN was proposed to train with unpaired samples [1]. It makes a two-step transformation of data from domain X to Y and then back to X, constructing two generators and discriminators accordingly. CycleGAN is powerful in generating realistic images with unpaired data, and thus is chosen as our baseline model.

2.2. Artistic style transfer

Artistic style transfer is an application of image-to-image translation. Inspired by the power of deep neural networks, Gatys et al [15] first studied how to transform natural images to famous painting styles with *Convolutional Neural Network* (CNN). They proposed to model the content and style of an image separately and recombine them to produce artistic images of high perceptual quality. The key idea behind their algorithm is to model the content of an artwork as a combination of features from a pre-trained CNN, and the style as its texture information by computing the correlation between CNN features. This pioneer work has attracted wide attention in the academic community and many subsequent studies had been proposed to extend or improve this model.

Gatyset at al. [15] 's model is based on iterative optimization and thus is computationally expensive with limited scalability. To alleviate this issue, many works proposed to directly learn feed-forward generator for a given style. Some of the fast methods include [16–18]. Among many artistic style transfer models, we focus on conditional Generative adversarial networks (cGANs), which automatically learn a loss function that tries to classify if the output image is real or fake, while simultaneously training a generative model to minimize this loss [19]. Images generated by GAN are usually perceived to be very realistic and less blurry compared to CNN-based models.

2.3. GAN and stable training

It is well-known that GAN-based models suffer from instable training and mode collapse [7–9, 20, 21]. We hope to address this issue of CycleGAN and improve the quality of generated images. In literature, a number of alternatives have been proposed to achieve stable training with GAN-based models, including least squares GAN [22], energy-based GAN [23], deep regret analytic GAN [7], and WGAN [8]. In this project, we will focus on WGAN and its following work of Improved WGAN [9]. WGAN enables stable training with weight clipping, and Improved WGAN introduced weight regularization into the framework [8, 9].

2.4. Chinese painting style transfer

To the best of our knowledge, there exist two related works that applied style transfer to Chinese paintings. Chen et al. [24] applied cGANs, DCGANs, WGANs and modified WGANs to their self-collected Chinese painting dataset. They trained models with inputs of Chinese paintings together with their self-converted sketches of the these paintings. However, their model is based on paired data and thus a Chinese painting sketch is needed for prediction. Besides, the oscillating behavior of their discriminator loss remained unexplained. Lin et al [25] proposed a deep multiscale deep neural network based on vanilla GAN to transform sketches into Chinese paintings. Their work is also based on paired data with the sketches extracted by the author. Besides, some of their predictive results generated from sketch images do not look quite realistic compared to paintings produced by artists.

3. Methodology

In this section, we will first introduce the CycleGAN, describe WGAN and Improved WGAN, and then present our proposed CycleWGAN and Improved CycleWGAN.

3.1. CycleGAN

To introduce CycleGAN, we first define two types of mapping: $G : X \to Y$ and $F : Y \to X$. The discriminator D_X distinguishes between x and $F(y)$, and D_Y distinguishes between y and $G(x)$. Our objective is to learn a mapping between a source domain X to a target domain Y given the training samples: $\{x_i\}_{i=1}^{n}$, $x_i \in X$ and $\{y_j\}_{j=1}^{m}$, $y_j \in Y$, with distributions $x \sim p_X(x)$ and $y \sim p_Y(y)$

CycleGAN introduces the idea that "if we translate from one domain to another and back again we should arrive where we started" [1]. The objective function of CycleGAN consists of two types of loss: *adversarial loss* and *cycle consistency loss*. Adversarial loss evaluates the distance between distribution of generated images and the real images. Cycle consistency loss enforces the $F(G(X)) \approx X$, and $G(F(X)) \approx y$.

In CycleGAN, we have two adversarial losses:

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

and

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) &= \mathbb{E}_{x\sim p_X(x)}[\log D_X(x)] \\ &+ \mathbb{E}_{y\sim p_Y(y)}[\log(1 - D_X(F(y))].\end{aligned} \tag{2}$$

The cycle consistency loss is defined by

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) &= \mathbb{E}_{x\sim p_X(x)}[||F(G(x)) - x||_1] \\ &+ \mathbb{E}_{y\sim p_Y(y)}[||G(F(y)) - y||_1].\end{aligned} \tag{3}$$

Combining the adversarial losses and the cycle consistency loss, we obtain the full objective function:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) &= \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &+ \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda\mathcal{L}_{\text{cyc}}(G, F),\end{aligned} \tag{4}$$

where λ controls the relative importance of the cycle consistency loss. In the training phase, the parameters in G, F, D_X, and D_Y are estimated by optimizing the full objective function and we get

$$G^*, F^* = \arg\min_{G,F}\max_{D_X,D_Y} \mathcal{L}(G, F, D_X, D_Y). \tag{5}$$

3.2. WGAN and Improved WGAN

GAN has achieved great success in generating images that are perceived to be real, however, they are often hard to train and suffer from training instability. The newly proposed Wasserstein GAN (WGAN [8]) makes progress towards those issues.

Arjovsky et al. [8] stated that training difficulty of GANs is due to the poor design of the loss function. Many commonly used loss functions used in GAN, such as JS divergence, are local saturated, causing the problem of vanishing gradients. Therefore, they proposed Wasserstein distance which has preferable continuity and differentiability properties.

Let the distribution of the real images and the generated images be $\mathbb{P}_r$ and $\mathbb{P}_g$ respectively. The Wasserstein distance between $\mathbb{P}_r$ and $\mathbb{P}_g$ is defined as

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{|f|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[f(\tilde{x})], \tag{6}$$

where the supremum is taken over all the 1-Lipschitz functions $f : \chi \to \mathbb{R}$ and χ is a compact metric space. In the context of GAN, the function f corresponds to the discriminator $D(x)$ and the objective function of WGAN becomes:

$$\min_G \max_{D\in\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})], \tag{7}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions and $\mathbb{P}_g$ is the model distribution defined by $\tilde{\boldsymbol{x}} = G(\boldsymbol{z}),\ \boldsymbol{z} \sim p(\boldsymbol{z})$, where $p(\boldsymbol{z})$ is some simple noise distributions such as uniform or Gaussian distribution. The 1-Lipschitz constraint on the discriminator is achieved by clipping the weights of discriminator to lie within a compact space $[-c, c]$. Compared to the original GAN, WGAN has the following changes:

- Clip the weight of D
- Remove log term in the loss
- Remove the sigmoid at the output of D
- Use RMSProp instead of ADAM

In a more recent work, Gulrajani et al. pointed out that weight clipping is sensitive to the choice of c and its performance is unsatisfactory in some cases [9]. They introduced Improved WGAN with *gradient penalty* (*GP*) as an alternative to weight clipping. The Improved WGAN, also termed as WGAN-GP, penalizes the norm of gradient of discriminator yielding an objective function:

$$\begin{aligned}\mathcal{L}(G, D, X) = &\underbrace{\mathop{\mathbb{E}}_{G(\boldsymbol{z})\sim\mathbb{P}_g}[D(G(\boldsymbol{z}))] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} \\ &+ \underbrace{\lambda_{\text{GP}} \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}[(||\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})||_2 - 1)^2]}_{\text{Gradient penalty}}.\end{aligned} \tag{8}$$

λ_{GP} is the penalty coefficient. $\mathbb{P}_{\hat{\boldsymbol{x}}}$ is the sampling distribution that uniformly samples along straight lines between pairs of points sampled from the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$ [9]. This method performs better than the standard WGAN and achieves stable training on a wide variety of GAN architectures.

3.3. Proposed Model

To improve the training stability of CycleGAN, we extended the idea of WGAN and Improved WGAN to CycleGAN.

3.3.1 CycleWGAN

The proposed adversarial losses for CycleWGAN are

$$\begin{aligned}\mathcal{L}_{\mathrm{WGAN}}(G, D_Y, X, Y) = {} & \mathbb{E}_{y\sim p_Y(y)}[D_Y(y)] \\ & - \mathbb{E}_{x\sim p_X(x)}[D_Y(G(x)],\end{aligned} \tag{9}$$

$$\begin{aligned}\mathcal{L}_{\mathrm{WGAN}}(F, D_X, Y, X) = {} & \mathbb{E}_{x\sim p_X(x)}[D_X(x)] \\ & - \mathbb{E}_{y\sim p_Y(y)}[D_X(F(x)].\end{aligned} \tag{10}$$

Combined with cycle consistency loss, our full objective for CycleWGAN is:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = {} & \mathcal{L}_{\mathrm{WGAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\mathrm{WGAN}}(F, D_X, Y, X) \\ & + \lambda_0 \mathcal{L}_{\mathrm{cyc}}(G, F).\end{aligned} \tag{11}$$

3.3.2 Improved CycleWGAN

By introducing gradient penalty, we present an Improved CycleWGAN, also called CycleWGAN-GP, with an objective function:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = {} & \mathcal{L}_{\mathrm{WGAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\mathrm{WGAN}}(F, D_X, Y, X) \\ & + \lambda_0 \mathcal{L}_{\mathrm{cyc}}(G, F) \\ & + \lambda_1 \underset{\hat{\boldsymbol{x}}_1\sim\mathbb{P}_{\hat{\boldsymbol{x}}_1}}{\mathbb{E}}[(||\nabla_{\hat{\boldsymbol{x}}_1} D_Y(\hat{\boldsymbol{x}}_1)||_2 - 1)^2] \\ & + \lambda_2 \underset{\hat{\boldsymbol{x}}_2\sim\mathbb{P}_{\hat{\boldsymbol{x}}_2}}{\mathbb{E}}[(||\nabla_{\hat{\boldsymbol{x}}_2} D_X(\hat{\boldsymbol{x}}_2)||_2 - 1)^2],\end{aligned} \tag{12}$$

where λ_0 controls the contribution of the cycle consistency loss, and λ_1 and λ_2 control the gradient penalty. $\mathbb{P}_{\hat{\boldsymbol{x}}_1}$ and $\mathbb{P}_{\hat{\boldsymbol{x}}_2}$ are the sampling distributions that uniformly samples along straight lines between pairs of points sampled from the data distribution and the generated distribution.

4. Experiments

To evaluate the performance of our proposed CycleWGAN, Improved CycleWGAN, and compare their performance with CycleGAN, we run experiments on the task of Chinese painting style transfer. Our baseline model is CycleGAN, and we aim to achieve stable training and high quality samples with CycleWGAN, and Improved CycleWGAN.

4.1. Dataset: CNPaintings

In this project, we collected a new dataset named CNPaintings which includes natural images and Chinese paintings of flowers. The reasons that we focus on the "flowers" data are two folds: (1) flower is a common theme in traditional Chinese paintings which enables us to collect a sufficient size of painting images; (2) flower themed Chinese paintings have relatively consistent styles through the years of its evolution, making it easier for the model to learn and for us to evaluate its performance.

There is no ready-to-use dataset of similar contents, so we collected the dataset by scraping images from Baidu using different keywords. Among many flower types, we concentrated on a few that are representative in Chinese paintings. Our keywords include: 牡丹国画 (Peony Chinese Painting), 梅花国画 (Plum Flower Chinese Painting), 菊花国画 (Chrysanthemum Chinese Painting) and 荷花国画 (Lotus Chinese Painting) for collecting Chinese Paintings with different flowers types; 牡丹 (Peony), 梅花 (Plum Flower), 菊花 (Chrysanthemum) and 荷花 (Lotus) for collecting natural images. We used the open source package on GitHub [26] to download all the images from Baidu. For each keyword, we downloaded about 200 images and reshape them to 256×256 pixels. Some sample natural and painting images are shown in Figure 1.

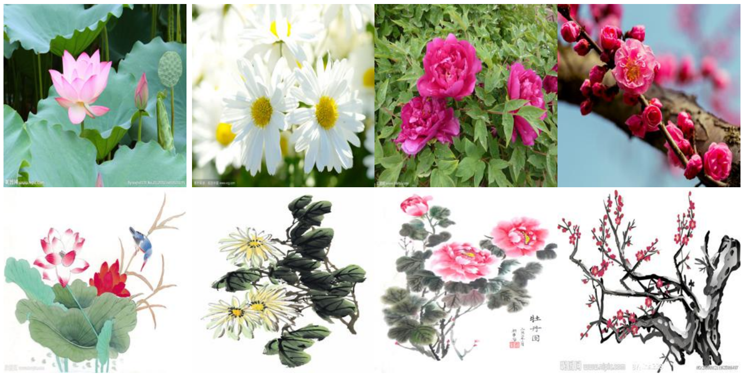

Figure 1: Four different types of nature flowers images and their corresponding Chinese paintings.

4.2. Implementation

4.2.1 Network Architecture

Our major network architecture is adapted from CycleGAN [1] and a Pytorch implementation of CycleGAN is provided at [27] as open source. We show the detailed network architecture for discriminator in Table 1 and for generator in Table 2.

4.2.2 Transfer Learning

While training the model, we adopted *transfer learning.* Instead of training the model on each type of flower completely separately, we first pretrained our model on the entire dataset of different types of flowers, then

Discriminator Layer Specification
4×4 Conv-LReLu layer, 64 filters, stride 2
4×4 Conv-Norm-LReLu layer, 128 filters, stride 2
4×4 Conv-Norm-LReLu layer, 256 filters, stride 2
4×4 Conv-Norm-LReLu layer, 512 filters, stride 1
4×4 Conv layer, 1 filter, stride 1

Table 1: The architecture and layer specifications of the discriminator in CycleGAN. Conv-LReLu represents a Convolutional-LeakyReLu layer, Conv-Norm-LReLu represents a Convolutional-InstanceNorm-LeakyReLu layer and Conv represents a Convolutional layer.

Generator Layer Specification
7×7 Conv-Norm-ReLu layer, 32 filters, stride 2
3×3 Conv-Norm-ReLu layer, 64 filters, stride 2
3×3 Conv-Norm-ReLu layer, 128 filters, stride 2
9 Residual blocks
3×3 Frac-Strided-Conv-Norm-ReLu layer, 64 filters, stride 1/2
3×3 Frac-Strided-Conv-Norm-ReLu layer, 64 filters, stride 1/2

Table 2: For our application, we use 9 Residual blocks. Each Residual Block contains two 3×3 convolution layers with the same number of filters on both layers. Conv-Norm-ReLu represents a Convolutional-InstanceNorm-ReLu layer, Frac-Strided-Conv-Norm-ReLu represents a fractional-strided-Convolutional-InstanceNorm-ReLu layer.

fine-tuned the pretrained model on each type of flower separately. Transfer learning allows rapid progress and improved performance when modeling each type of flower. It also reduces the computation cost by training each flower type from scratch.

4.2.3 Training Details

We implemented CycleWGAN and Improved CycleWGAN based on the initial implementation of CycleGAN in the following ways.

For CycleWGAN's implementation, compared with CycleGAN, we had these modifications:

- Clip the discriminator's weights into a range between $[-0.01, 0.01]$ for each gradient update and iterate the discriminator 5 times in every backpropagation process
- Use RMSProp optimizer for parameters update

After implementing CycleWGAN, we observed that the generator loss is one order of magnitude greater than the cycle consistency loss in the training phase, which significantly weakens the effect of cycle-consistency. So we run our experiments with the same settings but with different λ values of 50, 100 and 150 to increase the weight of cycle consistency loss (The default value of λ is 10).

Based on the CycleGAN's implementation, we made the following changes to achieve the Improved CycleWGAN model:

- Add gradient penalty for discriminators
- Use Adam optimizer for parameter update

For transfer learning, we first pretrained our model using the entire dataset, including 911 natural flower images and 900 flower paintings for 50 epochs. After the initial 50 epochs, we fine-tuned the pretrained model on each individual datasets. In the fine-tune stage, we fitted our networks with a learning rate of 0.0002 for first 100 epochs and then linearly decays to 0 over the next 100 epochs. We finished fine-tuning on the Lotus dataset with 200 natural images and paintings respectively. Next, we will explain the qualitative and quantitative evaluations based on the aforementioned models.

4.3. Results

4.3.1 Model Stability

Loss: To measure the stability of training, we considered tracking the change of the loss function. In particular, we will compare the fluctuations of the loss functions between different models. Recall that the losses for CycleGAN, CycleWGAN, and the Improved CycleGAN (CycleGAN_gp) model are defined in Equation 3, Equation 7, and Equation 11, respectively. In the following figures, the loss plots were obtained from the first 50 epochs of our fine-tuned model on the Lotus dataset and they were standardized on the same scale for comparison.

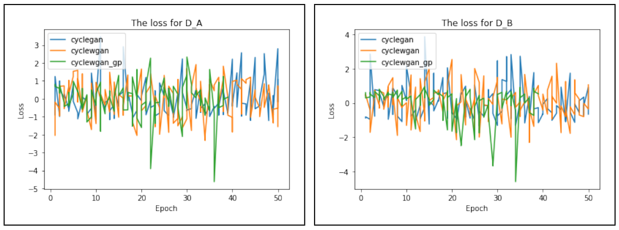

Figure 2: Loss function value of the discriminator for CycleGAN, CycleWGAN and Improved CycleGAN. We observed from the plots that the proposed CycleWGAN model shows the most mild oscillation in loss while the Improved CycleWGAN model shows the most chaotic loss distribution.

Weight Distribution: We've seen that CycleWGAN is more stable in the sense that its training loss is less variable over time. However, it suffers from

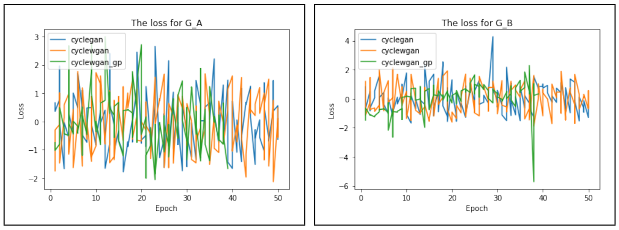

Figure 3: Loss function value of the discriminator for CycleGAN, CycleWGAN and Improved CycleGAN Similar to Figure 2, it can be observed that our proposed CycleWGAN shows the most mild oscillation in loss. Both Improved CycleGAN and CycleGAN shows more extreme values in their loss values.

optimization difficulties due to weight clipping. The drawback of this approach is that the weights will be stopped at the extreme values preventing them from reaching the optimum in the optimization. As a consequence, the discriminator tends to learn a simple mapping function with similar parameter values and the model fails to capture detailed information of the dataset. To address this problem, we proposed Improved CycleWGAN that introduces the gradient penalty to replace weight clipping. As shown in the below figure, both CycleGAN and Imrpoved CycleWGAN show diverse weight distribution compared to CycleWGAN.

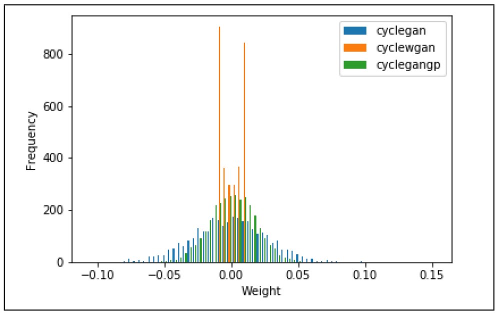

Figure 4: The weight distributions for CycleGAN, CycleWGAN and Improved CycleGAN.

4.3.2 Image Quality

We considered two ways to evaluate image quality: first by directly examining the generated images produced by CycleGAN, CycleWGAN and the Improved CycleWGAN; second by projecting the images to a 2-D domain and visualizing the locations of generated images with respect to real images of two different domains.

Figure 5 shows two examples of generated images. The images produced by CycleGAN are lack of color in general, resembling gray scale images with subtle color on the flower petals. The images produced by CycleWGAN looks slightly better, considering that it captures a wider range of color-tones. But still, they are lack of bright colors and the background is beige, which should be white in Chinese paintings. The images generated by Improved CycleWGAN seem to fit closest to real Chinese paintings compared to the other methods. The color is bright and vivid and the generated image also mimics the semi-transparent painting texture appeared in Chinese paintings. For further reference, we included some additional test examples in the Appendix 7.

Visually examining generated images is straight forward, but it is hard to compare all images qualities at once. Therefore, we propose a heuristic approach to compare the quality of generated images based on clustering. The key idea is to fit a binary classification deep neural net and extract features from its last fully connected layer to represent each image. We chose VGG-16 pretrained on Imagenet as the classification model. Then we added three additional layers: two fully connected layers of dimension 256 and 112 respectively, and a sigmoid layer at the end. The weights were learnt by optimizing the cross-entropy. Then we performed K-means clustering with the number of clusters $K = 2$ and visualized the clusters in the 2-D space with the coordinates being the first and second principle component of the design matrix. Because the extracted feature representations should be informative to distinguish the natural image domain and the Chinese painting domain. Ideally, the Chinese paintings and natural images should be set apart in two different clusters. Figure 6 shows such results that the natural images colored in pink and the Chinese paintings colored in orange largely locates in different clusters. The circles numbered 1 and 2 represent the center of each cluster.

To evaluate how closely the generated images fits in the Chinese painting domain, we fed the generated images to the trained deep neural network and obtain its feature representation Then we project them onto the 2-D space, still using first and second principle component as the coordinates. The generated images are colored in blue. As shown in Figure 3, Improved CycleWGAN has the best performance since most predicted images locate close to the cluster with Chinese paintings. Especially that many blue points are close to the center of the orange cluster. This finding is consistent with what we saw in Figure 5: images generated with CycleWGAN looks more like realistic paintings. We also notice that some images generated by CycleGAN and CycleWGAN locate in the middle of the two

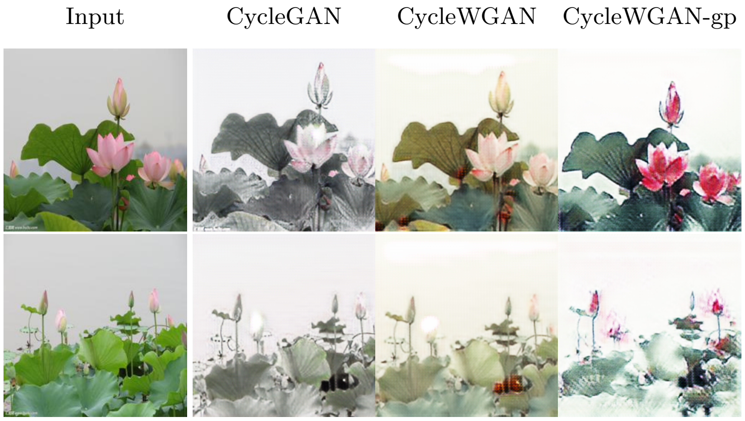

Figure 5: Two test images for CycleGAN, CycleWGAN and CycleWGAN_gp with selected images

domains, implying that some of their generated images are very successful as they have traits of both natural images and Chinese paintings.

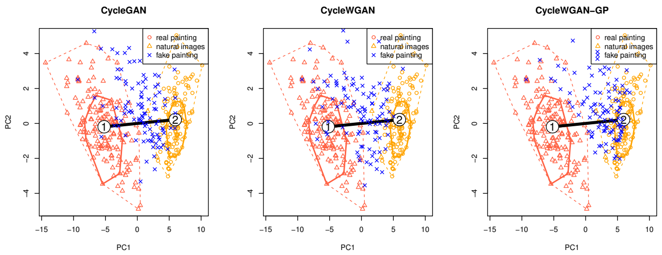

Figure 6: Model evaluation based on K-means. The shape defined by solid lines covers 50% of the points in that cluster, and the shape defined by the dotted lines covers 90% of the points in that cluster.

4.4. Conclusion and Future work

To summarize our key findings, we implemented CycleWGAN and Improved CycleWGAN, and compared them with CycleGAN in the aspect of model stability and image quality. We found that in terms of model stability, CycleWGAN has the most stable loss values during training while Improved CycleWGAN shows the most chaotic loss fluctuation. For image quality, Improved CycleWGAN generates images that looks most realistic to Chinese paintings. We also verified this by projecting images to a 2-D space and showed that images generated by Improved CycleWGAN locate closest to the center of the Chinese painting domain compared to CycleGAN and CycleWGAN.

There are many intriguing behaviors observed in our experiments and we wish to study them as our future work. During CycleWGAN's training phase with default parameters, we found that the losses for discriminator and generator are about 10 times larger than the cycle consistency losses. The training loss was dominated by the adversarial loss and the effect of cycle consistency loss is greatly weakened. As a result, it is difficult for CycleWGAN to transform images back to their original domain. We tried to increase the weight of cycle consistency loss to achieve a balance between the loss values and we hope to study if there is a better strategy to choose the weight.

In addition, we see from Figure 4 that the weights of CycleWGAN are clustered at the clipping constant, and in Figure 6 that many generated Chinese paintings are scattered in the middle of the two domains. We suspected this is because our selected weight clipping constant is too small, preventing model's parameters from changing much in each iteration, and constraining the learned mapping to be simple. In future work, we hope to verify this by tuning the clipping constant and explore alternatives such as adapting its value over time.

In terms of additional applications, we hope to apply our model to video artistic style transfer. In that case, we'll need to consider the temporal consistency between adjacent frames of the video. We'd also like to try learning multiple artistic styles at once, treating each style as a domain and performing translation across multiple domains with models such as StarGAN [28].

References

[1] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

[2] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[4] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. Automatic image colorization via multimodal predictions. In *European conference on computer vision*, pages 126–139. Springer, 2008.

[5] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[6] Xiaohan Jin, Ye Qi, and Shangxuan Wu. Cyclegan face-off. *arXiv preprint arXiv:1712.03451*, 2017.

[7] Naveen Kodali, James Hays, Jacob Abernethy, and Zsolt Kira. On convergence and stability of gans. 2018.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[10] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[11] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. In *ACM Transactions on Graphics (TOG)*, volume 28, page 124. ACM, 2009.

[12] Yichang Shih, Sylvain Paris, Frédo Durand, and William T Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)*, 32(6):200, 2013.

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[14] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[16] 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.

[17] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, pages 1349–1357, 2016.

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proc. CVPR*, 2017.

[19] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. Can: Creative adversarial networks, generating" art" by learning about styles and deviating from style norms. *arXiv preprint arXiv:1706.07068*, 2017.

[20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[21] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[22] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017.

[23] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[24] Yuan Chen Guanyang Wang, Ying Chen. Chinese painting generation using generative adversarial networks. 2017.

[25] Daoyu Lin, Yang Wang, Guangluan Xu, Jun Li, and Kun Fu. Transform a simple sketch to a chinese painting by a multiscale deep neural network. *Algorithms*, 11(1):4, 2018.

[26] Baidu image spide. `https://github.com/kong36088/BaiduImageSpider.git`. (Accessed on 03/10/2018).

[27] junyanz/pytorch-cyclegan-and-pix2pix: Image-to-image translation in pytorch (e.g. horse2zebra, edges2cats, and more). `https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`. (Accessed on 03/11/2018).

[28] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *arXiv preprint arXiv:1711.09020*, 2017.

5. Appendix

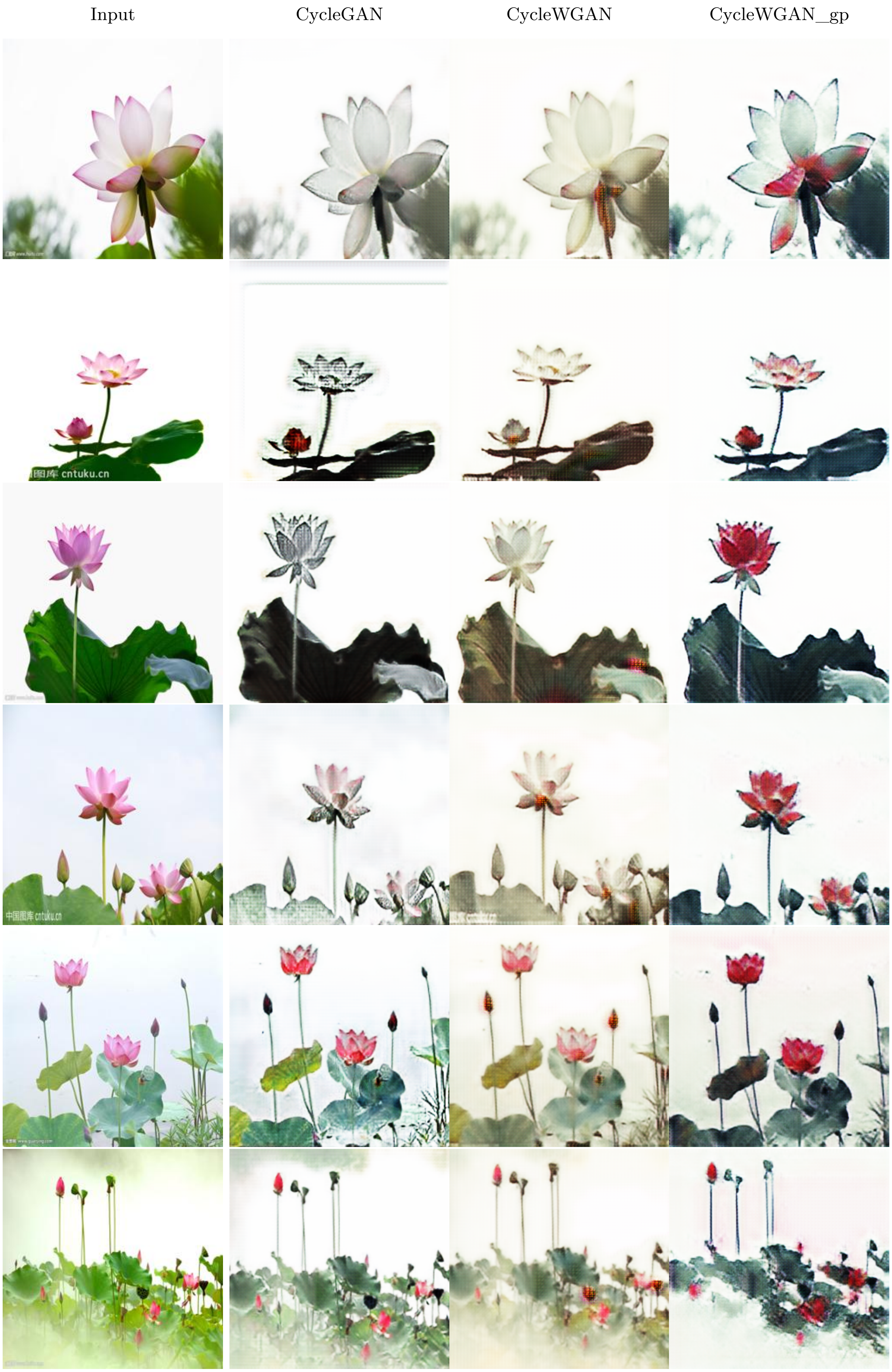

Figure 7: Test results for CycleGAN, CycleWGAN and CycleWGAN_gp with selected images.