Making UC San Diego Snowy Again

Kening, Zhang
A92119894
kez040@ucsd.edu

Shuyi, Ni
A92037903
sni@ucsd.edu

Yuhan, Liu
A53319530
yul139@eng.ucsd.edu

Zhirui, Dai
A53309763
zhdai@eng.ucsd.edu

Abstract

We propose a model based on cycleGAN and attention. It can be applied to scene transformation, for example, modifying a image of UC San Diego to be snowy. cycleGAN, one of the prevailing condtional GAN, proposed by Zhu et al.[7] is a good architecture for such tasks. In order to make the model more sensitive at inferencing which parts of the image should be changed most from one domain to another, we attempt to add an attention mechanism on cycleGAN. Mejjati et al.[4] propose incorporating an attention mechanism so that the attention network can accurately find the foreground that is within the domain of our interest. The main contribution of our paper includes: we test the features of different generator and discriminator architectures and conclude that generator with U-Net structure and patch discriminator and a Mean-Square-Error GAN Loss would generate high quality of images; we incorporate a attention mechanism with cycleGAN on specific task of adding snow scenery to images and make the modification reasonable;

1 Introduction

In this paper, we aimed to transform an image of UCSD to be in snowy winter. In the generated image, some objects like buildings and trees should be kept. In an ordinary GAN, the generator may remove these objects in the generated image and still succeed in fooling the discriminator. Therefore, the consistency between the input and the output is also a requirement, which is fulfilled by cycleGAN. A cycleGAN contains a pair of GANs, forward: $G : X \rightarrow Y, D_Y$, and backward: $F : Y \rightarrow X, D_X$. The cycle consistency is guaranteed by the cycle consistency loss which is calculated by the difference of X and $F(G(X))$ or Y and $G(F(Y))$.

In the task of image-to-image translation, the basic idea is to train a model to learn the distribution of two domains and the mapping between these two. But there could be thousands of mapping that works for the generator to fool the discriminator. The solution space is too large for a general GAN without any improvement to learn a stable parameter configuration. With cycle consistency loss, the space of solutions to search is much narrower, as described in Zhu et al.[7]. And it is more possible to train a mapping where X and Y is a meaningful pair as our expectation.

But there is still some potential improvement.

In the world of information, we are dealing with thousands of different types of signals, most of which are visual signals such images and videos. As languages need to be translated in between for better understanding and message exchange by paying attention to some keys in the context, attention mechanism can also improve model performance on image-to-image translation tasks, such as the translation of styles between images results in fantastic artistic effects[3], denoising images

[6], and completing missing part of images. With an attention mechanism, a cycleGAN model can be guided to work harder on some part of the image to make the output more reasonable and realistic.

Besides, since the original adversarial loss gives low gradient at the beginning of training generator, Zhu et al.[7] used least square sum to calculate the adversarial loss. In this paper, we used Wasserstein loss suggested in Arjovsky et al.[1] to train the generator, which should give a better outcome.

2 Motivation

Our project is motivated by the fact that style translation using cycleGAN usually performs domain changing on the whole image. Although cycleGAN is generally good at this task, we find that artifacts are generated sometimes because background are wrongly alternated. However, the background is actually not within our interested domain which should be left out unchanged. As a result, we added an extra attention mechanism to basic cycleGAN architecture to enforce the translation being applied only on interested domain of the image. For our specific summer to winter task, we hope the attention would help the generator to focus on the places where snow could be added.

3 Dataset

Summer2Winter Yosemite

Summer2winter Yosemite is a data-set created and used in Zhu et al.[7]. We will train our model on this data-set and compare our result with cycleGAN's to check the performance improvement.

Cityscapes and Snow100K

We created a data-set of city views in summer and winter by combining two existing data-sets, Cityscapes and Snow100K.

Cityscapes data-set provides 5000 annotated images with fine annotations under various conditions, e.g. 50 cities, different seasons, varying backgrounds etc. Snow100K contains 100k synthesized snowy images with corresponding snow-free ground truth images.

Since there are both summer and winter images in Snow100K, most images are from Snow100K. But in order to add more information of city scenery, we introduced some images from Cityscapes.

4 Methods

4.1 Network Architecture and Formulation

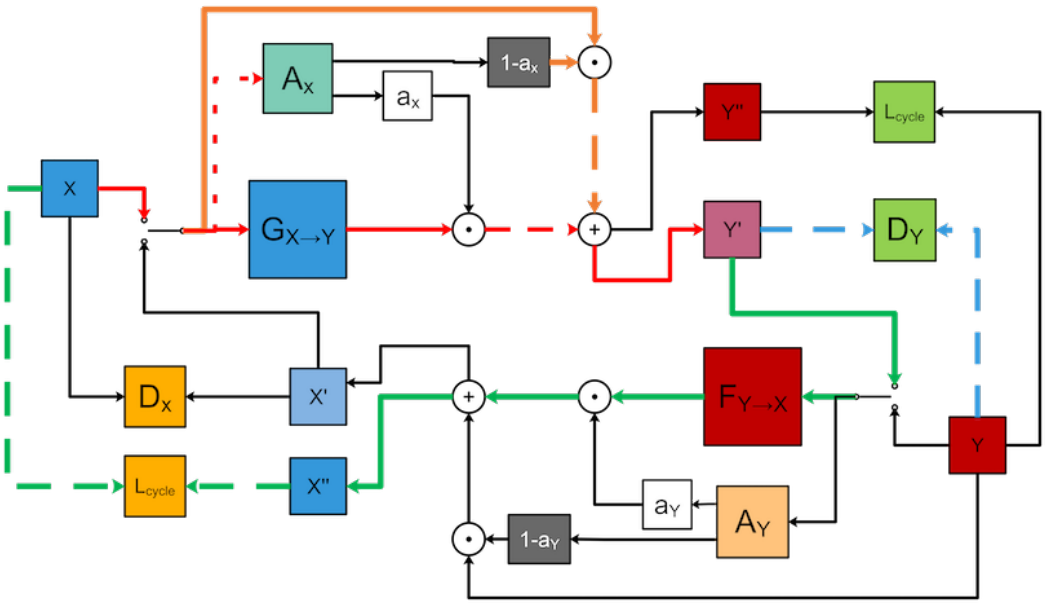

Figure 1: Network Architecture

Our framework will be built upon GAN and attention mechanism[5]. We will adapt our generator and discriminator from cycleGAN.[7].

4.2 Data Flow in the Pipeline

As the diagram in figure (1) shows, the network includes both cycleGAN and attention mechanism.

1. **generation of fake image**: as the red solid path shows, an image in the X domain X is fed into the generator $G_{X\to Y}$. A fake image is then generated but there is still something about attention to do before reaching the fake image Y'
2. **attention mechanism for generator**: the red dot path indicates that an image x is fed into the attention network and the network generates an attention map a_x where every element value varies between 0 and 1. A negative attention map is also generated from a_x by $1 - a_x$. The orange solid path shows that the negative map element-wisely multiplies with the original image x to get the original part, which is represented by the origin dash path. Similarly, the red dash path represents the modified part. By element-wisely adding the original part and the modified part together, the network finally generates the fake image Y'.
3. **attention mechanism for discriminator**: as the blue solid path shows, the discriminator also uses the attention map to pay more attention to the modified area.
4. **discrimination of images**: the blue dash path shows that the fake image Y' and the real image Y are both fed into the discriminator D_Y to distinguish fake images from those real ones and then calculate loss.
5. **cycle and cycle loss**: along the green solid path, a fake image Y' is sent to the generator $F_{Y\to X}$ to generate a recovered image X''. The attention mechanism is also used in this path. Then, as what the green dash path shows, the cycle loss is calculated by comparing X and X''.

4.3 Generator

In cycleGAN, there are two generators $G_{X\to Y}$ and $F_{Y\to X}$. $G_{X\to Y}$ and A_X are combined as a new generator G and F is similar. G and F are expected to be the inverse of each other. As mentioned in Zhu et al.[7], we are actually training two autoencoders, $F\circ G : X\to X$ and $G\circ F : Y\to Y$.

Since we are learning a pixel-to-pixel translation between two images of the same size and this task's input-output is similar to image segmentation, we introduced a U-Net structure with encoder, decoder and skip connections for our generator $G_{X\to Y}$ and $F_{Y\to X}$.

4.4 Discriminator

The discriminator in [3] utilizes the Markov assumption that pixels are independent with a distance apart in an image. It penalizes the generator at the scale of patches, and convolves these batches together to get the final output of discriminator. In this way, the network can run faster because there will be fewer parameters and adapt to images of different sizes.

4.5 Loss function

4.5.1 Cycle Consistency

The goal of our task is to map images between two domains. To make such mapping more robust, we add a constraint that enforces a backward translation which inverses the images in domain Y back back to domain X and vise versa. i.e. For each images in X domain, the cycle translation would bring it back itself and the same for Y domain.

$$L_{cycle}(G,F) =\mid G(F(x))-x\mid +\mid F(G(y))-y\mid \tag{1}$$

where G and F are both consist of the original generator ($G_{X\to Y}, F_{Y\to X}$) and their own attention mechanism (A_X, A_Y).

4.5.2 Wasserstein Loss

Comparing to original implementation in cycleGAN which uses least-square loss, Wasserstein distance [1] is proved to be more robust against noise and the imbalance between generator and discriminator. It is an efficient approximation of Earth Mover(EM) distance which is a good metric to measure the distance between distributions. Wassersteins distance adapted in GAN setting is:

$$L_{adv}(G,F,D_X,D_Y) = (\mathbb{E}[D_Y(y)]-\mathbb{E}[D_Y(G(x))]) + (\mathbb{E}[D_X(x)]-\mathbb{E}[D_X(F(y))]) \tag{2}$$

The above equation shows the competition between the generators and the discriminators: The discriminators want to maximize the difference between the two distributions (i.e. the distribution of the real images, and that of the generated images by the generators), while the generators try to minimize their difference so as to fool the generator. These differences are represented by the distance between the means.

4.5.3 Mean squared error

Following the loss function of LSGAN[**?**], we also implement a Mean-square Error as the adversarial loss comparing to Wassersterin Loss. It defines as:

$$L_{adv}(G,F,D_X,D_Y) = \frac{1}{2}\mathbb{E}[(D_Y(y)-a)^2]+\frac{1}{2}\mathbb{E}[(D_Y(G(x))-b)^2]+\frac{1}{2}(\mathbb{E}[(D_X(x)-a)^2]+\frac{1}{2}\mathbb{E}[(D_X(F(y))-b)^2]) \tag{3}$$

where a and b are the labels for real and fake data.

4.5.4 Generator Loss

We combined Wasserstein loss (L_{adv}) [1] with the cycle consistency loss (L_{cycle}) [7] in hoping to extract their advantages:

$$L = \min_{G,F}\max_{D_X,D_Y} L_{adv}(G,F,D_X,D_Y) + \lambda L_{cycle}(G,F) \tag{4}$$

The hyper-parameter λ governs the influence of the cycle consistency - the larger it is, the stricter it enforces the cycled image to be similar to the original image, thus making the two generators inverse of each other.

4.6 Attention

The attention mechanism is frequently used in sequence to sequence translation to strength the correlation between the sub-element in side a sequence. In image to image translation task, we want to learn a mapping from a source domain to a target domain. However, we find that generator sometimes generates low quality image because it lacks the ability to focus on parts where actual domain transformation happens. Thus, we want to add an attention mechanism to help generator detect the most discriminative parts between source and target domain.
To do so, we add attention networks to guide our generator. A_x and A_y are two attention networks for forward and backward directions respectively. It is expected that the attention networks will learn where the most discriminative parts are. The output of the attention network will be a continuous attention map. We will get the final generated image using Equation 5 where x_a is the attention map, so that the output image will be a blend of transformed foreground with unchanged background.

$$\boldsymbol{x}' = \boldsymbol{a}_x \odot G_{X \to Y}(\boldsymbol{x}) + (1 - \boldsymbol{a}_x) \odot \boldsymbol{x} \tag{5}$$

The discriminator will still use the full image without applying the attention map because unlike traditional domain translation performed on specific objects, the distinction between foreground and background is not that clear.

4.7 Evaluation Metric

We used Kernel Inception Distance (KID) proposed in [2] as our quantitative evaluation metric for comparison. It measures the dissimilarity between two probability distributions using samples drawn independently from each distribution. In [4], it reports the KID value for various GAN networks which we adopted for our result comparison. We will report KID on Summer2Winter Yosemite validation set as comparison.
We define X, Y as encoded samples from two domains, k as kernel function, m as the number of samples in X domain and n as the number of samples in Y. The following function defines an unbiased estimator of the Maximum Mean Discrepancy. X and Y should be drawn from hidden space mapped after the pooling of inception network.

$$MMD_u^2(X, Y) = \frac{1}{m(m-1)} \sum_{i \neq j}^{m} k\left(x_i, x_j\right) + \frac{1}{n(n-1)} \sum_{i \neq j}^{n} k\left(y_i, y_j\right) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k\left(x_i, y_j\right) \tag{6}$$

5 Optimization method

5.1 ReZero

By using ReZero technique, we are able to significantly reduce our training time. In our attention model and the ResNet generator, several residual blocks are used. To speed up the training of these blocks, we use the ReZero method, which is adding a learnable scalar α to the output of the convolution layer in a residual block:

$$x + Conv(y) \to x + \alpha \times Conv(y) \tag{7}$$

α is set to 0 at the beginning so that the problem of gradient vanishing or exploding problem is well solved.

By using ReZero, we speeded up our training of Attention model from about 200 epochs to 6-10 epochs.

5.2 Pretrain cycleGAN

This optimization is used along with attention. We find one problem with our attention mechanism is that the attention map is not confident enough, the most confident value is around 0.6 at first.(The attention map has continuous value between 0 and 1 for each pixel where a higher value indicates the pixel is more likely to be within the domain of interested.) As a result, the generated images are dominated by the background so that it is not so different from the original image. We think the reason behind this phenomenon is that our generator works poorly at the very beginning which misguides the attention map being prone to background in order to reduce the loss. To prevent this, we pretrained our generator before training our attention map.

6 Results and Discussion

We run experiments on four architectures: ResNet with MSE, UNet with MSE, UNet with W-Loss and UNet with MSE and attention where the ResNet and UNet are different architecture used for generator.

Generator Architecture	Loss Function	Ratio(D/G)	Optimizer	Adding Attention
ResNet	MSE	0.2	Adam	No
UNet	MSE	1	Adam	No
UNet	W-Loss	1	RMSprop	No
UNet	MSE	1	Adam	Yes

Table 1: Hyper-parameter Table

6.1 Training details

We set the ratio of discriminator training vs generator training to be 1:5 (i.e. we train one epoch of discriminator, and five epochs of generator after that) for Resnet MSE model because we experimented and found out that generator loss is harder to be optimized. It is also the reason why we tried different loss functions. But in other experiments, a ratio of 1 is suitable as generator and discriminator have the same strength.
For all experiments, we follow cycleGAN[7] and set $\lambda = 10$ in Equation 4. Batch size is 32. We use Adam as both generator optimizer and discriminator optimizer with same learning rate(except for WGAN, RMSprop is more stable). Learning rate decay is applied every 100 epochs with gamma decay rete 0.1. We follow cycleGAN's implementation of $2e^{-4}$ and the generator loss is optimized.

6.2 ResNet vs UNet

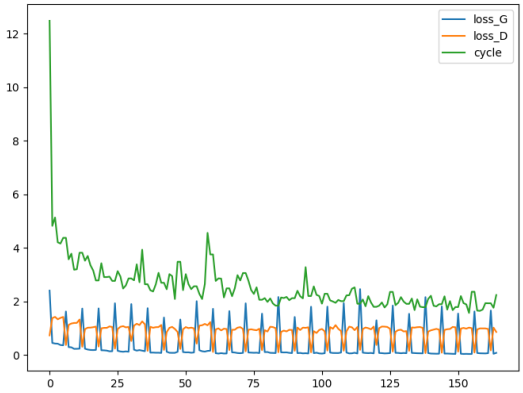

Figure 2: Validation loss of Resnet MSE

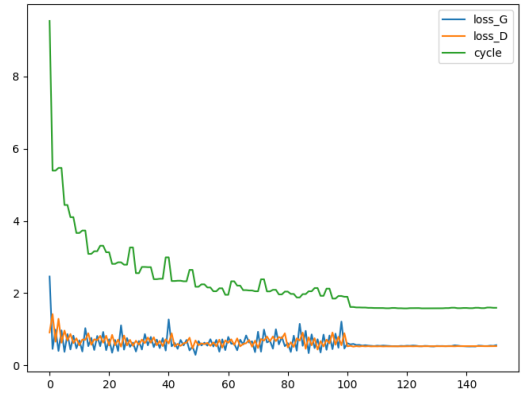

Figure 3: Validation loss of Unet MSE

In the architecture of U-Net, the information from the down-sampling side is allowed to flow to the up-sampling side through skipping connections. This special architecture allows U-Net to reserves some common features in both the input domain and the output domain and transforms some unique features of the input domain to the output domain.

From the two figures above, we can find that U-Net converges faster than ResNet because it learns the mapping between two domains much better than ResNet. Also refer to the test images generated in Figure 8, the images generated by U-Net are better than that generated from ResNet from the perspective of human evaluation. It indicates that the loss curve may not reveal correction information of models. U-Net model is better at adapting to a new domain comparing to ResNet.

6.3 MSE Loss vs W Loss

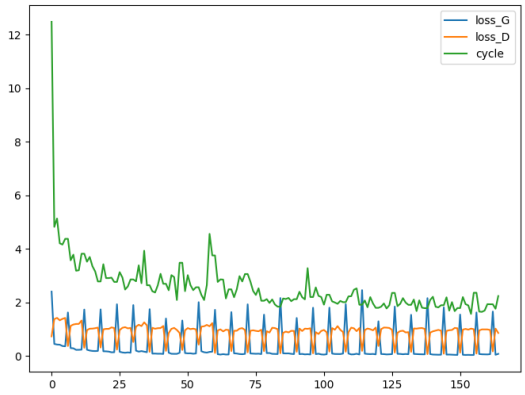

Figure 4: Validation loss of Resnet MSE

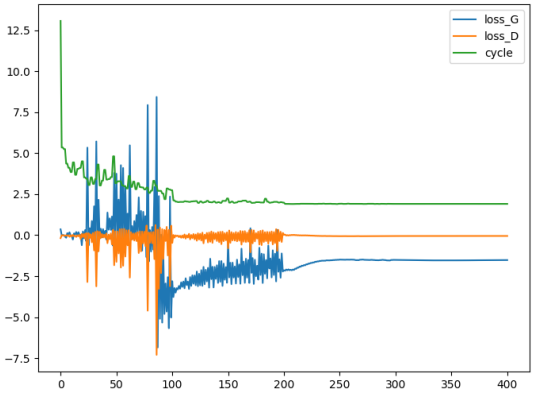

Figure 5: Validation loss of Resnet Wloss

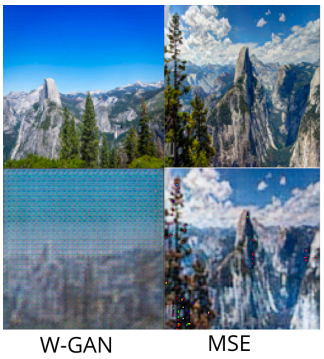

Figure 6: Validation loss of Resnet Wloss

By analyzing our validation loss curve, we could see that the generator and the discriminator are competing with each other more in WGAN. The variance of Generator loss is more than that of generator as the objective function of Wloss does not contain a regularization of loss norm. So we add a gradient penalty regularization to constrain the change of gradient each step. The cycle loss in WGan is not reduced as much as in the MSE Model, which means the images are not as precise as the images in MSE model. The KID score in Table 2 also shows that the quality of images generated by WGAN is further from the original input as it contains some upsample patterns. However, it does not contain small noisy dot appeared in MSE model. Such difference is the result of loss function.

6.4 Attention

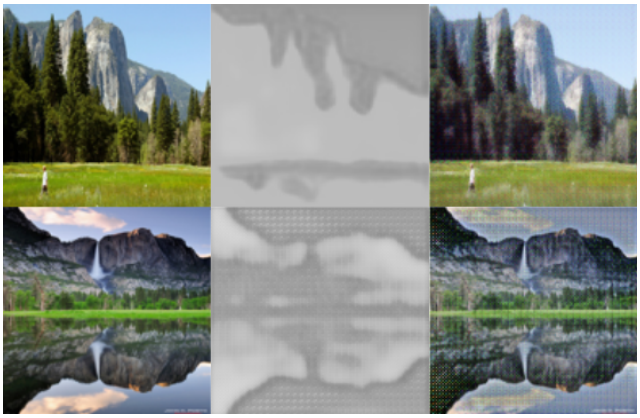

Figure 7: Original, Attention Map, Generated Image

The above is the original images in summer domain, its corresponding attention map and the generated images in winter domain. The brighter region in the attention map indicates the region that mostly discriminates the two domain. We could see that the result attention map is pretty good. The brighter region includes grass, trees and mountains which are reasonable regions to add snow. In the generated image, the color of grass and trees are faded to get close to white. However, we notice that the resulting image is not significantly different from the original image.We suspect the reason is that our attention map is still not confident enough. The best case would be the foreground being white in the attention map. We use the optimization of pretrain cycleGAN described in Section 5 and the attention map does has some improvement. However, because summer to winter domain does not have distinctive boundary between foreground and background, we find the involvement of background actually affects the performance of discriminator which tends to identify fake images as real. As a result, the generator improves so slow. It is possible that the result will get better with long enough training epochs.

6.5 Probelms

Generator Architecture	Loss Function	KID	variance
ResNet	MSE	0.0085	0.0026
UNet	MSE	0.0086	0.0022
UNet	W-Loss	0.0089	0.0025
UNet	MSE	0.0083	0.0020

Table 2: KID Table

Figure 8: Validation loss of Resnet Wloss

Our translation from summer domain of UCSD to winter domain is far from perfect. One of the reasons is that it is hard to find a suitable dataset that matches our expectation. The summer to winter dataset we used mainly contains summer and winter of natural sceneries. It would be better if we have summer and winter dataset for architectures. The snow100k dataset we try to use does not work well as its images various a lot and some of the images does not have clear distinction between seasons.

As the Table 2. indicates that the images generated with attention are the closet to the original images(KID is lowest). This is the result of reusing part of the original images as the fake images. The variance is also the lowest since the map learned to guide the signal is served as a constrain. However, the change of image is pretty mild and could be considered as "conservative".It only add some whiteness to the ground.

7 Conclusion

In We tried both U-Net and ResNet as the generator, and found that U-Net performs better We tried both patch and full image discriminator, and the results show that patch discriminator is a better option By combining attention and U-Net, we successfully generated a picture of UCSD in snowy winter Our goal is to transfer an image from summer to winter and add snow to it as a style transformer. We see the fake image is generated as applied a filter on the original image. The shape is conserved but a season transformer still needs much efforts. To achieve our goal, we need to find the best task-specific hyper-parameters and a MSE loss might be a suitable function for our task.

8 Team Member Contribution

- Kening Zhang: implemented cycleGAN model, KID
- Shuyi Ni: implemented attention model
- Yuhan Liu: implemented discriminators
- Zhirui Dai: implemented generator, training pipeline, dataloader

References

[1] Martin Arjovsky, Soumith Chintala, and Lon Bottou. Wasserstein gan, 2017.
[2] Mikoaj Bikowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2018.
[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks, 2017.
[4] Youssef A. Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image to image translation, 2018.
[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
[6] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image de-raining using a conditional generative adversarial network, 2017.
[7] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss, 2017.