Learning of Unpaired Image-to-Image GAN Method

Weijie Gan
weijie.gan@wustl.edu

Abstract

Most problems in image processing and computer vision is to translate input image into object image, which is called image-to-image problem. Recently, a novel GAN method introduce a common framework for most kind of such problem without paired image. In this project, we figure out how this method work and implement it by python code. What's more, a approximate similar method with paired image, pixel-to-pixel, is also implemented for comparison.

1 Introduction

Many problems in image processing and computer vision filed are to translate images into corresponding output images. For example, to convert a RGB image into an edge map, a depth map or a semantic label map. Since both the input and the output in these tasks are images, these problems can also be called as *image-to-image* translation.

In general, each kind of image-to-image tasks is finished by different methods. The goal in this field is to find a common framework can finish some or even all kind of tasks.

In this direction, researcher took an important step by deep-learning technology, especially convolutional neural network(CNN) which can easily build up a nonlinear map f_{CNN} between pixels from input image x to output image y. In CNN method, what we need to do in most case is to minimize the Euclidean distance between output of map function with input image as input $f_{\text{CNN}}(x)$ and output image y.

The advantages of CNN method are that it can build up a map for different kind of applications using same network structure and that it has many different structures for another difficult or advances problems. However, if we take native Euclidean distance as the objective function needed to be minimized, the result from CNN tend to be blurry[1]. This is due to Euclidean distance only measure **average** all value error between network outputs and ground-truth images.

It would be fairly desirable if we can involve more high-level measurement into the objective function and help output from network to be more natural and more satisfying.

Fortunately, this is actually what Generative Adversarial Networks(GAN) does. In GAN, it have an extra network(formally adversarial network) to estimate if the output image is fake(looks like an artificial image) or true(Looks like a natural image). Then, results from this estimation network will be also used to train CNN network discussed above(formally generator network). That helps the output image not only fits Euclidean distance constraint, but also looks more natural, since blurry image will not be tolerated by estimation network.

Experiments show great performances of GAN methods in image-to-image problems. The main contributed papers now are Pixel-to-Pixel paper[2] and Cycle-GAN paper[3]. The second paper provide a more flexible way that does not require paired data in training process(discussed later).

The goals in this project is (1) to understand and implement a image-to-image GAN method based on the the Cycle-GAN paper, (2) to estimate my experiment results and (3) to compare the different between Pixel-to-Pixel method and Cycle-GAN method theoretically and experimentally, though I only implement a approximate Pixel-to-Pixel method for comparison in experiment part. The source code is published in my GitHub: https://github.com/wjgancn/WashU/tree/master/cse659/proj2.

2 Background & Related Work

2.1 Generative Adversarial Network

GAN is based on zero-sum game framework with contest between two neural network[4]. These two neural network are a generative network generates candidates and the discriminate network evaluates if candidates is good.

The goal of discriminate network is trained to classify good candidates and bad candidates. The aim of generative network is to provide better and better outputs that make the discriminate network hard figure out if the output is good or bad. On the other word, the generative network need to improve itself and "cheat" the discriminate network successfully.

The training of two network is dynamic, together and respectively. An expected results is that the discriminate network is "responsible" and have good performance in classification problem while the generative network is also good enough to provide many "fake" but perfect candidates that can be not classified by the discriminate network.

Mathematically, the task in discriminate network is to Maximize expect of $\log(D(y)) + \log(1 - D(G(x)))$ given network D, where y is given image, x is noised input image, $D()$ means output possibility of discriminate and $G(x)$ means output of generative network. This formulation leads possibility output of discriminate network for given image y becomes 1 (real) and for image generated from generative network becomes 0 (fake).

Correspondingly, the problem in generative network is to "cheat" discriminate network, and can be shown as $\min_G \log(1 - D(G(x)))$, which on the contrary, leads possibility output of discriminate network from generative network becomes 1 (real).

Thus, the GAN problem can be formulated as:

$$\min_G \max_D \ \log(D(y)) + \log(1 - D(G(x)))$$

2.2 Pixel-to-Pixel Model

Given an expected image, the original GAN is to learn and generate results that is similar with the expected image, with noised image as input(make x in above discussion, become y). This idea is a unsupervised method and can be used to generate some fake, artifact but natural images.

However, it's not what we need to do in the image-to-image problem. In image-to-image problem, the generate network is not to provide a fake image, but to be a map function help one image can be converted into other image.

In Pixel-to-Pixel Model, the training of generator network is a supervised learning process, with input image and expected output image are given. It extends the original GAN by requiring the generate network to "cheat" the discriminate network and also to force the output to be conditioned on the input, using distance measurement in the loss function. That means the input x is not a random noised image, but also a given image. For example, in real-image to edge-image problem, the x will be the real-image while y will be related edge-image. The optimization formulation of generative network becomes:

$$\min_G \log(1 - D(G(x))) + v||x - y|| \tag{1}$$

where v is a parameter control strength of similarity between x and y. It's called Conditional GAN. Pixel-to-Pixel model is not the first one to apply condition GAN, but the first paper apply it into image-to-image problem.

This model is a successful framework for many kind of image-to-image problem. It still have a obvious limitation in practical application that it required x and y images are paired. For example, in real-to-depth problem, if x is the real image, y must be the corresponding depth map. However, for many tasks paired training data will not be available.

In order to solving this problem, a novel Cycle-GAN without such requirement is introduced and discussed as follow.

3 Method

The novel idea in unpaired GAN(cycle-GAN) is that it has two generative networks and there is no direct relationship between the input x and y in loss function in generative network.

For the first idea, it can be seen as figure (1). For one of inputs, x, the first generative network is similar with other GAN method, which means to generate an expected output $\hat{y}$ corresponding with x. The second generative network does a inverse operation, help the $\hat{y}$ become the input x again, mentioned as x_{recon}. For the other input y, the process is identical, but firstly passing the second generative network and secondly get the reconstruction y_{recon}.

When inferring, the given test image pass the first generative network and get the output. The discriminate network in this method is also applied to classify ground-truth input and the output of first generative network.

3.1 Loss Formulation

The loss function in cyclegan consists of two part: (1) the adversarial loss similar with general GAN: $\min_G \log(1 - D(G(x)))$ and (2) the cycle consistency loss.

In general, the adversarial training can make the first Generative network produce outputs that have the same distribution with the targets (in this case, the given image y). However, the target domain is too large to let random and wrong mapping be produced.

In conditional GAN model, we add a (Euclidean) distance constraint beside adversarial loss to make sure a correct mapping in generative network. Despite that, in unpaired method, we can not find a same constraint. So, instead of a constraint in forward path (from x to $\hat{y}$), we use constraint in the inverse operation by adding extra generative network, which means when passing the first then the second generative network, the output x_{recon} is still similar with the input x: $\min ||x_{recon} - x||$. With this term, we can make sure no random mapping, but not a correct mapping. We still need to use y as input to show the first generative network what a correct mapping result looks like. Similarly, we have $\min ||y_{recon} - y||$.

Thus, given the first generative network as G, the sec-

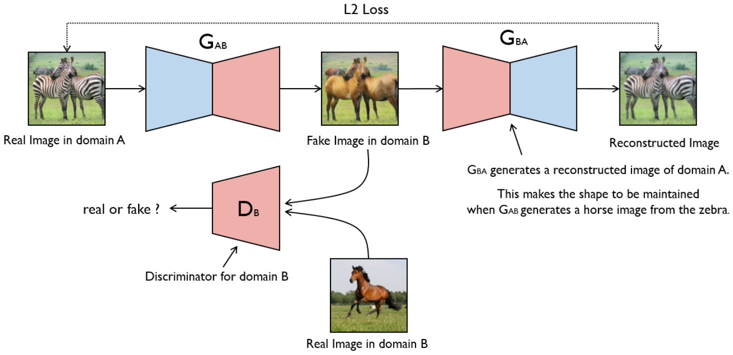
L2 Loss
GAB
GBA
Real Image in domain A
Fake Image in domain B
Reconstructed Image
GBA generates a reconstructed image of domain A.
This makes the shape to be maintained
when GAB generates a horse image from the zebra.
real or fake ?
DB
Discriminator for domain B
Real Image in domain B

Figure 1: Idea of Cycle GAN. Download From https://github.com/eriklindernoren/Keras-GAN

ond one as F, the generative loss formulation will be:

$$\begin{aligned} L_G = &\min \log(1 - D(G(x))) + \\ &||x - F(G(x))|| + ||y - G(F(y))|| \end{aligned}$$

And the loss function of adversarial network will be:

$$L_D = \max \log(D(y)) + \log(1 - D(G(x)))$$

Above formulation is the basic idea in this method. In practical, inspired by idea from LSGAN[5] and details in paper, these two loss functions should be modified as:

$$\begin{aligned} Loss_G = &\min ||1 - D(G(x))||_2^2 + \\ &\lambda(||x - F(G(x))||_1 + ||y - G(F(y))||_1) \end{aligned} \tag{2}$$

And,

$$\begin{aligned} Loss_D = &\min ||Ones - D(y)||_2^2 + \\ &||Zeros - D(G(x))||_2^2 \end{aligned} \tag{3}$$

where $Ones$ and $Zeros$ are matrix have same dimension with the output of D, the discriminate network with all values are 1 or 0. It's called patchGAN that is also introduced in pixel-to-pixel model.

3.2 Network Structure

Both generative network and discriminate network are Convolutional Neutral Network based on original paper, shown as figure (2). Both of two generative network, G and F have the same structure.

3.3 Optimization

As suggested in the original cycle-GAN paper, we need to train discriminate network at one time, then generative network and train it repeatedly. We use mini-batch SGD with batch size as 4 and apply the Adam solver, with a learning rate of 1e-4 for generative network training and 1e-6 for discriminate network training.

3.4 Why Unpaired GAN Works

The cycle gan use unpaired images. But this ”unpaired” character is shown in contents or corresponding pixel values. The style or image category are actually paired. We actually use this potential paired character, and this is the reason why I think cycle gan can work without paired images.

4 Experimental Results

In the experiment part, four kinds image-to-image problems are implemented, including the building image to label, building label to image, photo to cezanne and cezanne to photo.

A comparison pixel-to-pixel experiment, that drop reconstruction network F and uses paired images, is also implemented, using generative loss function in equation (1) and same discriminate loss function with cyclegan.

4.1 Difference with Original Paper

Due to limited time and GPUs source, experiments are simplified compared with original paper, including: (1) Drop one discriminate network for classify if $F(y)$ is fake or real. Only discriminate network for $G(x)$ is implemented. (2) In comparison pixel-to-pixel experiment, I keep the same generative network, shown as figure(2), while the generative network in pixel-to-pixel paper is U-Net.

4.2 Dataset and Preprocessing

Two datasets, ”facades.zip” and ”cezanne2photo.zip”, are downloads from https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/?C=S;O=A. Preprocessing includes resize images into 128 * 128 and scale its value to range [-1, 1].

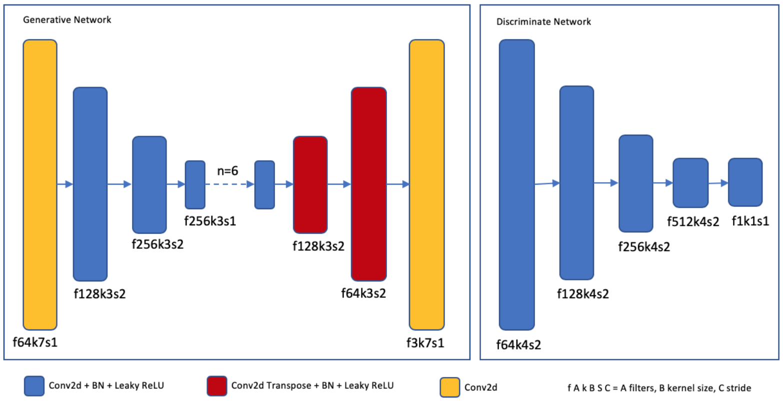

Figure 2: Network Structure

4.3 Image Results

Experiment results are shown in figure(3), (4), (5) and (6). Especially, the compared pixel-to-pixel experiment is only implemented in label to image and image to label problems, shown in figure (3) and (4). And no ground-truth paired image for photo and cezanne problem in the dataset.

From image results, we found that my code works normally but the result is not perfect. Results from pixel-to-pixel experiment and cyclegan are similar.

4.4 Parameters Selection

In this part, I compare different value of λ in equation (2) for value equals to 10(default value) and 1(test value). The results is shown in figure(5) and (6). The results indicates that the network can not work with low λ value.

4.5 Loss Value Curve

In this part, we prove the important of different learning rate when training generative network and discriminate network and show the normal training loss in figure (7).

In the figure, the normal curve of training loss are green line and pink line, where the discriminate loss keep in 0.5 and generative loss keep decreasing. However, if we set learning rate for all two network as 1e-4, we obtain the blue line and orange line in the figure, where the discriminate network is trained too good. When the loss of discriminate network reach almost zero, the generative network is broken and can not be trained continuously.

4.6 Time Consuming

In total, one epoch have about 120 batches and takes about 5 minutes. All training process takes about 15 hours with 1000 epochs. The GPU is GeForce GTX 1080 Ti.

5 Conclusion

This unpaired image-to-image method designs a cycle structure(two generative network for mapping and reconstruction) based on general common GAN. Input image is firstly mapped into object image then be reconstructed into input image.

This cycle structure help generative learn how to map correctly without constraint l2-norm between paired image in loss function. Also, this novel idea can use information in the style or category from images without paired information from content or pixels.

The experiment results show that this idea can finish image-label and photo-cezanne and my code works normally. However, the result is not good enough.

References

[1] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context Encoders: Feature Learning by Inpainting. *Arxiv.org*, 4 2016.

[2] Phillip Isola, Jun Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 5967–5976, 2017.

[3] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 2242–2251, 2017.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Neural Information Processing Systems*, 6 2014.

[5] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares

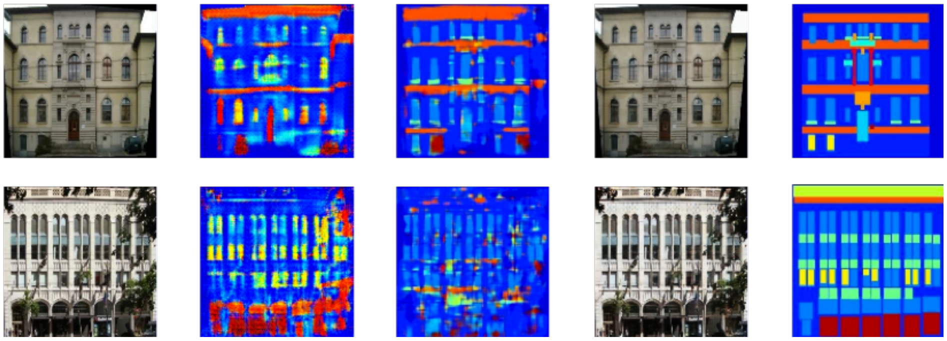

Figure 3: Image to Label. From left to right: input image x, output image $G(x)$, comparison pixel-to-pixel result, reconstruction image $F(G(x)$, ground-truth paired image y.

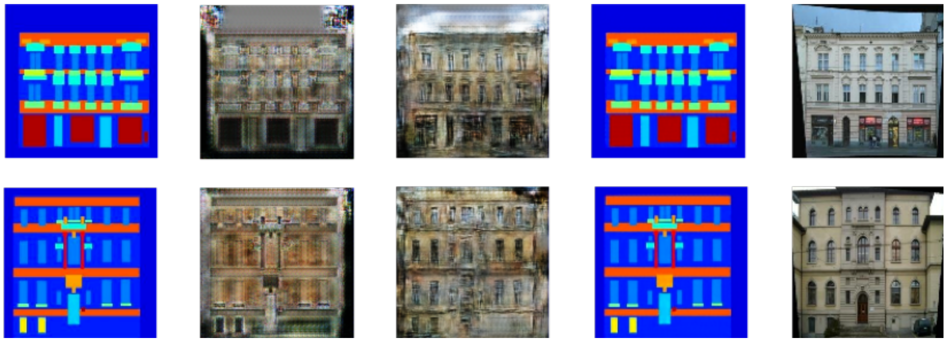

Figure 4: Label to Image. From left to right: input image x, output image $G(x)$, comparison pixel-to-pixel result, reconstruction image $F(G(x)$, ground-truth paired image y.

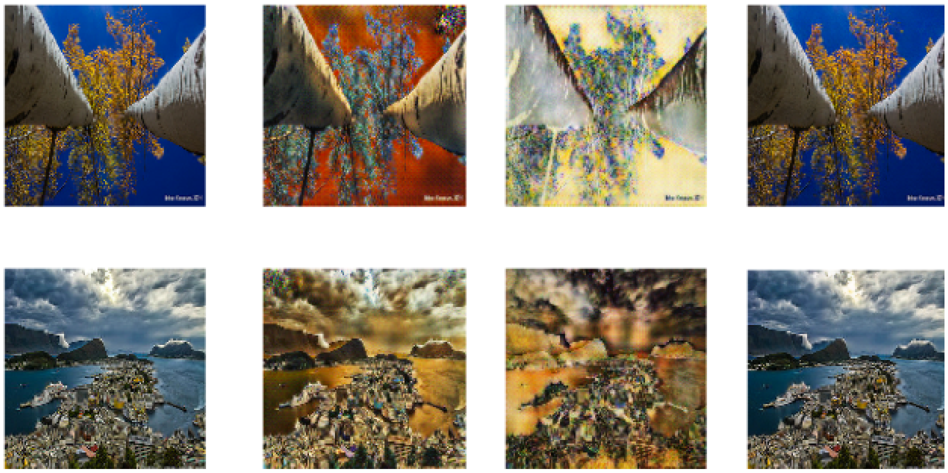

Figure 5: Photo to Cezanne. From left to right: input image x, output image $G(x)$, low value of λ result(see section 4.4), reconstruction image $F(G(x)$.

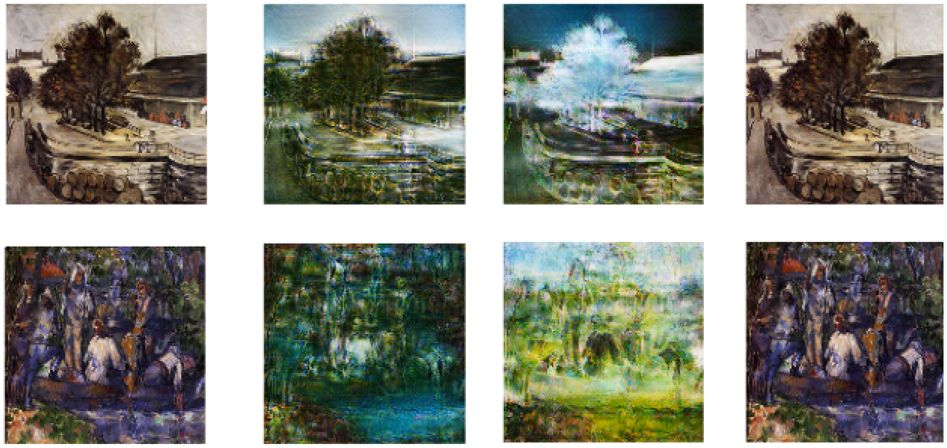

Figure 6: Cezanne to Photo. From left to right: input image x, output image $G(x)$, low value of λ result(see section 4.4), reconstruction image $F(G(x))$.

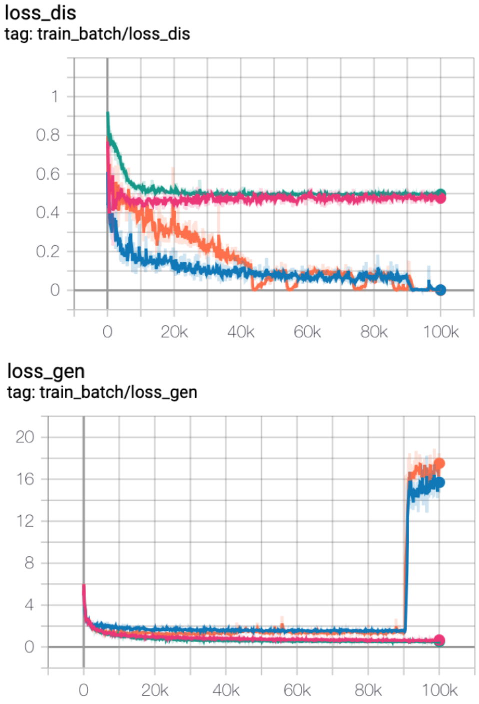

Figure 7: Loss Curve

Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2813–2821, 2017.