DLLAB WEEK 13 Pre-Report : Pix2Pix and CycleGAN

Hyunsoo Cha[1], Kwanghoon Sohn[2]
[1]Yonsei University. [2]Yonsei University.

1 Pix2Pix

1.1 Introduction

For image processing, there is a challenge in computer graphics and computer vision to translate input images and their corresponding output images. The goal of this task is to predict from pixel to pixel. The learning process is automated, but many manual effects are designed as effective lossss. Generative Adversarial Networks can be used to achieve this goal in that it produces outputs that are indistinguishable from reality. GANs learn losses to distinguish whether output images are real or fake. Generative models simultaneously minimize learning losses. A blurry image is a task that requires a very different form of loss function, not explicitly allowing it to be a fake image.[2]

In the paper, they implement a CGAN conditional generative model. The goal of this paper is to generate output images based on input images. Through this paper, conditional GAN can solve a wide range of challenges and analyze important architecture effects.

1.2 Related Works

They address structured loss in various papers, including ssim, nonparametric loss, conditional random field, and so on. output space is unstructed. Conditional GANs have never been used for video. It became the first case to be attempted in this paper.

1.3 Method

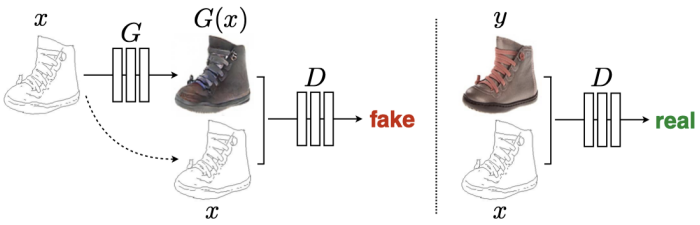

Figure 1: Figure 1 shows a schematic diagram of the conditional GAN from edge to photo. D is a discriminator that determines whether the input image is fake or genuine. Generator G is the ultimate goal of deceiving D.

GAN is a generative model that learns output image y generation from random noise vector z. In contrast, CGAN learns y which is an output image from z and x. Expressing this as a mapping function, $G : x, z \rightarrow$.

Objective Function

$$\mathcal{L}_{\mathbf{cGAN}}(G,D) = E_{x,y}[\log D(x,y)] + E_{x,z}[log(1 - D(G(x,z)))] \quad (1)$$

G is implemented to minimize objective when adversarial D tries to maximize. So, $G^* = \arg\min_G \max_D \mathcal{L}_{\mathbf{cGAN}}(G,D)$ is satisfied. The discriminator does not observe x. Assuming that there is no x, they test the importance of the condition under which the discriminator is at stake. The researchers found that L1 rather than L2 produced a blurred image. The final obejctive function is:

$$G^* = \arg\min_D \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G) \quad (2)$$

No distribution can be learned unless it is delta function.

Network Architectures Both models used convolutional layer, batch norm, and ReLU[4]. In the paper, generators and discriminators are applied to GANs. Previous work has used a large number of encoder-decoder networks, but it is inevitably down-sampled after passing through multiple

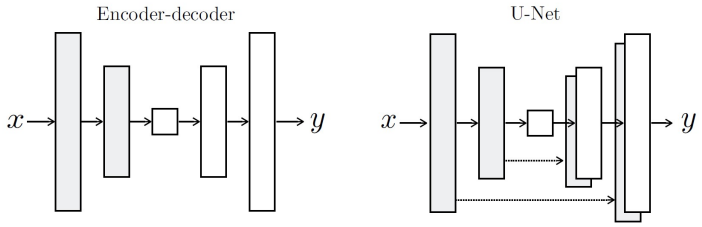

Figure 2: The figure above shows two options for the structure of the generator. A skip connection is applied to the encoder-decoder structure called U-Net.

layers. They add a skip connection between each layer, i and $n-i$. What n means is the total number of floors.

Markovian discriminator (Patch GAN) L1 and L2 loss produce blurry results[7]. This loss function detects low-frequency components well and does not detect high-frequency components. Therefore, the researchers decided to use the structure of the local image patch to enable good detection of high-frequency components while exploiting $L1$ loss function as it is. Patch GAN is a GAN that plays a role in penalizing structure. This discriminator serves to classify whether $N \times N$ patches are real or fake. The researchers applied this discriminator by averaging all responses for output D.

Optimization and inference The paper discusses how to optimize the network using GAN's standard method. The difference compared to conventional networks is that gradient description steps are added to D, G, one step at a time. Furthermore, the researchers divided the objective function by 2 while optimizing D. The researchers applied minibatch SGD to Adam-solver. Learning rate is 0002 and momentum parameter $\beta_1 = 0.5, \beta_2 = 0.999$. In the experiments of the researchers, batch size was used from 1 to 10 depending on the experiment.

1.4 Experiments

To understand the comprehensiveness of conditional GANs, graphics tasks such as semantic segmentation, vision, and photo generation were applied. It can be seen that input and output have one to three channel images.

Figure 3: The different results of Loss are shown in the table above.

Evaluation metrics To quantitatively analyze the quality of the synthesized image, per-pixel mean-squared error may be used, but the structure is poorly measured. Therefore, Amazon Mechanical Turk conducts a real vs fake investigation. The second is to use cityscape to make sure that the

object is correctly recognized. **Analysis of the objective function** Experiment with the results of GAN and L1 in the loss function formula. For the task of making photographs on two labels, they confirm that the results used as L1 are blurry. Although cGAN showed a more definite domain and better image, it can be seen that it produced an anantifact. With the addition of the two losses, the anti-fact disappeared and became closer to groundtruth. It is clear that cGAN performs better than GAN[3], and the addition of L1 can lead to better results. **Colorfulness** Conditional GANs creates the illusion of spatial structures that do not exist in input label maps. CGAN can be applied for cityscape data. **Analysis of the generator architecture** Encoder-decoder makes skip connections available for U-Net. However, the encoder-decoder was not able to create a realistic image in this experiment. **From PixelGANs to PatchGANs to ImageGANs** PixelGANs were 1×1 and ImageGANs were measured at 286×286. PixelGAN appears to be more effective in colorfulness than shapelessness. In the case of 70×PatchGAN, they find that artifact is relaxed, and they can see that they used L1+cGAN Loss.

2 CycleGANs

2.1 Introduction

As shown in the figure, CycleGAN's ultimate goal is to image transition from source domain X to target domain Y in the absence of training data through unsupplied learning.

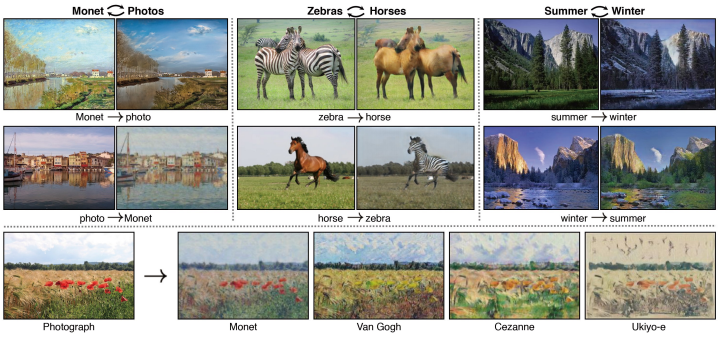

Figure 4: The above illustration shows an unordered collection of X and Y. Changing natural photographs into each painting style is what the picture above represents.

It is a technology that automatically transforms from horses to zebras to landscapes to Monet's paintings. This is accomplished by training dataset[1] through two image clusters. The target distribution Y trains through an adversarial loss that generates the distribution G(x). Adversarial losses result in the absence of binding force. Therefore, they guarantee that $F(G(x))$ is restored to x via cycling loss function. It's a sort of inverse functional relationship.

For training, they extract the properties of the image and learn how to change it. The case where the image x_i corresponding to the image y_i is passed is called pair and supervision learning. Conversely, when an uncorrelated image y is passed, it is called unpair and called unsupervised learning.

2.2 Cycle-Consistent Adversarial Networks

Specifically, they discuss how to learn the relationship of two domain X, Y as a training dataset through two functions:

Both models have two discriminators and observe whether all inputs are mapped to similar outputs.

2.3 Formulation

Adversarial Loss The loss function used by source domain X as a transformation of target domain Y is represented as follows.

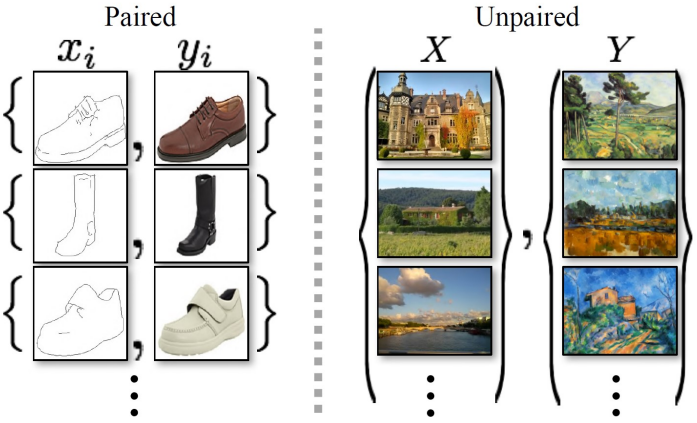

Figure 5: Paired training dataset and unpaired training dataset can be seen above figure. only source and target set has configured.

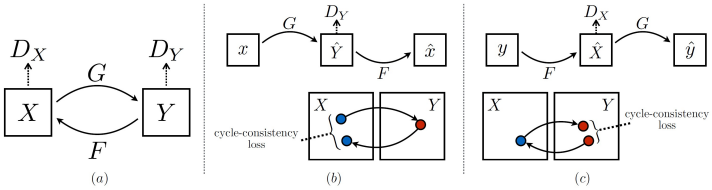

Figure 6: They show $G : X \rightarrow Y$ and $F : Y \rightarrow X$. Related rsarial discriminators is D_Y, D_X D_Y induces G to perform an X transformation from domain Y to an indistinguishable output. And the opposite is D_X, F.

$$\mathcal{L}_{\mathbf{GAN}}(G, D_Y, X, Y) = E_{y \sim p_{textbf data}(y)}[\log D_Y(y)] + E_{x \sim p_{\mathbf{data}}(x)}[\log(1 - D_Y(G(x)))] \tag{3}$$

G is a $G(x)$ generator that makes images close to target domain Y. D_Y is generated image that $G(x)$ is real image Y to find discriminator. The discriminator ensures that the objective function is maximum, but also that the image generated by the generator is minimal so that it can be mistaken for real. If the ratio of real to fake is 1:1, the following expression is created:

$$\mathcal{L}_{\mathbf{GAN}}(G, D_Y, X, Y) = \frac{1}{M} \sum_{y, y'} (p_{\mathbf{data}}(y) \log D_Y(y) + (1 - p_{y'}(y') \log(1 - D_Y(y')))) \tag{4}$$

Full Objective The final objective function[6] is:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\mathrm{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\mathrm{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\mathrm{cyc}}(G, F) \tag{5}$$

2.4 Result

Compared to fix2pix, SimGAN, Feature Loss GAN, ALI, BiGAN, and coGAN, the following results are shown by cycleGAN. The application of cycleGAN is characterized by the ability to imitate the entire work group, such as the natural style transfer. However, there are also some examples of failures, and transforming shapes is a difficult situation.

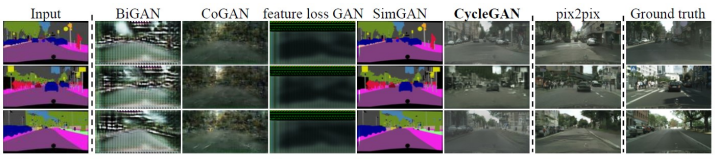

Figure 7: Comparative analysis photos of BiGAN, coGAN, feature loss GAN, simGAN, CycleGAN, and pix2pix[5] for cityscapes images are shown above.

3 Reference

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[2] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

[3] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.