Image Style Conversion using Cycle-Consistent Adversarial Networks

16341023 Fu-Rong Zhou
School of Data and Compute Science
Sun Yat-Sen University
543536324@qq.com

Abstract

Image style conversion refers to the conversion of image content from one domain to another.This tasks generally require paired pictures with the same content in the both domains as training data. Such as pix2pix[4], but this pair of training data is difficult to obtain. The innovation of CycleGAN is that can migrate picture content from the source domain to the target domain without paired training data. Paper"Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks"[13] presents an approach (Cycle-GAN)for learning to translate an image from a source domain to a target domain in the absence of paired examples. Quantitative comparisons against prior methods demonstrate the superiority of their approach.

1. Introduction

The original author realized the conversion of image style through CycleGAN: zebra to horse, summer to winter interchange. Can we apply this technology to the automatic conversion of the portrait of a comic character to the costume, and generate a new anime character through the original anime character's avatar and costume, which will greatly reduce the illustrator's design of a new anime character. time.

The core idea of CycleGAN is that if there is an image style converter G that can convert the image of the X domain into the style of the Y domain, and F can convert the image of the Y domain into the style of the X domain, then G and F should be reciprocal. That is, after the image of the X field is converted to $\hat{Y}$ by G, $\hat{Y}$ should be converted to X by F. Similarly, after the image of the Y field is converted to $\hat{X}$ by F, $\hat{X}$ should be converted to Y by G. That is: F(G(x))=x; G(F(y))=y. To implement this Cycle Consistency, the paper uses a Cycle Consistency Loss[12]:

$L_{cyc}(G,F) = E_{x\sim p_{data}(x)}[\|F(G(x)) - x\|_1]$
$+ E_{y\sim p_{data}(y)}[\|G(F(y)) - y\|_1]$[13]

CycleGAN is a circular structure consisting of two generators and two discriminators. X represents the image in X domain and Y represents the image in Y domain. The image in X domain is generated by generator G in Y domain, and then the original image input in X domain is reconstructed by generator F. The image in Y domain is generated by generator F, and the original image input in Y domain is reconstructed by generator G. Discriminator Dx and Dy play a discriminant role to ensure the style migration of images.

CycleGAN is essentially two mirror symmetrical GANs[3], forming a ring network. Two GANs share two generators with a Discriminator. One-way GAN has two losses, and CycleGAN has four losses. As shown in the figure 1 , CycleGAN is actually composed of two discriminators (Dx and Dy) and two generators (G and F), in order to avoid all X being mapped to the same Y, in order to avoid this situation, we adopts two generators, which can satisfy the mapping of X $\rightarrow$ Y and satisfy The mapping of Y $\rightarrow$ X, in fact, is the idea of variational self-encoder VAE, in order to adapt to different input images to produce different output images.

1.1. Related work

As researchers at home and abroad pay more and more attention to image migration, A series of Cyclegan algorithms have emerged. In this section, we briefly introduce four related work. (Generative Adversarial Networks (GANs) Image-to-Image Translation Unpaired Image-to-Image Translation Cycle Consistency)

Generative Adversarial Networks (GANs) [3, 11] have achieved impressive results in image generation[1, 8], image editing [14], and representation learning[8, 9]. The key to the success of GAN is the concept of confrontational loss, which forces the generated image to be indistinguishable from the real image in principle.

Image-to-Image Translatio: The idea of image to image conversion is to use a non-parametric texture model[2] on a single input-output training image pair. The latest method uses the data set of input and output checks to learn the parameter conversion function through CNN[7]. Our approach is based on the "pix2pix"[4] framework of Isola et al[5]. It uses conditional generation of the network to

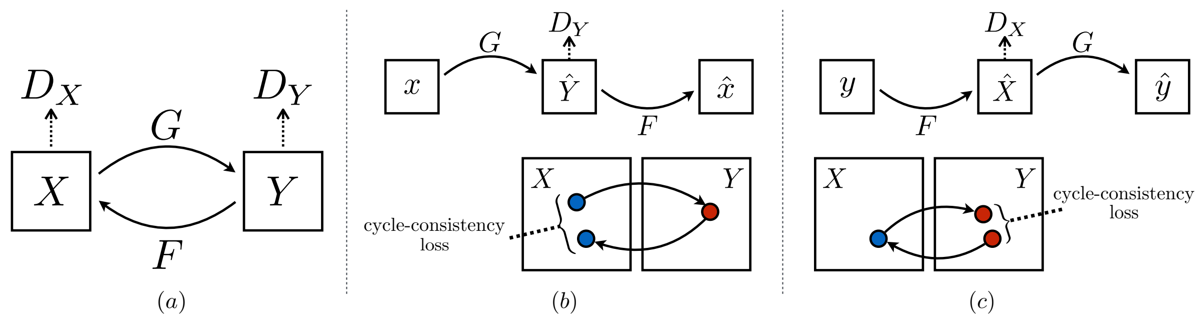

Figure 1. CycleGAN is actually an AB one-way GAN plus a BA one-way GAN. The two GANs share two generators, each with a discriminator, so adding up to a total of two discriminators and two generators. A one-way GAN has two losses, and CycleGAN adds up to a total of four losses.

learn the mapping from input to output images.

Unpaired Image-to-Image Translation: The similarity function between the predefined inputs and outputs that does not depend on any particular task, nor does it assume that we assume that the inputs and outputs must be in the same low-dimensional embedded space. This makes our approach a universal solution for many visual and graphical tasks.

Cycle Consistency: In target tracking, language translation, 3D shape registration and so on. There are also applications in CNN. A similar loss was introduced to push G and F to agree with each other.

2. technical background

2.1. formula :

Adversarial LossFor the mapping function G : X Y and its dis- criminator DY , we express the objective as

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

$$\mathcal{L}_{\text{GAN}}(G, D_X, X, Y) = \mathbb{E}_{x\sim p_{\text{data}}(y)}[\log D_X(y)] + \mathbb{E}_{y\sim p_{\text{data}}(x)}[\log(1 - D_X(G(x))] \quad (2)$$

Cycle Consistency Loss. The so-called Cycle consistency is to ensure

Forward consistent: x →G(x)→F(G(x))x

Backward agreement: y→F(y)→G(F(y))y

We incentivize this behavior using a cycle consistency loss:

$$\mathcal{L}_{\text{cyc}}(G, F) = mathbbE_{x\sim p_{\text{data}}(x)}[[F(G(x)) - x]_1] + \mathbb{E}_{y\sim p_{\text{data}}(y)}[[G(F(y)) - y]_1] \quad (3)$$

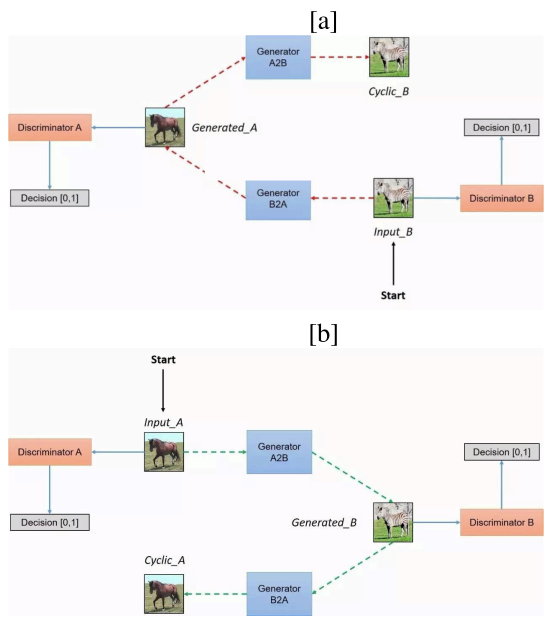

Figure 2. The model acquires an input image from the domain DA, which is passed to the first generator GeneratorAB, whose task is to convert a given image from the domain DA to an image in the target domain DB. This newly generated image is then passed to another generator, GeneratorBA, whose task is to convert back to the image CyclicA in the original domain DA, which can be compared to the autoencoder. This output image must be similar to the original input image and is used to define meaningful mappings that did not exist in the unpaired data set.

full objective loss function is:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda\mathcal{L}_{\text{cyc}}(G, F) \quad (4)$$

Depth generation model :

The network architecture of CycleGAN is shown in the figure2:

2.2. Algorithm flow

k, is a hyperparameter.We used k = 1, the least expensive option, in our experiments

Require: Image A dataset **R**, Image B dataset **X**, number of iterations n

for $i = 1$ to n **do**

  for $i = 1$ to k **do**

    Sample minibatch of m noise Samples from noise prior Pg(z)

    Sample minibatch of n examples from data generationg districution Pdata(x)

    Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} 1/M \sum_{n=1}^{M} [\log D(x^i) + \log(1 - D(G(z^i)))] \tag{5}$$

  end for

  Sample minibatch of m noise Samples from noise prior Pg(z)

  Update the discriminator by decending its stochastic gradient:

$$\nabla_{\theta_g} 1/M \sum_{n=1}^{M} \log(1 - D(G(z^i))) \tag{6}$$

end for

2.3. Experimental data set

The source of the experimental data set is the data set provided by the original author, the source of the data set:`https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets/`

2.4. Experimental result

The experimental results are as shown picture34

2.5. Performance comparison of different image migration algorithms

As shown in the table1, Cyclegan basically realizes the conversion between unpaired pictures, but the image conversion effect is slightly worse than other image migration algorithms. The results obtained from the experiment do not affect the intuitive judgment of the human eye.

2.6. Effect of algorithm parameters

When obtaining the optimal parameters of the model, we use the small epoch to continuously test the optimal parameters of the approximation model. Finally, the learning rate is best around 0.002.

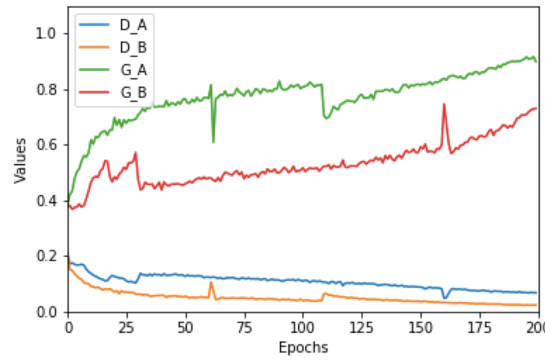

Figure 3. As the epoch continues to increase, the accuracy of GA and GB keep rising and gradually stabilizes. the stable values of D_A is about 0.83, the stable values of D_B is about 0.76.

Figure 4. Basically, the conversion between unpaired images and images has been implemented, but the conversion effect is not very good, and the converted image is not smooth enough. In the pixel processing, our model still needs improvement.

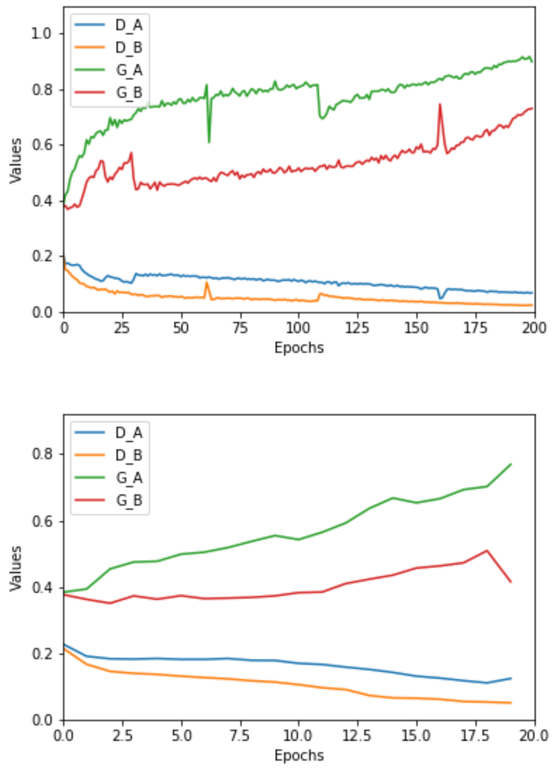

Figure 5. The epoch value increases and the value increases when the accuracy reaches stability.

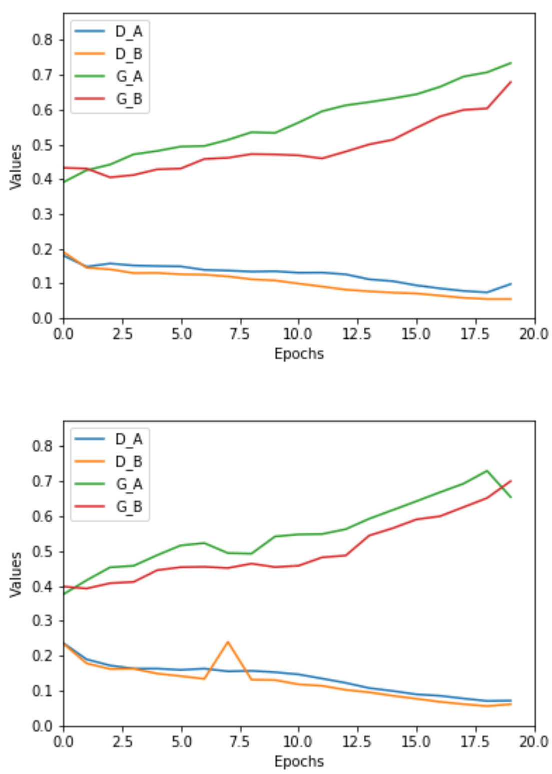

Figure 6. When the learning rate is equal to 0.003, there is an overfitting phenomenon. When the learning rate is 0.0015, the image is slightly worse than the learning rate of 0.002.

Table 1.

	Map to Photo	Photo to Map
	Error rate	Error rate
CoGAN[6]	1.1%	1.4%
SimGAN[10]	1.6%	2.8%
CycleGAN(our)	23.6%	22.4%

3. conclusion

In this report, we implement the code of "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". The success of our cyclegan model is due to the standing of the work of our predecessors. This recurring experiment gives us a deeper understanding of GAN and encourages us to focus on our deep learning.

4. Code Implementation

Code implementation is a fine-tuning, referring PyTorch-GAN and pytorch-CycleGAN-and-pix2pix

Since some code lines are too long, the cvpr template cannot display the code properly. The code item address is given below. `https://github.com/zfr0411/CycleGAN`

The folder where the code is located: code

References

[1] E. L. Denton, S. Chintala, R. Fergus, and et a. Deep generative image models using a laplacian pyramid of adversarial networks. *NIPS,*, 2015.

[2] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. 1999.

[3] I. Goodfellow, J. Pouget-Abadie, B. Xu M. Mirza, D. Warde-Farley, S. Ozair, A. Courville, and Y. Ben-gio. Generative adversarial nets. *NIPS*, 2014.

[4] JP. Isola, J.-Y. Zhu, T.Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. 2017.

[6] M.-Y. Liu and O. Tuzel. Coupled generative adversar- ial networks. 2016.

[7] J. Long, E. Shelhamer, , and T. Darrell. Fully convolutional networks for semantic segmentation. 2015.

[8] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional gen- erative adversarial networks. *ICLR*, 2016.

[9] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. 2016.

[10] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. 2017.

[11] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *ICLR*, 2017.

[12] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d- guided cycle consistency. *CVPR*, 2016.

[13] Jun-Yan Zhu, Taesung, Park Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CVPR*, 2017.

[14] J.-Y. Zhu, P. Kra henbu hl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. *ECCV*, 2016.