Comparative Analysis of Pix2Pix and CycleGAN for Image-to-Image Translation

Eason Lin

University of Washington, Seattle, United States

yichl42@uw.edu

Abstract. Image-to-Image translation technology is nowadays a prevailing research orientation in computer-vision domain, which aims to translate the styles and features in images from one image domain to another. With the rapid development of the convolutional neural networks, especially generative adversarial network technology, breakthroughs have been made in the performance of image translation, which has been widely used in many fields such as labeling photos to synthesize photos, reconstructing objects from line drawings, and coloring pictures. The Image-to-Image Translation problem is essentially a pixel-to-pixel mapping. Limited by specific task settings, the performance of different general frameworks often fluctuates greatly when dealing with different translation tasks. In this paper, focusing on the above problems, the advanced and commonly used Image-to-Image Translation frameworks such as Pix2Pix and CycleGAN, are selected to compare and analyze the advantages in detail from different dimensions including network structure, loss function, applications and model accuracy. Furthermore, for different practical application scenarios, we discuss solutions based on these two representative frameworks and show the results after image translation processing. Finally, the existing problems and future research directions are discussed and summarized.

Keywords: Pix2Pix; CycleGAN; GAN; Image-to-image Translation.

1. Introduction

In the research fields of image processing and computer graphics, many problems could be thought of as "transforming" input pictures into the output ones. For instance, a scene can be rendered as an RGB image, gradient domain, edge or semantic map, and so on. In this context, Image-to-Image Translation has gradually become a widely-discussed research topic, which aims to translate the styles and features in images from one image domain to another. Specifically, it can be seen as removing a certain attribute X of the original image and giving it a new attribute Y, that is, the cross-domain conversion between images. At present, Image-to-Image Translation has been widely used in many fields such as synthesizing photos by labeling pictures, reconstructing objects from line drawings, and coloring images.

With the rapid development of convolutional neural networks, they have become the cornerstone for solving various image prediction tasks. For different vision tasks, convolutional neural networks design specific loss functions to penalize the difference between predictions and ground truth. Even though the learning procedure is automated, myriads of human effort is required to design an effective loss function as before. In contrast, if we only need to specify a high-level goal, such as "generate an output that is indistinguishable from real images", and then automatically learn a loss function that fits the goal, this is very satisfying. Fortunately, this is exactly what the recently proposed Generative Adversarial Networks GANs do. The Image-to-Image Translation task essentially learns a pixel-to-pixel mapping, which happens to be described as a class of image generation problems. In this case, the image content is the inherent content of the image and is the basis for distinguishing different images. The image domain, that is, a group of images with similar characteristics, can be considered as image content endowed with some of the same properties. If you see a picture of a cat, then the content of the picture is a specific cat; if you give the image the attribute of pencil drawing, then a "pencil cat" comes out.

However, the effect of Image-to-Image Translation often depends on the specific task setting, such as giving the image the attributes of pencil drawing or oil painting. In addition, the performance of

different general frameworks often fluctuates greatly when dealing with similar Image-to-Image Translation tasks. In response to the above problems, how to choose the most suitable Image-to-Image Translation method combined with the task remains an open issue. Representative Image-to-Image Translation methods mainly include Pix2Pix and CycleGAN. Pix2Pix implements image translation based on conditional GAN (cGAN). The cGAN is able to guide image-generation by supplementing conditional information. Therefore, in image translation, the input image could be conditioned to learn the mapping from the input to the output, which finally gain the specific output in this case. CycleGAN is committed to solving the problem of lack of paired training data in actual model training, whose basic idea is to transform an image from one domain to another domain, and then inversely transform it back. By doing this, the inversely transformed image can be the same as the original image.

Considering that different algorithms have specific advantages in specific parts, in the paper, we introduce the two most representative Image-to-Image Translation methods at first, Pix2Pix and CycleGAN, respectively. The advantages and disadvantages of the two methods are compared in detail from the perspective of loss function and practical application. Furthermore, for different practical application scenarios, we discuss solutions based on these two representative frameworks and show the results after image translation processing. Finally, the existing problems and future research directions are discussed and summarized.

2. Pix2Pix

Pix2Pix is a GAN model mainly designed and used for image-to-image translating. On the basis of conditional Generative Adversarial Network (cGAN) [1], Pix2Pix utilizes it to learn the mapping from an input image to an output image. It's worth mentioning that Pix2Pix is not only for specific application, but it could also be applied to wide-ranging tasks, including turning black and white images to color photos, translating pictures from multiple apps, generating facial images for face aging, and even converting sketches into photos.

2.1 Conditional Generative Adversarial Network

To solve issue that GAN is too irregular to handle, a traditional idea is adding several restrictions to GAN itself, thus the cGAN is coming out. The central idea of cGAN is to control the pictures generated by GAN, rather than generating pictures simply and randomly. Specifically, cGAN adds several additional condition information to the inputs of the generator and the discriminator, and the pictures which generated by the generator could pass the discriminator only if they are real enough and consistent with the conditions. As a matter of fact, in the generation model without conditional constraints, controlling the mode of data generation is relatively strenuous. However, by restricting the model with additional information, the cGAN is able to guide the data generation process properly.

As the formula (1) and (2) shown, cGAN is equivalent to adding a condition to both the generator and the discriminator based on the original GAN [2]. Realizing the purpose of cGAN, the principles and the training methods of generating and discriminating networks should be changed. In the model part, additional information y is added to both the discriminator and the generator. In this way, y can be a category label or other types of data, and it can be dropped into the discriminator and the generator as an additional input layer.As a traditional generative model, cGAN still has several defects, such as blurred image edges and low image resolution. Nonetheless, it leads the path for the subsequent GAN branches, especially Pix2Pix and CycleGAN.

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim P_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \quad (1)$$

$$\min_{G}\max_{D} V(D,G) = \mathrm{E}_{x\sim P_{data}(x)}[\log D(x\backslash y)] + \mathrm{E}_{z\sim P_z(z)}\left[\left(1-D\left(G(z\backslash y)\right)\right)\right] \quad (2)$$

2.2 Structure of Pix2Pix

2.2.1 Generator

The two vital parts of Pix2Pix are the generator (U-Net) and the discriminator (PatchGAN) [3]. Compared with traditional Encoders-Decoders, U-Net could improve the performance of image-to-image translation, making it more widely used. Generally, U-Net is a typical type of Convolutional Neural Network (CNN), and is also a deep learning representative algorithm. The structure of U-Net is shown in Figure 1. Every small blue box relates to a multi-channel feature map (different colors), and the amount of those channels is indicated at the top of the box. The x-y dimension is located at the lower left edge of the box, and the white box represents the copied element map. Moreover, the arrows show the different operations. In the left side, the contracting path which composed of the convolution layer down-samples the data while extracting the information. And in the right side, the expansive path which composed of an upper transposed convolution layer up-samples the information [4].

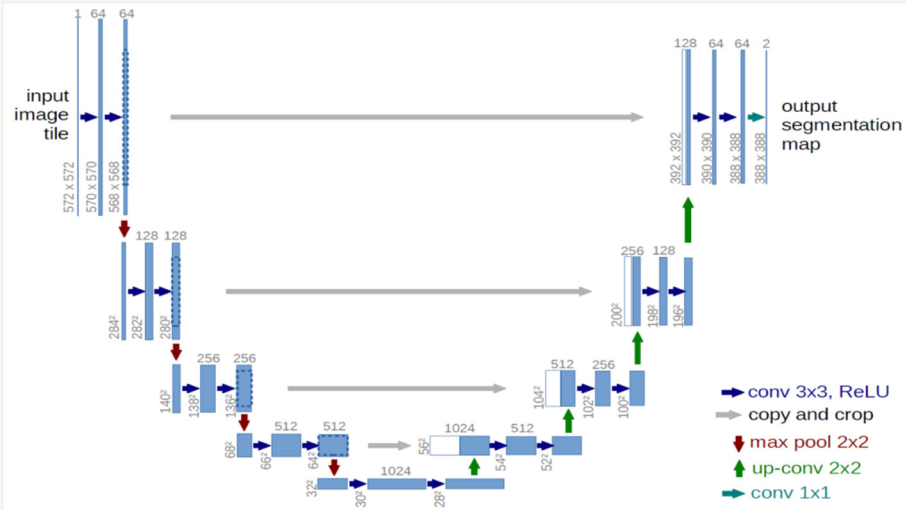


Figure 1. U-Net structure

Figure 2. shows the Encoder-Decoder of Pix2Pix. The inputs are a chain of encoders and the outputs are a chain of decoders. From these different sizes of volumes, it is not that hard to see the shape of the tensor dimension next to them. In this example, the input is a 256 x 256 x 3 image with only black and white, and the output is the image with three color channels (red, green, and blue). The generator in this case accepts some inputs and tries to reduce it to a smaller representation using a series of encoders. By compressing the data, the researchers hope to reach a higher level of data representation after the final coding layer. Meanwhile, the decoder layer performs the opposite operation and reverses the action of the encoder layer.

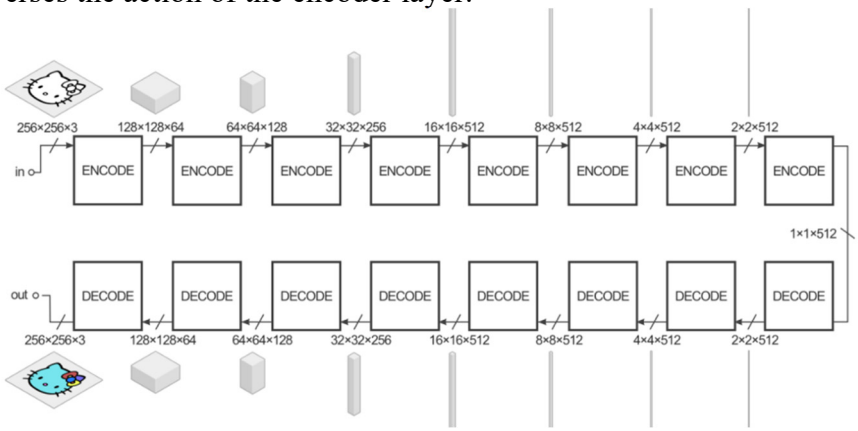


Figure 2. The structure of the Encoder-Decoder of Pix2Pix

2.2.2 Discriminator

The discriminator of Pix2Pix mainly used the PatchGAN, which could better judge the local part of the image. In this way, it divides the image into patches equally, judge the authenticity of each patch separately, and finally take the average. In other words, PatchGAN could be regarded as another texture or style loss form. When judging, the efficiency would be extremely low if the picture is too large. Therefore, choosing the appropriate size of the patch is also essential. Figure 3 shows the main function of the PatchGAN: rather than judging through images entirely, judging them by parts separately. It takes the $N\,x\,N$ portion of the picture and attempts to find out the accuracy of the image. On this condition, N may be any random number, and it could be even smaller than the original one but sill yield a relatively high-quality outcome. The discriminator convolution is applied to its entire image. Additionally, since the size of discriminator is relatively small, thus it has less parameters than generator, which is actually faster.

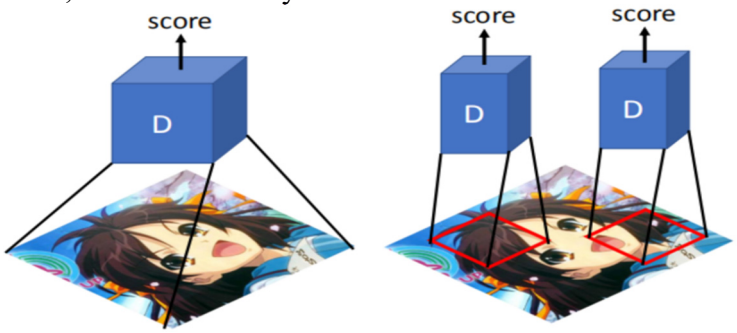


Figure 3. The score calculation of PatchGAN

2.3 Loss Functions

The loss function of Pix2Pix is showed below as:

$$L_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x,z}\left[\log\left(1 - D\left(x, G(x, z)\right)\right)\right] \quad (3)$$

The equation has two parts for the discriminator D and the generator G. By mixing the loss function with L1, generator G confuses discriminator D, as well as generates an image similar to the ground truth. For this sake, an additional L1 loss of the generator has been added to the loss function.

$$L_{L1}(G) = E_{x,y,z}[\|y - G(x, z)\|_1] \quad (4)$$

And finally, the loss function is represented by:

$$G^* = arg\min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G) \quad (5)$$

However, the L1 loss could not catch highly-frequent details. Therefore, blurred images remain occur and the PatchGAN should be used in this case.

2.4 Pipeline of Pix2Pix

It can be seen (in Figure 4) that for the generator G, the researchers input a contour map of a shoe as a priori condition x[5]. And for the discriminator D, they input the generated image $G(x)$ of the generator G with the a priori condition x. Furthermore, the real input image is represented by y, which relates to the edge image x. In this way, Pix2Pix requires paired images (x and y) during training. In general, x is taken as the input of the generator G and forms the generated image $G(x)$, then x and $G(x)$ are combined together based on the channel dimension. Finally, the predicted probability is gathered as the input image of the discriminator D, which indicates whether the input is a pair of real images or not. Moreover, considering the channel size, the real images y and x are combined in this case. Therefore, the training goal of the discriminator D is to output a small probability value when the input is the unpaired real images [6]. On the other hand, the training target of the generator G is to enlarge the predicted probability as large as possible. At the same time, the generated image $G(x)$ and original image x are utilized as the inputs of the discriminator D, thus cheat the discriminator D.

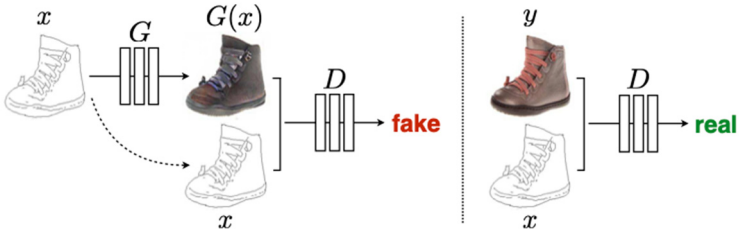


Figure 4. Pipeline of Pix2Pix

3. CycleGAN

CycleGAN is a useful model for training deep Convolutional Neural Networks (CNN) for image-to-image translation tasks. which is characterized by first converting the image from one domain to another through a cycle, and then back again. Through such a cycle, CycleGAN pairs the pictures before and after the conversion, which is similar to supervised learning and improves the conversion effect. An important application field of CycleGAN is the domain adaptation, such as changing an ordinary landscape photo into an artist's work, or converting the virtual picture into a real-world image [7].

Structure of CycleGAN

3.1 Encoder-Decoder

CycleGAN mainly uses ResNet as the generator. The structure of the generator is quite simple: dividing them into the encoder structure and the decoder structure. To be more specific, the decoder part mainly uses CIL (Convolutional, InstanceNormalization, and Leaky ReLU), and also utilizes the image optimization method (ReflectionPad2d) to symmetrical the image up, down, left, and right along the edge to increase the resolution of the entire image. And for the encoder part, it uses CTIR (ConvolutionalTranspose, InstanceNormalization, and ReLU) to restore the size of the image. Finally, the image resolution is improved by ReflectionPad2d, and restored to the original size of the image by convolution, which effectively solves the edge information of the object.

3.2 Discriminator

In the paper of [8], the authors tested the effects of using different patch sizes N (N × N) in the discriminator receiving domain: from 1×1 PixelGAN to the entire 256×256 ImageGAN. From the changes of patch sizes, the uncertainty of the output indicates that the difference lies in the different loss functions. At L1, the uncertain area becomes blurred and diluted. On this condition, a 1×1 PixelGAN leads to a larger color difference, but ineffective on spatial statistics; a 16×16 PatchGAN creates a locally clear result, but then generates a tiled artificial image after exceeding its observable scale; and a 70×70 PatchGAN forces output to be much clearer, even if it is not correct in the spatial dimension and the spectral dimension. Therefore, the discriminator adopts the PatchGAN with a patch size of 70×70.

3.3 Loss Functions

The loss function of the CycleGAN consists of two parts: LossGAN and Losscycle, which $Loss = Loss_{GAN} + Loss_{cycle}$. Separately, LossGAN ensures that the generator and the discriminator evolve mutually, so as to assure that the generator could produce more realistic images. On the other hand, Losscycle makes sure that the output image of the generator is different from the input image only in style, but the content is still the same. Specifically, they are:

$$Loss_{GAN} = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y) = E_{y \sim p_{data}(y)}[log\, D_{Y(y)}] + E_{x \sim p_{data}(x)}\left[log\left(1 - D_Y(G(x))\right)\right] + E_{x \sim p_{data}(x)}[log\, D_{X(x)}] + E_{y \sim p_{data}(y)}\left[log\left(1 - D_X(F(y))\right)\right] \quad (6)$$

$$Loss_{cycle} = E_{x\sim p_{data}(x)}[\|F(G(x)) - x\|_1] + E_{y\sim p_{data}(y)}[\|G(F(y)) - y\|_1] \quad (7)$$

3.4 Pipeline of CycleGAN

The main purpose of the CycleGAN is to realize the domain adaptation, and the picture above is a good example. In this case, domain x and domain y are two datasets to store landscape photos and the artist's painting respectively. The target is to train a generator G to make it consume a landscape photo and create an artist's work. At the same time, another generator F should be trained to consume an artist's work and create a landscape photo, that is Y. The researchers also need to train two *discriminators* D_x and D_y in order to judge the quality of the pictures generated by the two generators G and F. Under this circumstance, if the pictures generated by the generators are not like the image y in the data set Y, the discriminator D_y should give them a low score. Otherwise, if the pictures generated by the generators are like the image y in the dataset Y, the discriminator D_y should give them a high score. In addition, the discriminator D_y should always give high scores to real images y, and the discriminator D_x is the same as well. The whole model structure of the CycleGAN is showed in Figure 5.

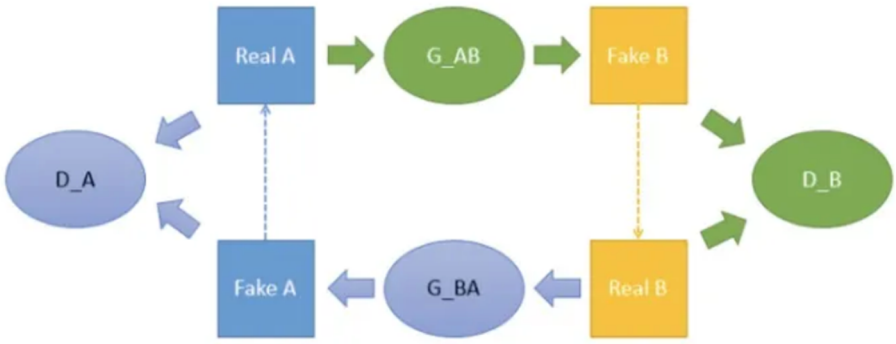


Figure 5. Pipeline of the CycleGAN

4. Comparison

4.1 Definitions

Pix2Pix and CycleGAN are both useful GAN models designed and used for image-to-image translation tasks. Nevertheless, one of the biggest differences between them is whether the data they used is paired or not. Specifically, Pix2Pix needs the data to be well-paired, but CycleGAN doesn't need paired data. In other words, unpaired data is suitable for CycleGAN but not for Pix2Pix. Considering these, a question (1) arises: since the CycleGAN does not need the data to be paired, does that mean CycleGAN is more useful than Pix2Pix? To this end, we analyze the advantages in detail from different dimensions including network structure, loss function, application range and model accuracy.

4.2 Applications

Based on paired input images, Pix2Pix is able to produce the corresponding outputs. Nevertheless, the output outline should be exactly the same as the input image, except for the filling information of the output image. As for the CycleGAN which the domain adaptation is a significant application filed, it could do several straightforward migrations such as image styles, image coloring, objects conversion and etc. By comparing the restored image with the original image, the accuracy of the generated image and its relevance to the input image can be ensured. In general, Pix2Pix is expert in directly convert images on the basis of paired data, and CycleGAN is skilled in translating between

images by inputting a set of data (such as a bunch of images of cats) and outputting a set of data (such as a bunch of images of dogs).

Pix2Pix could ensure that the output image is related to the input image. In other words, given an input image, the output one could not only meet the characteristics of the real image, but also reflect and retain the original information of the input image. Moreover, it ensures the relationships between the input and output data through the well-paired data set. There is no doubt that CycleGAN got its name because of the cycle above. Compared with the Pix2Pix which assures the relationships between input and output by optimizing the data set, the CycleGAN ensures by optimizing the structure of neural network. To be specific, CycleGAN adds a restorative network in view of the traditional GAN's network structure, which is often used to restore the output. The pixels of the restored image in this case are compared with the original input pixels to make sure that the corresponding relationships between the input and output. Furthermore, both the generation and restoration process are the restoration steps after feature extracting. Therefore, the output and input are not necessarily identical in form, but in the deep-seated feature.

4.3 Model structure

4.3.1. Generator

In the original conditional GAN (cGAN), the generator adopts the Encoder-Decoder structure of first down-sampling "encoding" and then up-sampling "decoding". But in Pix2Pix algorithm, it utilizes U-Net as the generator since the U-Net structure uses multiple scale fusion for cross-layer connections. Admittedly, in traditional neural networks, more layers mean better networks. However, due to the vanishing gradient problem, the weight of the first layer cannot be updated correctly by back propagation. When the error gradient is back propagated to the front layer, the gradient is reduced by repeated multiplication. Therefore, with the increase of the middle layer of the network, its performance becomes saturated and starts to decline rapidly. At this time, the famous ResNet was born, and almost all the Gans (especially CycleGAN) utilizes the ResNet version generator with residual block as a component. Compared with U-Net, ResNet adopts skip connections, which helps to solve the problem of gradient disappearance by allowing alternative paths for gradient flow. Additionally, it uses identity functions, improving the performance of higher layers to be as good as that of lower layers.

4.3.2 Discriminators

Both the Pix2Pix and the CycleGAN adopt PatchGAN as their discriminators. However, they are still not completely identical. The main idea of the PatchGAN is: it is not necessary to input the entire images into the discriminator since GAN is only used to construct high-frequency information. Considering this, to decrease the number of parameters and accelerate the training speed, researchers let the discriminators to judge whether each patch with size $N \times N$ is true or false. The discriminator of PatchGAN outputs a 30×30 feature map, and each point in this map corresponds to a 70×70 size patch. While calculating the anti loss, the dimension of this feature map is reduced to take the mean value to measure the difference between each patch of the generated image and the real image. Under this circumstance, it is able to increase the clarity of the generated picture, and that is the reason why a huge number of GANs adopts this algorithm as their discriminators.

4.4 Loss Functions

The loss function of Pix2Pix is:

$$L_{cGAN}(G,D) = E_{x,y}[log\, D(x,y)] + E_{x,z}\left[log\left(1 - D\big(x, G(x,z)\big)\right)\right] \tag{8}$$

And finally becomes

$$G^* = arg\min_G \max_D L_{cGAN}(G,D) + \lambda L_{L1}(G) \tag{9}$$

Using L_1 regularization helps to make the generated image clearer, and the final G_loss is the maximum minimum game of the generator and the discriminator under regular constraints.

And the loss functions of CycleGAN is:

$$Loss_{cycle} = E_{x\sim p_{data}(x)}\left[\left\|F\left(G(x)\right) - x\right\|_1\right] + E_{y\sim p_{data}(y)}\left[\left\|G\left(F(y)\right) - y\right\|_1\right] \quad (10)$$

Which is essentially based on the Least Squares loss functions:

$$\min_D J(D) = \min_D \frac{1}{2} E_{x\sim P_r}[D(x) - a]^2 + \frac{1}{2} E_{z\sim P_z}\left[D\left(G(z)\right) - b\right]^2 \quad (11)$$

And

$$\min_G J(G) = \min_G \frac{1}{2} E_{z\sim P_z}\left[D\left(G(z)\right) - c\right]^2 \quad (12)$$

It's worth mentioning that there is no sigma layer at the end of the discriminator in the Least Squares GANs (LSGANs). In this case, when an image is inputted into the discriminator of the CycleGAN, the output maybe any value (not necessarily between 0 and 1).

Compared with Pix2Pix, CycleGAN adopts Least Squares Loss to relatively improve the quality of generated images which incisively aim at the defects of low-quality images generated by standard GAN and the unstable training processes.

4.5 Training Results

After training, the CycleGAN loss over time is shown in Figure 6. On this condition, the two generators (G_A, G_B) and the two discriminators (D_A, D_B) all play a role in the training process.

Based on this graph, CycleGAN_A reaches higher loss when the epoch is relatively low, and generally the entire CycleGAN have a better effect when epoch = 2. When epoch changes from 0 to 3, the loss of CycleGAN_A drops from approximately 5 to 2; and the loss of CycleGAN_B first drops, then rises when epoch = 2 and epoch = 3. In the case of epoch = 5, CycleGAN_A gradually reaches loss = 1 and CycleGAN_B varies within the range between loss = 1 and loss = 2.

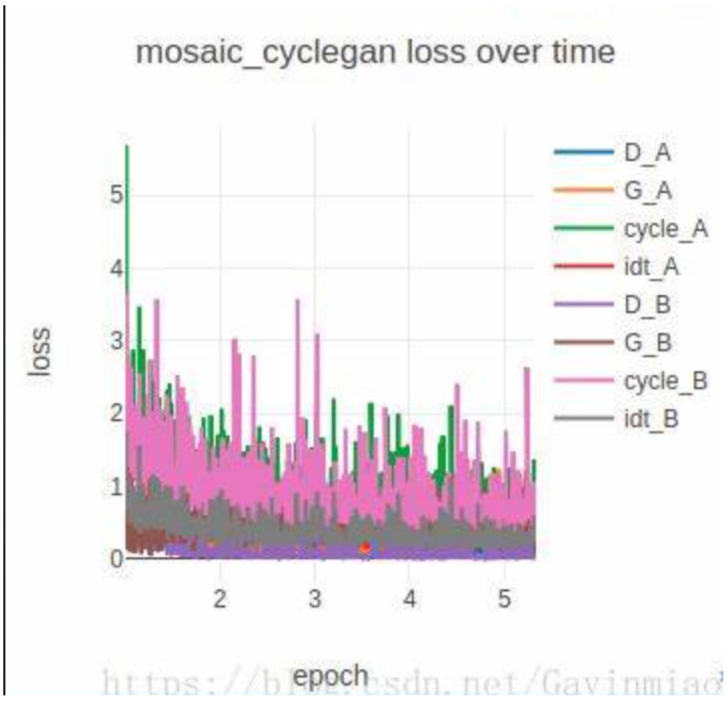


Figure 6. The training loss of CycleGAN

While the Pix2Pix loss is shown in Figure 7. As for Pix2Pix, L1 loss is a simple difference loss with strong robustness. However, reducing the diversity of the generated graph is easy, which tends to generate real image. Considering this, the generator of L1 loss (G_L1) reaches much higher losses than other generator. Compared with Pix2Pix, each part of CycleGAN reaches a certain loss with corresponding epoch. In general, CycleGAN has been more successfully used with a higher epoch than Pix2Pix.

According to this graph, specifically, the overall trend of G_L1 is within the range between loss = 20 and loss = 50. When the epoch changes from 0 to 2, the loss of G_L1 drops from 50 to 40 at first,

and then rises back to 50 again. Within the range between epoch = 2 and = 3, the G_L1 loss is unstable and varies from 35 to 50. As for G_GAN, the general trends of loss remain between 0 to 5 (all below 10) when the epoch changes from 0 to 3.

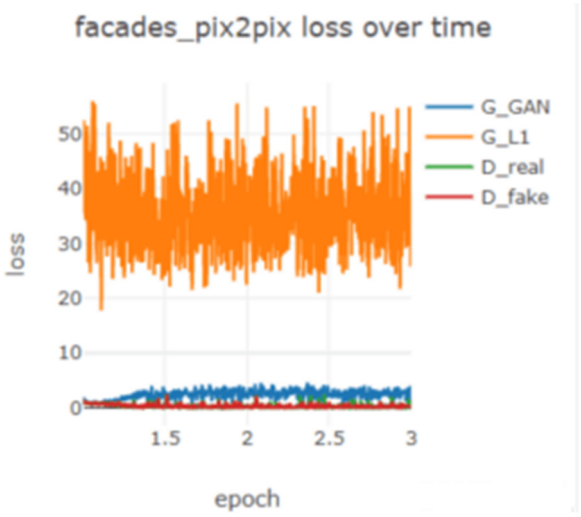
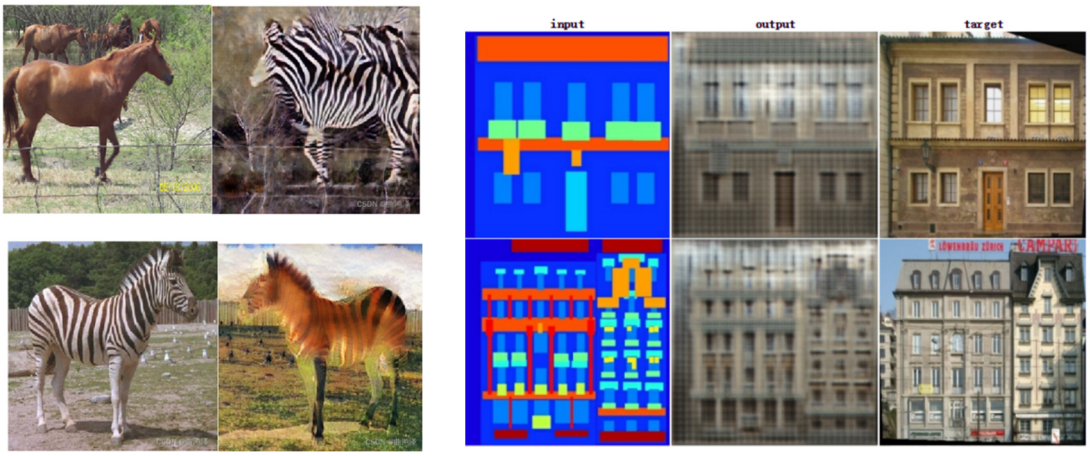


Figure 7. The training loss of Pix2Pix

The example of the CycleGAN and Pix2Pix output result can be found in Figure 8. Compared between these two algorithms, several differences could be easily seen. In fact, the image transformation refers to the conversion of objects from one category to another. Nevertheless, the sizes and the structures of the two images should be similar. In the example of CycleGAN above, it realizes the mutual conversion between zebra and horse images. As horses and zebras are similar in size and body structure, this image transformation is meaningful. On the contrary, however, Pix2Pix does not have such constraints translating between images. If the requirements are too high, then the details of the image generated by Pix2Pix would still not be clear enough, even if PatchGAN and random cropping are used in this case. The reason is that as the size of the generated image becomes larger and larger, the requirements for the abilities of GAN discriminators and generators would become higher and higher.



(a) CycleGAN output result (b) Pix2Pix output result

Figure 8. Image-to-image translation results of CycleGAN and Pix2Pix

5. Conclusion

This paper introduces Pix2Pix and CycleGAN from several different dimensions and compares them in details [9]. Specifically, These two algorithms are both widely used in image-to-mage translation tasks, but they are still different in several ways: Pix2Pix needs the data to be well-paired,

yet CycleGAN is able to use the unpaired data; Pix2Pix is more good at directly converting images, yet CycleGAN is expert in translating images between inputting and outputting data; Pix2Pix ensures the relationships between input and output by optimizing the data set, yet CycleGAN assures the relationships by optimizing the neural network; Pix2Pix adopts U-Net as the generator, yet CycleGAN mainly uses ResNet as the generator; Pix2Pix essentially uses Vanilla GAN to calculate loss, yet CycleGAN adopts Least Square Loss based on LSGANs; Pix2pix employs Batch Normalization (BN) for normalization, yet CycleGAN utilizes Instance Normalization (IN) for normalization process [10]. However, they do have similarity as well: they both adopt PatchGAN as their discriminators, even though the patch size might be a bit different. From these comparison results, question (1) ("since the CycleGAN does not need the data to be paired, does that mean CycleGAN is more useful than Pix2Pix?") can be easily answered: Pix2Pix comes before CycleGAN in the image-to-image translation field. In this case, CycleGAN is more advanced and up-to-date than Pix2Pix, which means Pix2Pix approximately have more defects than CycleGAN. In the early time translating image to image, Pix2Pix could meet basic needs and provide researchers a wonderful model to immediately convert between images. In real life, however, finding images that appear in pairs in two domains is quite difficult. Therefore, CycleGAN was born. In the same research field, what is born behind is often to make up for the downsides of what is in front [11]. On this condition, CycleGAN is more widely used than Pix2Pix since it only needs the data of two domains and does not need them to have a strict correlation. Therefore, CycleGAN is more convenient than Pix2Pix in most cases, but not more useful since Pix2Pix is still primarily used in specific cases.

From those comparison, distinguishing Pix2Pix and CycleGAN is relatively easy, which is what our paper mostly wants to do. In recent research fields, researchers and authors often choose models and algorithms they are used to. Considering this, this paper hopes to give a sharp contrast between these two popular GAN models and benefits for novices hesitating to choose GANs for their research topics.

After contrasting, this paper shows the advantages and limitations of Pix2Pix and CycleGAN. Even though the CycleGAN is made on the basis of conditional GAN (cGAN) and Pix2Pix, it still has several shortcomings: the image is not encoded before conversion since CycleGAN is a direct translation method. In this case, the content before and after the generator is generated would not be too different. For this sake, researchers nowadays usually use CycleGAN for style conversion and so on, instead of directly generating images.

Every GAN has its pros and cons (advantages and limitations) in different ways. Considering this, the shortcomings of this paper is the topics (the two GAN models) may not be representative enough since they are both mainly used for image translation process. Because the types of GANs are various and there are many other interesting applications for each GAN, the future improvement for this paper would be: choose GANs from different applications and make comparison in different groups.

References

[1] "Pix2pix: Image-to-Image Translation with a Conditional Gan: Tensorflow Core." TensorFlow, https:// www. tensorflow. org/tutorials/generative/pix2pix.

[2] Reisinger, Don. "What Is pix2pix, and How Do You Use It?" Tom's Guide, Tom's Guide, 16 June 2017, https://www.tomsguide.com/us/pix2pix-faq,news-25334.html.

[3] Barla, Nilesh. "Pix2pix: Key Model Architecture Decisions." Neptune.ai, 22 July 2022.

[4] "NET: Convolutional Networks for Biomedical Image Segmentation." U, https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/.

[5] Brownlee, Jason. "How to Develop a pix2pix Gan for Image-to-Image Translation." Machine Learning Mastery, 18 Jan. 2021.

[6] Wei, Jerry. "Cyclegan: How Machine Learning Learns Unpaired Image-to-Image Translation." Medium, Towards Data Science, 28 June 2019.

[7] Brownlee, Jason. “How to Develop a Cyclegan for Image-to-Image Translation with Keras.” Machine Learning Mastery, 1 Sept. 2020.

[8] Mittal, Aditi. “Introduction to U-Net and RES-Net for Image Segmentation.” Medium, Medium, 26 Aug. 2021.

[9] Chablani, Manish. “Cyclegans and pix2pix.” Medium, Towards Data Science, 28 Apr. 2019.

[10] Karimi, Akbar. “Instance vs Batch Normalization.” Baeldung on Computer Science, 13 Oct, 2021.

[11] “Latest Research Progress of GaN and Technology to Improve Its Performance.” ATYUN Artificial Intelligence Media Platform, http://www.atyun.com/36792.html.