Factors Affecting Accuracy in Image Translation based on Generative Adversarial Network

Fumiya Yamashita, Ryohei Orihara, Yuichi Sei, Yasuyuki Tahara and Akihiko Ohsuga
Graduate School of Informatics and Engineering, University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, Japan

Keywords: Deep Learning, Domain Transfer, Generative Adversarial Network, Unsupervised Learning.

Abstract: With the development of deep learning, image translation has made it possible to output more realistic and highly accurate images. Especially, with the advent of Generative Adversarial Network (GAN), it became possible to perform general purpose learning in various image translation tasks such as “drawings to paintings”, “male to female” and “day to night”. In recent works, several models have been proposed that can do unsupervised learning which does not require an explicit pair of source domain image and target domain image, which is conventionally required for image translation. Two models called “CycleGAN” and “DiscoGAN” have appeared as state-of-the-art models in unsupervised learning-based image translation and succeeded in creating more realistic and highly accurate images. These models share the same network architecture, although there are differences in detailed parameter settings and learning algorithms. (in this paper we will collectively refer to them as “learning techniques”) Both models can do similar translation tasks, but it turned out that there is a large difference in translation accuracy between particular image domains. In this study, we analyzed differences in learning techniques of these models and investigated which learning techniques affect translation accuracy. As a result, it was found that the difference in the size of the feature map, which is the input for the image creation, affects the accuracy.

1 INTRODUCTION

Humans can easily perform analogies between two different domains in photographs, images, and the like. For example, you can easily imagine what scenery the landscape you saw during the day will be in the evening. As described above, in this paper, we call “image translation” to convert to another form and style while holding semantic information of the image before conversion between two image domains[1].

A Study which becomes the root of image translation was done from around 2001(Hertzmann et al., 2001). Especially since the approach using Convolutional Neural Networks in Deep Learning has appeared, we succeeded in creating more accurate images(Gatys et al., 2016; Johnson et al., 2016; Yoo et al., 2016). In recent years, approaches using Generative Adversarial Network (GAN) have appeared, and more advanced image translation has become possible. “pix2pix”(Isola et al., 2016) allows general-purpose learning in arbitrary pairs, such as aerial photographs to maps, monochrome images to color images, by using any two image domain data sets explicitly paired together. Therefore, more realistic and highly accurate images can be created.

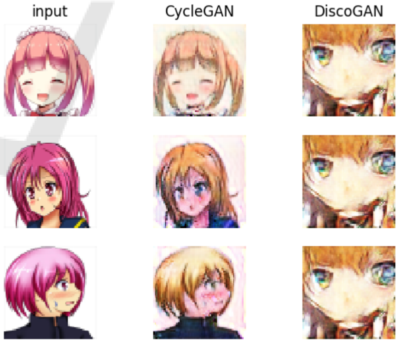

Figure 1: Result of pink hair to blond hair(64×64).

By contrast, there is also a GAN approach that allows image translation with unsupervised learning that does not require explicit pairs of images (Taigman et al., 2016; Kim et al., 2017; Zhu et al., 2017). “DiscoGAN”(Kim et al., 2017) has appeared as a state-of-the-art unsupervised learning model capable of performing highly accurate image translation. It

[1] Sometimes called “Domain Transfer”.

Yamashita, F., Orihara, R., Sei, Y., Tahara, Y. and Ohsuga, A.
Factors Affecting Accuracy in Image Translation based on Generative Adversarial Network.
DOI: 10.5220/0006591204460453
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 446-453
ISBN: 978-989-758-275-2

does not require an explicit pair of images and you only have to prepare arbitrary image domains. “CycleGAN”(Zhu et al., 2017) does not require explicit pairs as well, and translation can be done with high accuracy between various image domains. Regarding the above two models, except for differences in detailed learning techniques, the networks share the same architecture. Both can perform high-precision translation among arbitrary image domains, however it was found that there is a large difference in translation accuracy among certain image domains.

Therefore, in this study, we investigate factors affecting translation accuracy by using the two models and consider ways to improve a model for a particular domain.

2 PRELIMINARY EXPERIMENT

Figure 1 shows the output results of two models in the task of translating from the source domain (pink hair) to the target domain (blond hair), using illustration for both domains. As can be seen from the results, CycleGAN is translated into blond hair while maintaining semantic information of the input image (in this case, the shape of the illustration). On the other hand, in the case of DiscoGAN, images with completely different shapes from the input image were output. Furthermore, the same image is output in all three results. This phenomenon is called “mode collapse”, which means that it generates only images similar to a particular image for any input, which is a serious problem in the GAN(Goodfellow, 2016).

In this way, when illustration was targeted for image domain translation, it was found that there was a difference in the results of DiscoGAN and CycleGAN. Despite the same network architecture, why did the result change so much? We will conduct experiments and identify the cause in Section 5.

3 RELATED WORK

3.1 Generative Adversarial Network(GAN)

Generative Adversarial Network (GAN)(Goodfellow et al., 2014) is a generative model of unsupervised learning proposed by Goodfellow et al. Particularly in image generation, GAN can generate a realistic and highly accurate image.

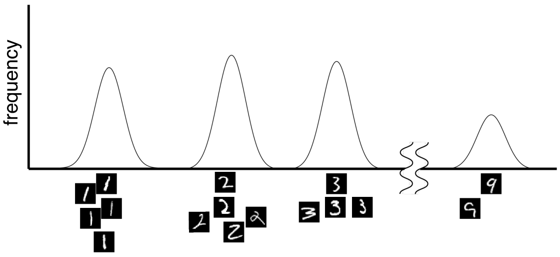

Figure 2: Two-dimensional graphed MNIST dataset.

There are two main networks in the GAN, and these are configured as “Adversarial Network” which optimize in conflict.

One is a generator G, and the goal of this network is to learn the distribution $p_x(x)$ for the training data x. The generator randomly samples the vector z from the prior distribution $p_z(z)$ such as uniform distribution and Gaussian distribution. Then, the generator outputs $G(z)$ and maps it to the generation distribution $p_g(G(z))$.

The other is a discriminator D, which is a function aimed at discriminating whether the input data is training data (real) or data generated by a generator (fake).

Given the fact that GAN is “Adversarial Network”, the goal of the two networks can be paraphrased as that the generator optimizes so as to generate data which makes the discriminator mistakenly recognize what it has generated as “real”. Then, the discriminator optimizes to be able to distinguish “real” and “fake” clearly.

Therefore, the objective function can be formulated as the following minimax optimization.

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ + \mathbb{E}_{\boldsymbol{z}\sim p_{z}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z}))]$$

For the input data, the discriminator derives the probability that it is training data (real). From Equation 1, the discriminator is aimed at maximization. Therefore, $D(\boldsymbol{x})$ becomes large and $(1-D(G(\boldsymbol{z})))$ also becomes large when the discriminator can be well discriminated. On the other hand, if the generator can successfully fool the discriminator, $D(G(\boldsymbol{z}))$ gets bigger so $(1-D(G(\boldsymbol{z})))$ gets smaller.

GAN can be useful in various aspects such as generating an image matching the context of the input text, repairing a missing part of the image, and predicting the scene a few seconds after an input image. In addition, although the original GAN(Goodfellow et al., 2014) is made up of perceptrons, a model called Deep Convolutional Generative Adversarial Networks(DCGAN)(Radford et al., 2015) that adopts CNN for the GAN network has been proposed to further enhance learning efficiency and generation accuracy of GAN (Figure 3).

We can assume various categories over each training data. For example, in the case of MNIST (handwritten digits) image data set (LeCun and Cortes, 2010), the digits correspond to the category. When a set of training data is represented by a two-dimensional graph as shown in Figure 2, it is assumed to be composed of mixed distributions formed by "peaks" of multiple distributions where each category concentrates. In the previous study, the range of the horizontal axis corresponding to each peak is defined as "mode"(Goodfellow, 2016). The generator learns so that the distribution $p_g(G(z))$ approaches the shape of the distribution $p_x(x)$ for each mode. If the generator relies on a mode that can reliably deceive the discriminator, the "mode collapse" described in section 2 will occur. In order to the avoid this, it should generate data in each mode as $p_x(x)$. In addition, because we can assume various categories with the complex structure on a real world dataset, it is much more complicated than MNIST. Mode collapse is a major issue in GAN, and recent studies have proposed several methods to solve the problem.

3.2 Image Translation

The study which becomes the roots of the image translation began when "Image Analogy"(Hertzmann et al., 2001) appeared. This is a nonparametric method, which learns the filter that is applied to the converted image from the pair of images before and after conversion and applies the same filter to any other image. In recent years, studies based on CNN approach have been actively conducted(Gatys et al., 2016).

3.3 Image Translation using GAN

3.3.1 Supervised Learning

In this paper, "supervised learning" refers to learning using training data explicitly paired in two image domains.

Yoo et al. proposed a GAN-based model that can generate an image of clothes worn by a person(target domain) from a model image wearing the clothes(source domain) using paired training data. "pix2pix"(Isola et al., 2016) can perform general translation learning between various image domains if the target image can be specified for any source image as paired training data.

3.3.2 Unsupervised Learning

In contrast, several studies have been proposed that take approaches in unsupervised learning. "DiscoGAN"(Kim et al., 2017) does not require paired training data. They proposed a model that allows image translation between two image domains only by defining them. CycleGAN (Zhu et al., 2017) proposed simultaneously can do the same thing.

DiscoGAN and CycleGAN share the same architecture although they have differences in detailed learning technics. In this study, in order to investigate the two models, we will describe the details of the network algorithm in Section 4.

4 ANALYSIS OF DIFFERENCE BETWEEN *DISCOGAN* AND *CYCLEGAN*

4.1 Algorithm

As described in Section 1 and 3, the network architecture of DiscoGAN and CycleGAN are identical (Figure 4). Therefore, we will explain the details of the algorithm using DiscoGAN. Let us call two arbitrary domains domain A and domain B respectively.

First, let us define the generator G_{AB} that takes an image x_A of domain A and convert it to one of domain B. In image translation, it is important to find a meaningful relationship (semantic information) between both domains and to convert an element of the source domain to the target domain while keeping the relationship. In order to find the relationship, DiscoGAN constrains the relation of each mode between domains to bijection. Therefore, it is necessary to define G_{BA}, the inverse translation with which the mapping from domain B to domain A can be performed. G_{BA} calculates its output $x_{ABA} = G_{BA}(x_{AB})$ using the image x_{AB} generated by G_{AB}. Finally, we derive the loss function L_{CONST_A} of the difference between x_A and x_{ABA} using arbitrary distance function (mean square error, cosine distance, etc.). As a result, we can identify the corresponding element in domain B given an element in domain A. Next, we need to build the discriminator D_B of domain B to check the appropriateness of the translation. D_B is supposed to discriminate whether an image is a training example x_B taken from the domain B (real) or a generated image x_{AB} (fake) using G_{AB} and a training example x_A drawn from the domain A. In discrimination, the D_B learns the feature of the domain B and increases discrimination ability. Therefore, G_{AB} learns to generate x_{AB} which is indis-

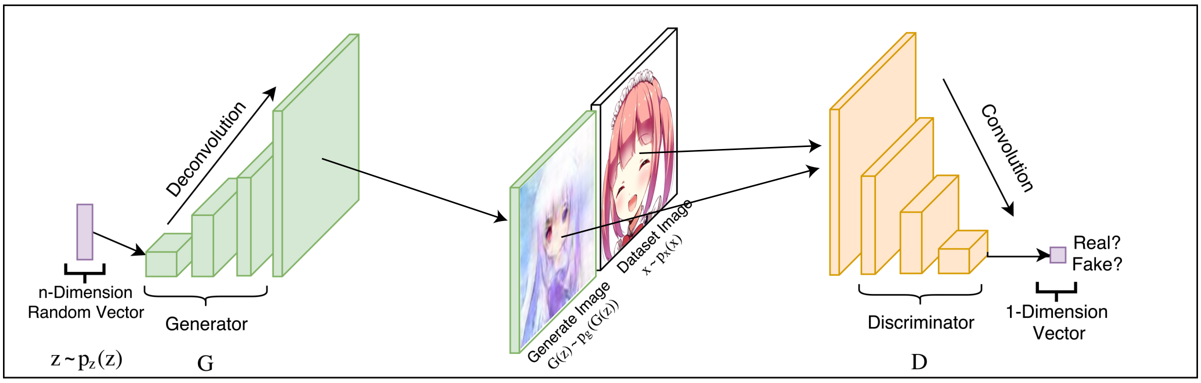

Figure 3: DCGAN network(when handling images).

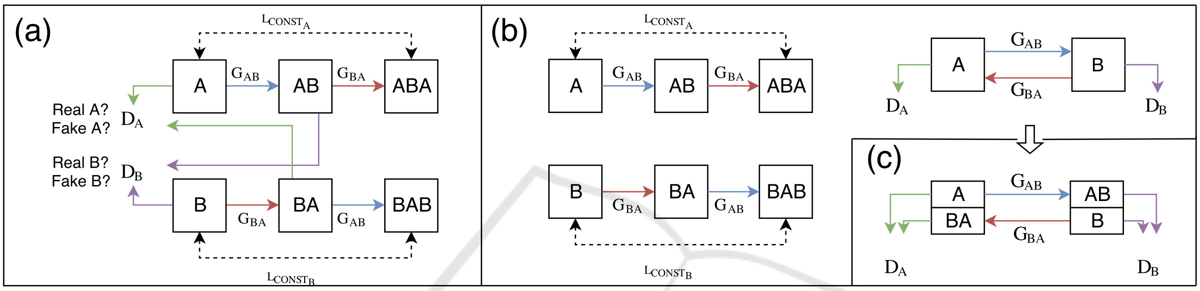

Figure 4: DiscoGAN(a) and CycleGAN(b) network(Kim et al., 2017)(Zhu et al., 2017). The part of the discriminator seems to be a different structure. However, interpreting (b) as (c), we can understand that the two networks share the same architecture.

tinguishable from the real image x_B. Therefore, the objective functions of generators G_{AB} and discriminator D_B are as follows.

$$\begin{aligned} L_{G_{AB}} &= -\mathbb{E}_{\boldsymbol{x}_A \sim p_A}[\log D_B(G_{AB}(\boldsymbol{x}_A))] \\ &+ L_{CONST_A} \end{aligned} \tag{1}$$

$$\begin{aligned} L_{D_B} &= -\mathbb{E}_{\boldsymbol{x}_B \sim p_B}[\log D_B(\boldsymbol{x}_B)] \\ &- \mathbb{E}_{\boldsymbol{x}_A \sim p_A}[\log(1 - D_B(G_{AB}(\boldsymbol{x}_A)))] \end{aligned} \tag{2}$$

Regarding the formulation, it is also necessary to impose constraints for translation from domain B to domain A in addition to the above formula. The reason is that this formulation alone does not satisfy the bijection condition. Figure 5(b) shows an example in the case of performing image translation with the original GAN. This suggests that multiple modes of Domain A may map to a single mode of Domain B if there are no constraints. This is a phenomenon called “mode collapse” mentioned in Section 2. It is a phenomenon that the generator depends on a “mode” by which discriminator easy to be fooled, and any inputs yield in the output similar to a particular image. Figure 5(c) is the result of posing L_{CONST_A}. As a result, we can identify the corresponding element in domain B given an element in domain A. However, we have not solved the mapping from multiple modes to single mode. That is, the mode collapse has not been solved. Therefore, formulation on translation from domain B to domain A must be considered. The following is the final objective function.

$$L_G = L_{G_{AB}} + L_{G_{BA}} \tag{3}$$

$$L_D = L_{D_A} + L_{D_B} \tag{4}$$

The parameters to be learned are G_{AB}, G_{BA}, D_A, D_B. When these conditions are satisfied, the mapping must be a bijection, and it becomes possible to have correspondence between the domains.

From Figure 4, CycleGAN also has the same formula as DiscoGAN. Only the name of the reconstruction loss has changed[2].

4.2 Elements of Models

In the following, we will describe the five learning techniques which stand out as clear differences between the two models from our analysis.

4.2.1 LSGAN

LSGAN (Mao et al., 2016) is a method for improving learning stability and generation accuracy for GAN. Regarding the objective function, the log likelihood

[2] Zhu et al. named this a cycle-consistency loss.

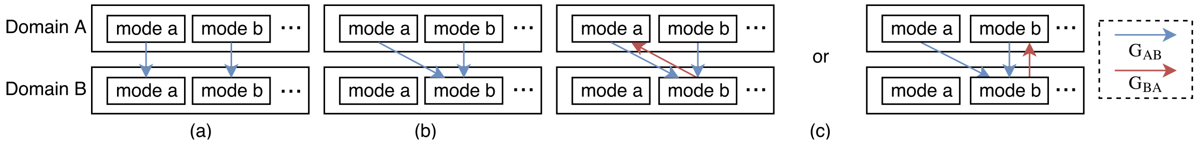

Figure 5: Mode-to-mode mappings of our models. (a): ideal result, (b): a possible result when using original GAN, (c): possible results when using L_{CONST_A}(Kim et al., 2017).

is originally used, whereas the method of Mao et al. uses the least squared error.

4.2.2 PatchGAN

PatchGAN (Isola et al., 2016) is a technique used for improving learning efficiency and stabilizing image translation by supervised[3] learning. With this technique, the discriminator is not given the whole image but $N \times N$ patches to judge it as genuine or fake. Since the normal discriminator is given the whole area of the image, the number of the learning parameter becomes large depending on the image size. Since PatchGAN is given an image locally, the number of the learning parameters can be kept low, and the discriminator is strengthened by increasing the learning efficiency.

4.2.3 Buffer

SimGAN (Shrivastava et al., 2016) is a model that can convert a synthetic image created for training data into a more realistic image. In order to cope with lack of training data, they used a simulator to create synthetic images. However, there was a risk that the synthetic images were not realistic enough, resulting unstable learning. SimGAN was proposed to mitigate its risk. Buffer is used as a method of improving accuracy in this model. When giving an image to the discriminator, it uses not only a newly generated image but also an image generated in past (buffer) with a certain probability.

4.2.4 Definition of L_{CONST}

In DiscoGAN, L_{CONST} is shown as an arbitrary distance function. For its implementation, mean square error was used. On the other hand, CycleGAN uses the L1 norm.

4.2.5 ResNet

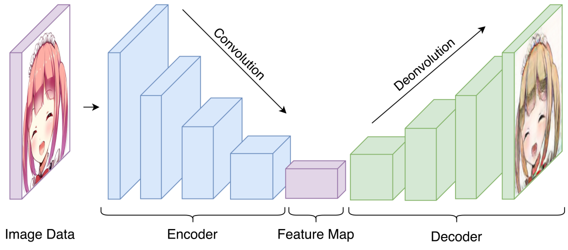

Figure 6: Generator used in this study.

In image recognition, it is said that the deeper the layer of CNN becomes, the higher the recognition accuracy will be. However, there was a problem that recognition error increases if the layer is made too deep. Deep Residual Learning (ResNet) (He et al., 2016) is a method for constructing a deep network of layers. Learning the residual between the outputs and inputs of that layer rather than learning the optimal output at a layer eliminates the need to use unnecessary weights. By doing this, it becomes possible to extend the limit of the depth of the layer and to stabilize learning.

5 EXPERIMENTS

5.1 Experimental Setups

In this study, we examine the influence of the presence or absence of the five learning techniques on the generation accuracy in image translation. Using DiscoGAN as a baseline model, experiments are conducted with a total of 32 cases that defined by presence or absence of learning techniques. For the data set, we used 5000 pink hair portrait illustration images in the source domain and 5000 blond ones in the target domain. The image size was 64×64 resolution.

Regarding the architecture of the generator and discriminator, it is composed of multiple convolutional layers based on DCGAN(Radford et al., 2015). Originally, the generator adopts a random vector for input. In image translation, an encoder composed of a convolution layer is built in order to handle an image as input. Next, the vector output from the encoder is taken as input and handled as a feature on the input image. Then, by entering the vector to a decoder composed of deconvolution, an image is generated. The configurations of encoder and decoder are often collectively called a generator (Figure 6). On the other hand, the discriminator is composed of a convolution layer in the same way as the original GAN.

[3] Learning by explicit pairing of training data

Table 1: Architecture of DiscoGAN and CycleGAN.

DiscoGAN							
Net	Layer	C	F	S	P	BN	A
G	c1	64	4	2	1		leaky
	c2	128	4	2	1		leaky
	c3	256	4	2	1		leaky
	c4	512	4	2	1		leaky
	dc5	256	4	2	1		ReLU
	dc6	128	4	2	1		ReLU
	dc7	64	4	2	1		ReLU
	dc8	3	4	2	1		tanh
D	c1	512	4	2	1		leaky
	c2	256	4	2	1		leaky
	c3	128	4	2	1		leaky
	c4	64	4	2	1		leaky
	c5	1	4	1	0		identity

c	convolution(down convolution)
dc	deconvolution(up convolution)
leaky	leaky ReLU
identity	identity mapping
RN	ResNet
unp	unpooling
C...Channel, F...Filter Size, S...Stride, P...Padding, BN... Batch Normalization, A...Activation Function	

CycleGAN							
Net	Layer	C	F	S	P	BN	A
G	c1	32	7	1	3		ReLU
	c2	64	4	2	1		ReLU
	c3	128	4	2	1		ReLU
	RN4	128	3	1	1		ReLU
	RN5	128	3	1	1		ReLU
	RN6	128	3	1	1		ReLU
	RN7	128	3	1	1		ReLU
	RN8	128	3	1	1		ReLU
	RN9	128	3	1	1		ReLU
	RN10	128	3	1	1		ReLU
	RN11	128	3	1	1		ReLU
	RN12	128	3	1	1		ReLU
	unp13	64	2	2	0		identity
	c13	64	3	1	1		ReLU
	unp14	32	2	2	0		identity
	c14	32	3	1	1		ReLU
	c15	3	7	1	3		tanh
D	c1	64	4	2	1		leaky
	c2	128	4	2	1		leaky
	c3	256	4	2	1		leaky
	c4	512	4	2	1		leaky
	c5	1	3	1	1		identity

Table 2: Shape evaluation result. For each list of digits, set the method to be applied to "1" and those not apply "0". From the left "PatchGAN, LSGAN, L_{CONST_A}, ResNet, Buffer". We will use this notation hereafter.

Failure				Success		
00000	00101	01100	**10101**	**01010**	10011	11110
00001	00110	**01101**	11000	01011	10110	11111
00010	**00111**	10000	**11001**	01110	10111	
00011	01000	10001	**11100**	01111	11010	
00100	01001	10100	**11101**	**10010**	11011	

Table 1 shows the details of each CNN structure. For PatchGAN and ResNet, use DiscoGAN network when not used, and use CycleGAN network when used.

5.2 Experimental Result

Several representative examples out of 32 experimental results are shown in Figure 7.

Models are compared with each other by checking whether x_{ABA} holds the form of the original input x_A after application of G_{AB} and G_{BA}. We call the method "shape evaluation". We have visually evaluated whether the shape of the original input x_A is maintained at x_{ABA} from the Figure 7. It is assumed to be "success" when the face outline, eyes, mouth were reconstructed along the original input. On the other hand, it is evaluated as "failure" when the mode collapse occurs, when the shape of the face is malformed or when a part of the part is missing. The results are shown in Table 2.

Boldface items show failed patterns despite using three or more learning techniques and successful patterns despite using two or less learning techniques. From these results, PatchGAN, LSGAN, ResNet can be cited as methods having an important role in shape reconstruction.

The condition for success can be summarized as the following. In the case where PatchGAN is not used, shape reconstruction will succeed if both LSGAN and ResNet are used. In the case where PatchGAN is used, shape reconstruction is successful if ResNet is used. All other cases failed. From these results, it turns out that the most important learning technique among the five techniques is ResNet.

Table 3: Details of failure pattern.

Mode collapse			No mode collapse	
00000(0)	00110(1)	01101(1)	00011(1)	11100(2)
00001(0)	00111(1)	10000(1)	10001(1)	11101(2)
00010(1)	01000(1)	10100(1)	10101(1)	
00100(0)	01001(1)		11000(2)	
00101(0)	01100(1)		11001(2)	

From the above results, it seems that Buffer and L_{CONST} have no particular effect. As for L_{CONST}, it has no particular impact in this experiment, just as Kim et al. stated that an arbitrary distance function could be used. However, the effect of Buffer is unexplainable from the above result. We have found an evidence that Buffer could contribute to avoid the mode collapse, after classifying the failure patterns based on the presence or absence of the mode collapse. In this experiment, it is judged that mode collapse occurs when an image similar to a particular image is generated for two different input images. The result is shown in Table 3.

In this table, the numbers in parentheses indicate the number of techniques used in a model among three techniques (PatchGAN, LSGAN, ResNet) that are considered important. From the results, it can be seen that mode collapse occurs if only one main technique is used or not used at all. On the other hand, when both PatchGAN and LSGAN are used, mode collapse has not occurred. Buffer + ResNet pattern and Buffer + PatchGAN pattern also did not cause mode collapse, and the influence of Buffer on major techniques was observed. However, as for Buffer + ResNet, since the mode collapse pattern (00111) also exists, it was found from this result that the effect of mode collapse prevention by combination of PatchGAN and Buffer is significant.

6 DISCUSSION

Experimental results show that PatchGAN and LSGAN and ResNet are three important methods. Among them, ResNet has an important role in both shape reconstruction, and it is considered to be the most important method among the five.

Although GAN originally uses random noise of about 100 dimensions for input, this time we use images for input. As seen in Figure 6, the input image is compressed by the convolution layer (encoder), and it is used as the input of the generator in the form of a feature map. In this experiment, the training examples are 64×64 resolution images. When we use the

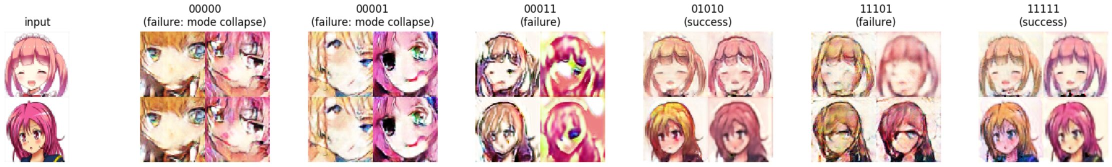

Figure 7: Results. Top left: x_{AB}. Top right: x_{ABA}. Bottom left: x_{AB}(another input). Bottom right: x_{ABA}(another input).

DiscoGAN network, we will downsample them to a very small feature map of 4 × 4. Therefore, the dimension of the input space of the generator becomes much lower than the training data distribution. That makes mode collapse more likely and leads to generation failure. On the other hand, since CycleGAN with ResNet employs a 16x16 feature map, mode collapse is unlikely and the shape becomes stable.

From the above, we make one hypothesis. That is, by increasing the size of the feature map output by the encoder, shape reconstruction succsessful rate may be improved. In order to verify these, two additional experiments are carried out. First, by using images of 256 × 256 resolution as the input of DiscoGAN (00000), we examine the influence of the image size on the quality of the generated image (Figure 8(c)). Next, by using images of 64 × 64 resolution as the input of DiscoGAN (00000), we examine the influence of the number of convolution layers on the quality of the generated image (Figure 8(d)). The encoder parts of both DiscoGAN and CycleGAN are composed only of convolution layers that perform downsampling, and the size of the feature map is determined by the image size and the number of layers. In other words, the two additional experiments are common in that the feature map is extended rather than the ordinary DiscoGAN. When the result of image translation by additional experiment is better than that of original DiscoGAN, it is understood that the difference in accuracy between DiscoGAN and CycleGAN is attributed to the size of the feature map used as input to the decoder. The results are shown in Figure 9.

From the results, with 256 × 256 resolution images, the results were as good as CycleGAN. Although the quality is poor as the result of reducing the layer, the shape is stable without mode collapse. This result is a category of “success” in this experiment. As a result, we can attribute the improvement to the size of the feature map as stated in the hypothesis.

Also, from the Table 2, it turns out that the result with ResNet alone is a failure (00010). This is because the filter size for convolution in Conv 1 of Table 1(CycleGAN) is large, which caused the loss of information. In this regard, however, it will be successful if only PatchGAN or LSGAN is added to ResNet. Therefore it can be seen once again that these two are largely attributed to image quality improvement.

As for the mode collapse, Table 3 shows that there is an influence by buffer. In particular, the combination of PatchGAN and Buffer improves mode collapse despite the small feature map, which shows that it is an important technique.

The mode collapse is caused by imperfect nature of the discriminator. Learning of discriminators can be tricky. Suppose there are two modes, one in which the discriminator can accurately discriminate real or fake, and another in which the discriminator performs poorly. The generator learns to rely on the second mode to reliably fool the discriminator and thereafter the distribution of generated data only includes the second mode.

Both PatchGAN and Buffer are approaches to discriminators and are used as important learning techniques to improve discriminator's performance. However, as for Buffer, when two or more major techniques were used, the quality was improved regardless of its presence. PatchGAN is more important for quality improvement.

7 CONCLUSION

In this study, we examined learning techniques that are responsible to the difference in the image generation accuracy in a particular domain for the two image translation models. Among them, ResNet was used as the most influential one. It turned out that the feature map size is an important parameter. As for the mode collapse, it was found that it can be improved by using two or more major techniques or by combining PatchGAN + Buffer.

CycleGAN is a model that is good at texture/style translation. However, it performed poorly in tasks that change shape and were cited as future tasks. Meanwhile, in DiscoGAN, there is a successful example of a task to change shape, leaving very precise results. However, the images belonging to the domain are relatively similar to each other, and DiscoGAN has not succeeded in the case where there are various images in the domain. For example, this model has succeeded tasks that change shapes, such as facial expressions and gender. Both of these use human faces and

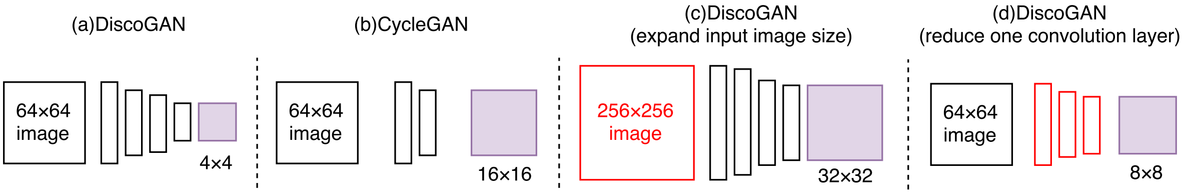

Figure 8: Feature map size output from encoder. In the additional experiments (c) and (d), the feature map size is larger than that of the ordinary DiscoGAN(a).

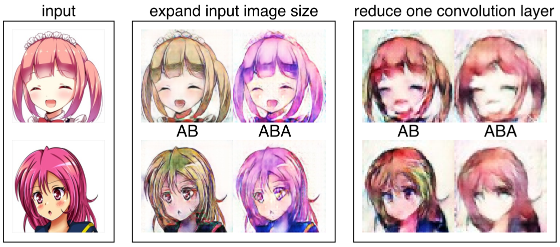

Figure 9: Additional experiment results.

are relatively similar. As can be seen from these, we consider that five learning techniques need to be adjusted not only by the generation accuracy but also by the image domain to be handled. In this study we conducted experimental verification focused on shape evaluation, but we did not consider domain translation accuracy (in this paper, it is an indicator whether it is translated into blond hair or not). The indicator is difficult for this domain translation and we will treat it as a future work.

Based on the result of the domain handled in this study, we will try targeting illustrations in the future and try to realize socially useful agents for creators who deal with them.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP16K12411, JP17H04705.

REFERENCES

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 2414–2423.

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1612.07828*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems(NIPS)*, pages 2672–2680.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 770–778.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H. (2001). Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques(SIGGRAPH)*, pages 327–340. ACM.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision(ECCV)*, pages 694–711. Springer.

Kim, T., Cha, M., Kim, H., Lee, J., and Kim, J. (2017). Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Smolley, S. P. (2016). Least squares generative adversarial networks. *arXiv preprint arXiv:1611.04076*.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2016). Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*.

Taigman, Y., Polyak, A., and Wolf, L. (2016). Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

Yoo, D., Kim, N., Park, S., Paek, A. S., and Kweon, I. S. (2016). Pixel-level domain transfer. In *European Conference on Computer Vision(ECCV)*, pages 517–532. Springer.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*.