2023 International Symposium on Nonlinear Theory and Its Applications
NOLTA2023, September 26-29, 2023, Catania and Online

Multi-domain translation from few data by CycleGAN applying data augmentation

Shuhei Kanzaki†, Hidehiro Nakano‡

†Graduate School of Science and Engineering, Tokyo City University
1–28–1, Tamazutsumi, Setagaya-ku, Tokyo, 158–8557 Japan
Email: g2281419@tcu.ac.jp, hnakano@tcu.ac.jp

Abstract— In machine learning and deep learning, a huge amount of data is required for training. The image generation model GAN exists as a method to supplement the huge amount of training data. Data Augmentation is one of the methods to increase the number of data. It has been shown that the application of Data Augmentation to GANs can improve the performance of the GANs. This research proposes a method to apply Data Augmentation to CycleGAN, which uses two GANs, and the effectiveness of the method on the model is verified.

1. Introduction

Deep Learning uses multi-layered neural networks to learn the features in the data step by step. This technology has been actively studied not only in the field of data recognition such as image recognition and speech recognition, but also in the field of data generation and transformation. Recently, Generative Adversarial Networks (GANs)[1] have attracted much attention as a generative model. GAN consists of two networks, Generator and Discriminator, where Generator generates data and Discriminator determines whether the input data is generated by Generator or not. The data generated by GAN can be used to supplement the data needed in the machine learning field. One of derivative models of GANs is CycleGAN[2], which enables the transformation between different image groups. CycleGAN consists of two GAN networks, each of which has a Generator and a Discriminator. Data Augmentation (DA)[3] is one of the methods to compensate for the lack of data. DA is a technique for processing an image without destroying the original meaning of the image. It has been shown that applying this DA to GANs improves the performance of the GANs. In this research, we apply DA, which has been effective for GAN, to CycleGAN. By applying DA to training data for CycleGAN, it is possible to train models with a small amount of training data. Numerical experiments will be conducted to evaluate the effectiveness of our method.

ORCID iDs Shuhei Kanzaki: 0009-0002-2652-7639, Hidehiro Nakano: 0000-0003-0302-123X

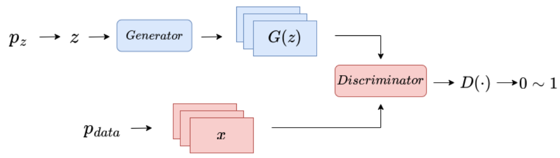

Figure 1: Overview chart of GAN.

2. GAN

Generative Adversarial Networks (GAN)[1] is a model for estimating the probability distribution p_{data} of arbitrary data. An overview diagram of GAN is shown in Fig.1.

GAN is composed of two networks, Generator and Discriminator. Generator takes as input a noise vector z sampled from a probability distribution p_z , and outputs data. This output is denoted as $G(z)$. The Discriminator takes as input the data x sampled from the probability distribution of the training data and the output $G(z)$ from the Generator and the output represents the probability that the input data belongs to p_{data}. This output is denoted as D(•) and the output range is [0,1]. The purpose of the Generator is to generate data such that the Discriminator identifies the output from the Generator, $G(z)$, as data sampled from p_{data}. The purpose of the Discriminator is to correctly identify whether the input is data sampled from p_{data} or data generated by the Generator. The evaluation function of GAN is shown in Eq. (1).

$$\begin{aligned} V(G,D) &= \min_G \max_D (G,D) \\ &= \mathbb{E}_{x\sim p_{data}}[\log(D(\boldsymbol{x}))] + \mathbb{E}_{z\sim p_z}[1-\log(D(G(\boldsymbol{z})))] \end{aligned} \tag{1}$$

The Generator learns to minimize the evaluation function, and the Discriminator learns to maximize the evaluation function. Where $\mathbb{E}$ denotes the expected value for the random variable.

3. CycleGAN

CycleGAN[2] is a model that does not use paired data but learns the relationship between two domains. An overview diagram of CycleGAN is shown in Fig.2.

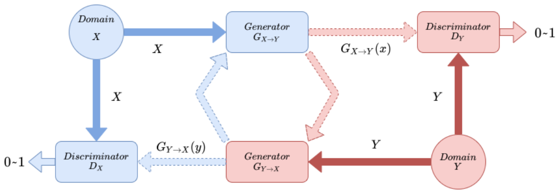

Figure 2: Overview chart of CycleGAN.

CycleGAN consists of two GANs. Two Generators are used to convert between domains, and two Discriminators are used to discriminate between domains. Let the two domains be X and Y, and the Generator $G_{X\to Y}$ (respectively, $G_{Y\to X}$) transforms data from the Domain X to Y (respectively, from X to Y). The Discriminator D_X (respectively, D_Y) identifies whether the data are sampled from the Domain X (respectively, Y) or transformed by the Generator $G_{Y\to X}$ (respectively, $G_{X\to Y}$). The evaluation function of CycleGAN is shown in Eq.(2).

$$
\begin{aligned}
&\min_{G_{XtoY},G_{YtoX}} \max_{D_X,D_Y} (G_{XtoY}, G_{YtoX}, D_X, D_Y)\\
&\quad = L_{adv}(G_{XtoY}, D_Y, X, Y)\\
&\quad + L_{adv}(G_{YtoX}, D_X, X, Y)\\
&\quad + \lambda L_{cyc}(G_{XtoY}, G_{YtoX}, X, Y)
\end{aligned} \tag{2}
$$

Adversarial Loss L_{adv} is an evaluation function used in GAN. It is used to express the distance between probability distributions. Since CycleGAN consists of two GANs, two types of L_{adv} are used, described by

$$
\begin{aligned}
&L_{adv}(G_{XtoY}, D_Y, X, Y)\\
&= \mathbb{E}_{y\sim p_{data}(y)}[\log(D_Y(\boldsymbol{y}))]\\
&+ \mathbb{E}_{x\sim p_data(x)}[1 - \log(D_Y(G_{XtoY}(\boldsymbol{x})))]
\end{aligned} \tag{3}
$$

$$
\begin{aligned}
&L_{adv}(G_{YtoX}, D_X, X, Y)\\
&= \mathbb{E}_{x\sim p_{data}(x)}[\log(D_X(\boldsymbol{x}))]\\
&+ \mathbb{E}_{y\sim p_data(y)}[1 - \log(D_X(G_{YtoX}(\boldsymbol{y})))]
\end{aligned} \tag{4}
$$

One of the major features of CycleGAN is Cycle Consistency Loss L_{cyc}. Since the generators are inverse functions to each other, the original data is recovered by passing the input data through two Generators. That is $G_{YtoX}(G_{XtoY}(\boldsymbol{x})) \approx \boldsymbol{x}$, and $G_{XtoY}(G_{YtoX}(\boldsymbol{y})) \approx \boldsymbol{y}$. We take the L1 norm of the input and the restored output, which is called Cycle Consistency Loss. No pair data is needed to recover the input data, and the correct recovery of the input data implies that the mapping between the two domains is correct. In other words, CycleGAN can learn the relationship between domains by unsupervised learning without using pair data. The is L_{cyc} described by

$$
\begin{aligned}
&L_{cyc}(G_{XtoY}, G_{YtoX}, X, Y)\\
&= \mathbb{E}_{y\sim p_{data}(y)}[||G_{XtoY}(G_{YtoX}(\boldsymbol{y})) - \boldsymbol{y}||]\\
&+ \mathbb{E}_{x\sim p_{data}(x)}[||G_{YtoX}(G_{XtoY}(\boldsymbol{x})) - \boldsymbol{x}||]
\end{aligned} \tag{5}
$$

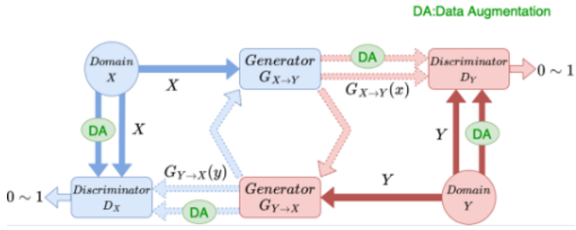

Figure 3: Overview chart of Method 1.

4. Proposed Method

We propose two methods for applying Data Augmentation to Cycle GAN.

4.1. *Apply Data Augmentation to the network (Method1)*

We propose an image generation model by applying DA to the network of Cycle GAN. An overview diagram of the proposed model is shown in Fig.3.
CycleGAN consists of two GANs: two Generators, which perform the transformation between domains, and two Discriminators, which perform the identification in each domain. Let us consider the cases where DA is applied to CycleGAN. Let the processing of DA be $T(\cdot)$. The real image is the data sampled from the Domain X or Y, and the fake image is the data generated from the Generator $G_{X\to Y}$ or $G_{Y\to X}$. $T(\boldsymbol{x})$ or $T(\boldsymbol{y})$ represents the real image data applied DA. Also, $T(G_{X\to Y}(\boldsymbol{x}))$ or $T(G_{Y\to X}(\boldsymbol{x}))$ represents the fake image data applied DA. Therefore, four applicable locations to the DA processing exist in CycleGAN, and the number of the combinations is $2^4 = 16$.

4.2. *Apply Data Augmentation to the Training data set(Method2)*

By applying DA to the training dataset, we propose a method to train models under the situation where the number of training data is limited. The extended dataset by DA is shown in Fig. 4. First, we extract n data from the original training dataset. Next, we apply m types of DA to the extracted n data. Then, $(M + 1)N$ data can be prepared as the extended dataset. Using this method, the original dataset can be extended depending on the number of types of DA applied.

In this research, DA is applied to each segmented data and the experiments are conducted to verify each effect during data expansion. DA method for segmented data is shown in Fig. 5. As shown in Fig. 5, the original data set of 1000 is divided into five datasets, and a new dataset is created by applying one type of DA to each of them.

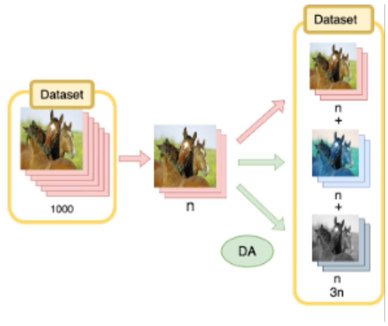

Figure 4: The extended datasets (Method 2)

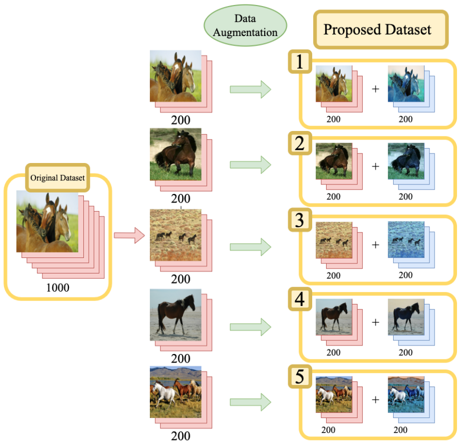

Figure 5: The separated datasets (Method 2)

5. Experiments and Evaluation

Experiments on the dataset hose2zebra are performed in order to validate the effectiveness of the proposed methods. The dataset used in the experiments consists of 2 domains, horse and zebra. For the evaluation of the generated images, we use the Frèchet Inception Distance(FID)[4]. The distance between the probability distributions of the sample images and the generated images is used to evaluate how close the probability distribution mapped by the Generator is to the true probability distribution of the sample images. FID is described by Eq.(6).

$$||m - m_w||_2^2 + Tr(C + C_w - 2(CC_w)^{\frac{1}{2}}) \qquad (6)$$

m and m_w are the mean of the feature vectors obtained from the sample and generated image groups, and C and C_w are the covariance matrix. The evaluation is better if the FID value approaches to 0.

5.1. *Method 1*

We applied grayscale's DA to four locations in the network of CycleGAN described in Method 1. The images generated by changing the ratio to apply DA are shown in

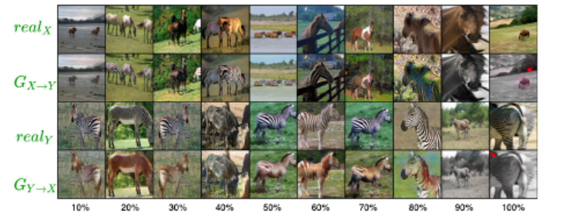

Figure 6: Generated images by grayscale for application ratio.

Fig. 6, and the FID results for the generated images are shown in Table 1.

It can be confirmed that the influence of DA is widely reflected in the generated images when the ratio of DA is gradually increased. The appropriate ratio of DA can reduce the influence of DA to the generated images. As shown in Fig. 5, transformation between each domain is performed well if the ratio of DA is among 10% and 60%. However, the generated images are strongly affected by DA if the ratio is higher than 70%. We also confirmed that domain conversion may not be performed in some cases.

Table 1: Results for application ratio

0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
16.31	**13.62**	**14.62**	**16.12**	**14.52**	17.62	26.78	33.42	39.7	120.63	126.01

Table 1 shows the FID values as the ratio to apply DA varies from 0% to 100%. Adjusting the ratio appropriately, the evaluation values of the generated images become better and the performance can be improved

5.2. *Method 2*

CycleGAN training was performed on multiple DA training datasets. Table 2 shows the numerical results evaluated by FID. In Table 2, "1" is the result for using the original dataset only. As the dataset number increases, the number of types of DA applied increases. For example, the dataset number "3" with 100 data means that 100 original data is used, and 100+100 data is added by using TranslationY and TranslationX.

Table 2: Results of sequential addition of multiple DAs

	Data Augmentation	1000	500	250	200	100
1	Original	16.30	20.77	28.89	32.42	38.13
2	TranslationY	-	**18.58**	**22.48**	**25.39**	**22.91**
3	TransltaionX	-	**15.20**	**19.52**	**22.41**	**27.49**
4	TranslationX&TranslationY	-	**14.60**	**20.47**	**22.07**	**30.16**
5	Blueness	-	**12.93**	**16.45**	**20.53**	**24.83**
6	Grayscale	-	**13.16**	**17.82**	**18.80**	**23.32**
7	Noise	-	**10.30**	**16.43**	**20.01**	**26.73**
8	Hue Change	-	**12.45**	**15.76**	**19.45**	**26.88**

In Table 2, all the results with DA shows better performance than those without DA. From these results, it was confirmed through the experiments that the application of DA to CycleGAN enables data extension in an environment with a

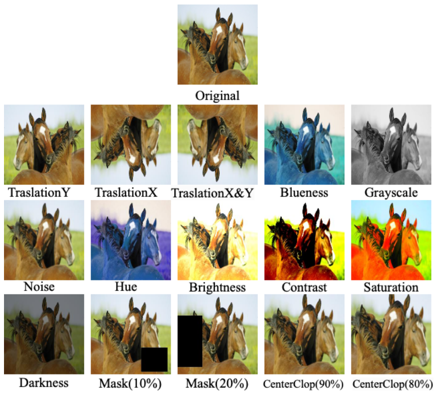

Figure 7: Applied data augmentation methods

limited number of data. However, it is found that even if the number of types of DA applied is increased cumulatively, the performance improvement reaches a ceiling, and there are cases where the performance deteriorates instead. We should verify the performance by the order and combinations to apply DA, in more detail.

Next, we show the results in which one type of DA was applied to the original data sets separated into five. The number of original data used is 200. Figure 7 shows the applied DA.
Mask(10%) and Mask(20%) are the processes to add a mask of 10% and 20% to the original data, respectively. CenterClop(90%) and CenterClop(80%) are the processes to crop 90% and 80% of the image from the center, respectively.

Table 3 shows the evaluation results applied the each separated data to one type of DA.

Table 3: Results of applying one type of DA to the separated datasets

	Data Augmentation	1	2	3	4	5
1	Original	30.59	24.96	26.05	29.23	30.52
2	TranslationY	**23.50**	25.27	**24.13**	**25.82**	**25.75**
3	TranslationX	**23.24**	**24.78**	**21.02**	**22.13**	**23.08**
4	TranslationX&TranslationY	**25.26**	25.12	**20.30**	**21.95**	**23.95**
5	Blueness	**24.48**	25.12	**23.45**	**21.76**	**22.23**
6	Grayscale	**20.93**	**22.40**	**21.29**	**19.94**	**22.88**
7	Noise	**27.66**	26.96	**25.53**	29.64	**26.30**
8	Hue Change	**25.23**	25.67	**23.39**	**23.84**	**25.22**
9	Brightness	**22.78**	**21.70**	**22.55**	**22.68**	**24.05**
10	Contrast	**21.14**	**20.91**	**21.05**	**21.05**	**21.82**
11	Sturation	**23.64**	25.41	**21.08**	**24.46**	**23.40**
12	Darkness	47.93	49.50	37.04	47.17	41.24
13	Mask(10%)	**21.40**	**23.74**	**22.76**	**22.20**	**22.22**
14	Mask(20%)	**25.45**	**24.15**	**22.49**	**22.39**	**24.18**
15	CenterClop(90%)	**24.96**	26.73	**24.61**	**23.48**	**25.39**
16	CenterClop(80%)	**24.97**	25.79	**23.08**	**24.49**	**27.51**

From Table 3, we confirmed that there is variation in each separated data. For dataset 2, the performance for the data with DA is worse than that for the original data. Not that the results tended to be better when DAs that change the color tones, such as Brightness, Contrast, and Saturation, were applied. We should verify the performance for the types of the applied DA and theire combinations, in more detail.

6. Conclusion

In this research, we proposed two methods to apply Data Augmentation, which has been effective for GAN, to CycleGAN. In Method 1, we applied DA to the network and investigated the effect of DA on the applicable locations. We also confirmed the effectiveness of learning the model by adjusting the ratio of applying DA. In Method 2, we proposed a method of adding a training dataset based on the data expansion method. By supplementing the training data with Data Augmentation, we were able to demonstrate the stability of learning on a limited number of data. Future issues to be addressed are the selection of DAs to be used based on the results of this research. In addition, it is necessary to consider the order of the DAs to be applied.

Acknowledgements

This work was supported by JSPS KAKENHI Grand-in-Aid for Scientific Research (C) Number: 22K12200.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio: ” Generative Adversarial Nets ” , In Proceedings of the 27th International Conference on Neural Information Processing Systems, Vol.2, pp.2672-2680, 2014.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros: ” Unpaired Image-to- Image Translation using Cycle-Consistent Adversarial Networks. ” , In Proceedings of the International Conference on Computer Vision, pp.2242-2251, 2017.

[3] A survey on Image DA for Deep Learning, In : Jornal of Big Data. 2019

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter: "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium", In Proceedings of the 30th International Conference on Neural Information Processing Systems, pp.6626-6637, 2017.