Generative Adversarial Networks with Data Augmentation and Multiple Penalty Areas for Image Synthesis

Li Chen
School of Computer Sciences, Universiti Sains Malaysia, Malaysia
chenli0926@student.usm.my

Huah Yong Chan
School of Computer Sciences, Universiti Sains Malaysia, Malaysia
hychan@usm.my

Abstract: *The quality of generated images is one of the significant criteria for Generative Adversarial Networks (GANs) evaluation in image synthesis research. Previous researches proposed a great many tricks to modify the model structure or loss functions. However, seldom of them consider the effect of combination of data augmentation and multiple penalty areas on image quality improvement. This research introduces a GAN architecture based on data augmentation, in order to make the model fulfill 1-Lipschitz constraints, it proposes to consider these additional data included into the penalty areas which can improve ability of discriminator and generator. With the help of these techniques, compared with previous model Deep Convolutional GAN (DCGAN) and Wasserstein GAN with gradient penalty (WGAN-GP), the model proposed in this research can get lower Frechet Inception Distance score (FID) score 2.973 and 2.941 on celebA and LSUN towers at 64×64 resolution respectively which proves that this model can produce high visual quality results.*

Keywords: *Generative model, image synthesis, data augmentation*.

Received May 20, 2022; accepted January 23, 2023
https://doi.org/10.34028/iajit/20/3/15

1. Introduction

Image synthesis is one of the important application areas of probabilistic generative models. Such models usually need to focus on the sample distribution itself. Previous researchers combined these approaches to deep neural networks and proposed many influential models. Variational Autoencoders (VAEs) [22] depend on approximate inference methods that introduce restrictions of both reconstruction error and Kullback-Leibler (KL) divergence. With the help of parameterization trick, data sampling can be achieved in training process. However, Due to the limitation of the model, the generated samples from VAE are often blurry. Autoregressive models [20] directly model the probabilistic distribution. Since they need to sample data pixel by pixel, the generation efficiency is not satisfied, and the computation cost is relatively large. Flow-based models [6, 7] are also named change of variable models, they need to add many restrictions on the generator, thus cause limitations of the ability of generator. Generally, these models can be categorized into explicit density methods model that they all need to estimate probabilistic density of the real data distribution.

Unlike the models above, Generative Adversarial Networks models (GANs) [9] are implicit density model which do not require direct estimation of the probabilistic density function and log-likelihood. The architecture of GANs normally contains two components which are generator and discriminator. The task of generator is to map the latent code to the real data space, while the discriminator can be considered as a binary classifier that distinguishes real data between generated data. The loss function is like the minmax game which is basically defined as below:

$$\min_G \max_D E_{x \in pdata}[\log D(x)] + E_{x \in pg}[\log(1 - D(x))] \quad (1)$$

In formula (1), G and D represent generator and discriminator respectively. $x \in pdata$ indicates the data from real data distribution and $x \in pg$ stands for fake data from the generator (x ($x \in pg = G(z)$. The fake data can be obtained by $G(z)$ where latent code z is normally sampled from standard Gaussian distribution or uniform distribution. During training process, generator aims to generated data which can cheat the discriminator. In other words, the object of generator is to minimize the Jensen-Shannon (JS) divergence between real data distribution and model distribution. Discriminator is used to measure the divergence between these two distributions, if the divergence between the two is large, the discriminator will give a lower score to the *pg* and a higher score to the *pdata* and vice versa. As training progresses, the generator generates data that is close to the real data distribution. When *pg* is exactly as the same as *pdata*. That is, the divergence is equal to 0 between real distribution and generated distribution, the probability for both the real data and the generated data is 0.5. Finally, the generator and discriminator reach at

Nash equilibrium.

However, JS divergence does not accurately measure the difference between two distributions. Since *pg* and *pdata* are manifolds of high-dimensional spaces, these two distributions often hardly overlap. So that, JS divergence is always equal to log2 for non-overlapping distributions. That is, the discriminator can distinguish two kinds of data very well in the beginning which leads to gradient vanishing. Moreover, it also causes the generator obtain less gradient information to update parameters during optimization process resulting in inability to generate better results.

This research proposes the GAN model with data augmentation. To be specific, it acquires extra data with noise $G(z+noise)$ from generator, and feed them to discriminator. During training, an encoder network needs to reconstruct $G(z+noise)$ to z which forces generator to learn more concrete feature from real data. Meantime, $G(z+noise)$ will be fed to discriminator along with real data $p(x)$ and generated data $G(z)$. In order to fulfill 1-lipchiz constraints, it is necessary to consider adding $p(z+noise)$ into the gradient penalty areas. That is, not only the areas between real data and fake data but also the areas form real data between data with noise should also be regarded as penalty area. Experiments have proved that these methods can improve the ability of discriminator, thereby helping generator to generate more realistic data.

The contributions of this research are as follows:

- A GAN model based on data augmentation and multiple penalty areas is proposed.
- It has been proved that adding multiple penalty areas term to loss function can improve quality of generated images based on FID score.
- The results on LSUN towers [26] and celebA [14] at 64×64 resolution can surpass previous model DCGAN and WGAN-GP.

The related work is introduced in section 2, and proposed method is shown in section 3. In section 4, the details of experiment analysis is mentioned. Finally, the conclusion of this study and future work are discussed in section 5.

2. Related Work

In order to improve the model's ability to get better generated results, some studies used data with added noise to participate in training process. Denoising autoencoder [25] introduced an idea that takes corrupting data as input to get the better reconstruction performance in autoencoders training. Under this condition, the encoder can learn more robust data features, thereby enhancing model's performance. Contractive autoencoder [23] added the Frobenius norm of the Jacobian matrix as penalty term to the loss function. This method reduces the impact of input changes on the latent code to a certain extent. So, there would be some benefit to model robust. Denoising feature matching [25] combined features distribution with denoising autoencoder to modify the loss function which get both qualitative and quantitative improvements in generated data.

Instead of adding extra data to generator, some studies adopted other distribution distance metrics for discriminator. WGAN [1] modified the loss function of discriminator which took Wasserstein distance to replace JS divergence. In practice, it adds weight clipping trick to limit the range of parameters which contribute to convergence of loss function. Another [10] way is to bring the penalty area between *pg* and *pdata* as a regularization term to the loss function which force the gradient of these areas to be as close to 1 as possible. SNGAN [18] achieved control over the gradient range by regularizing the parameters matrix. In addition, both LSGAN [16] and RGAN [12] try to solve the problem that JS divergence causes the gradient to be 0. Specifically, LSGAN adopted least square loss function for discriminator which performs more stable. RGAN suggested using "relativistic discriminator" in the network architecture, the whole training process should contains increasing the probability that *pg* is real and decreasing the probability that *pdata* is real.

In addition to the improvement of the loss function, many studies have enhanced ability of discriminator by adding extra information to the generator. Thus, it is effective for stabilizing the training of the model to get the good results. For instance, CGAN, AC-GAN, info GAN [5, 17, 19]. They control the output of the generator by adding label information. Under this circumstance, generator needs to consider both latent code and label information to generate samples, which can be seen as putting constrains on the generator's generation space. Meanwhile, the discriminator also needs to consider whether the label information matches the generated data.

Some other methods have changed the structure of the model. DCGAN, improved DCGAN [21, 24] take advantage of the convolution layer instead of fully-connected layer to extract data features, and obtains high-quality generated samples.

BiGANs, VAEGAN, BEGAN, EBGAN [3, 8, 13, 27] contribute image quality by combining autoencoder structure and GANs. The autoencoder structure can ensure the reconstruction ability of the model, and the discriminator can make the image more realistic.

Inspired by these researches, this research introduces a GAN model based on data augmentation. Concretely, the ability of generator has been improved through an extra encoder. Under this design, generator could learn how to let generated data get rid of noise. Moreover, the ability of discriminator is also enhanced by adding multiple penalty areas as regularization term to the loss function. The details of the whole architecture are shown in section 3.

3. Proposed Methods

3.1. Data Augmentation for Generator

This research proposes a novel GAN model with data augmentation structure. To be specific, a noise vector is added into latent code *z* which is from prior distribution. Then it is reconstructed into *E*(*G*(*z*+*noise*)) with the help of an extra decoder. For the generator, it not only needs to generate *G*(*z*) and *G*(*z*+*noise*), but also needs to minimize the norm between *E*(*G*(*z*+*noise*)) and *z*. That is, a noise penalty term is added to the loss function of the generator which constrains the generation space of the generator to a certain extent, and improves the robustness of generator to noise. Under this circumstance, generator can not only learn to produce the image but also handles to filter noise from generated data.

In practice, the noise vector comes from a Gaussian distribution. To simplify the computation, square of L2(Euclidean) norm is adopted as metric in this experiment. Finally, the Penalty Term (NP) can be defined as below:

$$NP = \| E(G(z + noise)) - z \|_2^2 \tag{2}$$

So, the loss function of generator is shown as follows:

$$\min_G E_{x \in pg} - (D(x)) + \lambda_1 (NP) \tag{3}$$

The first term of (3) is as the same as original GAN that generator should maximum the output of fake data in discriminator. While λ_1 is the hyperparameter which balance the ability of generator between generation power and robustness to noise. If the λ_1 is too big, the generator focuses more on robustness to noise but loss generation ability to noise. However, if the λ_1 is too small, it will bring down the robustness. So that, the model should be evaluated under different value of λ_1 during experiments.

3.2. Multiple Penalty Areas under 1-lipchiz Constrains

Structurally, the discriminator is not quite different from original GAN, but now it is required to distinguish real samples from *G*(*z*) and *G*(*z*+*noise*). So, the loss function can be defined as:

$$\max_D E_{x \in pdata}(D(x)) - \frac{1}{2} E_{x \in pg}(D(x)) - \frac{1}{2} E_{x \in p(z+noise)}(D(x)) \tag{4}$$

The formula (4) explains that the real data should be discriminated as real, but *G*(*z*) and *G*(*z*+*noise*) should be discriminated as fake.

In practice, this model utilizes Wasserstein distance as distribution metric to measure these two distributions. For the purpose of making discriminator to be converged, it should satisfy $D \in 1-Lipschitz$. In response to this constraint, WGAN requires weight clipping to realize this limitation. To be specific, it sets an upper bound *c* and lower bound −*c* for gradient of weights. To be specific, if the parameter W is greater than *c*, change w equals to *c*; if the parameter W is less than −*c*, change w equals to −*c*. In this way, the value of discriminator between *pg* and *pdata* will not be pulled away infinitely.

However, weight clipping may cause long time for training [10] or result in gradient vanishing if the clipping parameter is too large or small. WGAN-GP Gulrajani *et al*. [10] proposed another way to achieve better performance which added a gradient penalty term ($GP = (\| (\nabla x D(x)) \| - 1)^2$) to the loss function which limits the gradient range of *x*. However, it is intractable to give enforcing to everywhere. In order to simplify the calculation, only the area between *pdata* and *pg* is penalized in the implementation.

In this work, since the generator produces extra data *G*(*z*+*noise*), the Gradient Penalty (GP) area should also contain area between real data and *G*(*z*+*noise*) intuitively. Under this condition, discriminator can better form smooth decision boundary and provides more effective gradient information to generator. So that, the data are sampled from these two straight lines of *p*(data), *p*(*z*) and *p*(*z*+*noise*). This process can be descried by Figure 1 as below:

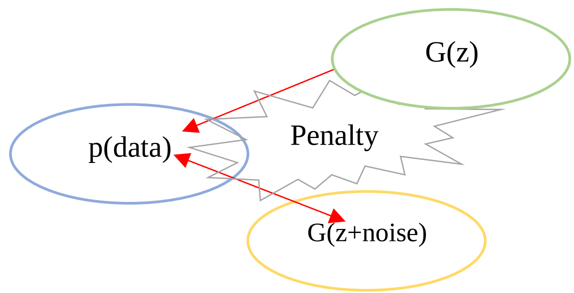

Figure 1. Multiple penalty areas.

All in all, the loss function of discriminator is shown as below:

$$\max_D E_{x \in pdata}(D(x)) - \frac{1}{2} E_{x \in pg}(D(x)) - \frac{1}{2} E_{x \in p(z+noise)}(D(x)) + \lambda_2 GP \tag{5}$$

In formula (5), discriminator needs to maximize the value of real data while reduce the value of *G*(*z*) and *G*(*z*+*noise*).Where λ_2 represents the penalty coefficient. In this work, the $\lambda_2=1$ is used, since it has been found it works very well on datasets celebA and LSUN towers. The difference between this model and previous models is shown as below.

According to Table 1, GAN and DCGAN utilize JS divergence and negative log-likelihood respectively, it is difficult to accurately measure the difference in distribution distance, while WGAN uses weight clipping which will limit the model ability and lead to unstable training. However, WGAN-GP and this model adopt the gradient penalty method to satisfy 1-lipschitz, which does not limit the model capability and is easy to calculate. The special envoy is the multiple penalty areas proposed in this study, which enables discriminator to form a better decision boundary and thus bring good gradient information to generator.

Table 1. The comparisons with previous models.

Model	Loss function
GAN	JS divergence
DCGAN	Negative log-likelihood
WGAN	Wasserstein with weight clipping
WAGAN-GP	Wasserstein with gradient penalty(between real data and fake data *G*(*z*)
Model in this study	Wasserstein with multiple penalty areas (between real data and fake data *G*(*z*), real data and G(Z+noise))

3.3. The Architecture of Networks

Generally speaking, generator is a deconvolution structure model with Leaky ReLU [15] nonlinearity that transfer latent code z into an image of 64×64 spatial resolutions. Our generator is shown as below, where "G", "D" and "E" indicates generator, discriminator and encoder respectively. The discriminator is a deep convolution neural network that classifies whether the input data is real or not. As mentioned before, the encoder aims to encode the *G*(*z+noise*) into latent code *z*, so the architecture is comparable to the discriminator, only the output layer is 100 dimensions. The data flow and hyper parameters are shown as Figure 2 and Table 2:

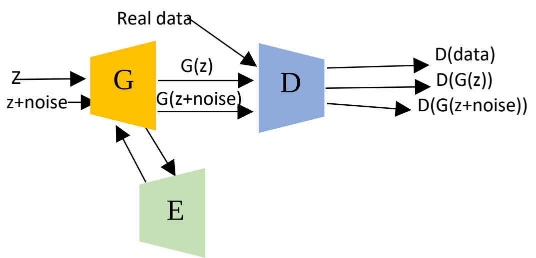

Figure 2. The data flow of our model.

Table 2. The hyperparameters of our model.

Nonlinearity (G)	LeakyReLU, LeakyReLU LeakyReLU tanh
Nonlinearity (D)	LeakyReLU, LeakyReLU LeakyReLU tanh
Nonlinearity (E)	LeakyReLU, LeakyReLU LeakyReLU tanh
Depth (G)	3
Depth (D)	3
Depth (E)	3
Batch norm (G)	True,True
Batch norm (D)	True,True
Batch norm (E)	True,True
Base filter count (G)	512,256,128
Base filter count (D)	64,128,256
Base filter count (D)	64,128,256

4. Experiments

The model proposed in this research was trained on both CelebA and LSUN tower datasets at 64×64 resolution which contains 202599 and 708201 images respectively for 300k epochs with 0.0001 learning rate and 256 batch size.

Due to the variety of criteria, there are several methods for measuring quality of image synthesis. Broadly speaking, the most common ways are Inception Score (IS) [2] and FID score [11]. Both two evaluate fidelity and quality of images based on Inception model. However, IS does not compare the generated images with the real images directly and only relying on ImageNet may miss useful features. FID can take into account the difference between the real distribution and the generated distribution. A lower FID means that the two distributions are closer, which means that the quality and diversity of the generated images is higher. Some other methods, such as RMS Contrast [4], is not widely used. In these experiments, FID score is used as the criteria for model evaluation. First, the model is trained on these datasets to figure out the best noise intensity based on FID score. The hope is to identify the ideal noise intensity for each dataset. In the second part, it is proved that penalty area including *p*(data), *G*(*z*) and *G*(*z+noise*) can benefit the results comparing with that only consider area of *p*(data) and *G*(*z*) Finally, this model is compared with DCGAN and WGAN-GP. FID score and image samples are used to provide a perceptually comparison.

4.1. The Effective Noise Intensity

The goal of this analysis is to find out the ideal noise intensity on celebA and LSUN tower datasets. Hence the parameter λ_1 are set as 0.1, 0.5, 0.9, 1.5, and 3 respectively to acquire the generated images at 64×64 resolutions and feed these generated samples to a pre-trained Inception network to calculate the FID score. The architecture and hypermeter are all the same, only the noise intensity is different.

Table 3. The FID score on celebA with different λ_1.

Noise λ_1/celebA	FID score
0.1	3.353
0.5	3.255
0.9	**2.941**
1.5	10.631
3	19.155

Table 4. The FID score on LSUN towers with different λ_1.

Noise λ_1/LSUN	FID score
0.1	3.304
0.5	**2.973**
0.9	6.717
1.5	18.281
3	31.173

The left column of Tables 3 and 4 indicates the different noise intensity and the right column is the FID score accordingly. It can be found that the noise intensity with 0.9 achieves the best result comparing to others in celebA. And noise intensity with 0.5 gets the lowest score in LSUN towers. These settings are used in the subsequent experiments.

4.2. Measuring the Effect of the Multiple Penalty Areas

It is meaningful that the penalty areas should contain *p*(data) *G*(*z*) and *G*(*z*+*noise*) (multiple penalty areas) in this model. Under this circumstance, discriminator can obtain more information to update the parameters which benefit the results.

In this section, the noise intensity parameter is set as 0.9 and 0.5 which are the best settings of experiment 5.1. To evaluate the effect of multi-penalty area, it has been compared to the penalty area only contains *p*(data) and *G*(*z*). The results are shown as below:

Table 5. The FID score on celebA and LSUN under different penalty areas.

Dataset	Noise	FID score	Penalty area
celebA	0.9	3.356	*p*(data), *G*(*z*)
celebA	0.9	2.941	*p*(data), *G*(*z*) and *G*(*z*+*noise*)
LSUN	0.5	3.318	*p*(data), *G*(*z*)
LSUN	0.5	2.973	*p*(data), *G*(*z*) and *G*(*z*+*noise*)

Table 6. The FID score on LSUN towers under different penalty areas.

Dataset	Noise	FID score	Penalty area
LSUN towers	0.5	3.318	*p*(data), *G*(*z*)
LSUN towers	0.5	**2.973**	*p*(data), *G*(*z*) and *G*(*z*+*noise*)

It can be found from Tables 5, and 6 that the model with regular penalty area only gets 3.356 and 3.318 in celebA and LSUN respectively. Nevertheless, the model with multiple penalty areas can get the lower FID score both in celebA and LSUN. This proves that the quality of images can be improved with multiple penalty areas.

4.3. Comparison to Previous Models

In this section, this model was compared with DCGAN WGAN-GP based on FID score. DCGAN tries to improve the quality of generated images and introduces the concept of deconvolution into the model. WGAN-GP adopts the same Wasserstein loss function as the model proposed in this study, but only contains regular penalty regions since there is no data augmentation. Basically, the results of comparison on celebA and LSUN towers are shown as below:

Table 7. The comparison to different models in celebA and LSUN towers.

DCGAN	WGAN (GP=1)	OURS (λ_1=0.9,GP=1)
		
FID		
6.198	3.327	2.973

Table 8. The comparison to different models in LSUN towers.

DCGAN	WGAN (GP=1)	OURS (λ_1=0.9,GP=1)
		
FID		
8.026	3.327	2.941

According to the results of Tables 7 and 8, DCGAN cannot get better results under the same number of iterations, and the generated pictures are blurry. Due to the lack of noise reduction ability and the help of multiple penalty areas, the results of WGAN-GP are clearer than DCGAN, but there is more or less noise. Intuitively, the model proposed in this research is the best which also has the smallest FID score.

5. Conclusions

This research introduces a mechanism of data augmentation which can be achieved by considering extra data *G*(*z*+*noise*). Thereby when using Wasserstein distance as the metric of discriminator, discriminator needs to give the penalty on multiple areas. That is it not only needs to enforce constraints between the real data and $G(z)$, but also need to limit gradient norm between the real data and $G(z+noise)$. Experiments have proved that this method can help discriminator to learn smoother decision boundaries and improve the quality of generated images. However, this method requires the design of an additional encoder and the complexity of the model structure is slightly increased. Moreover, the model has only been verified on the celebA and LSUN towers at resolution 64*64, and only one evaluation standard is used. In future work, it is hoped that this method can be extended to higher resolution or text generation, or to verify the effect of this method in the loss function of other GAN models.

Reference

[1] Arjovsky M., Chintala S., and Bottou L., "Wasserstein Generative Adversarial Networks," *in Proceedings of the 34th International Conference on Machine Learning*, Sydney, pp. 214-23, 2017.

[2] Barratt S. and Sharma R., "A Note on the Inception Score," *ArXiv:180101973 [Cs, Stat]*, 2018.

[3] Berthelot D., Schumm T., Metz L., "BEGAN: Boundary Equilibrium Generative Adversarial Networks," *ArXiv:170310717 [Cs, Stat]*, 2017.

[4] Bhuiyan A. and Khan A., "Image Quality Assessment Employing RMS Contrast and Histogram Similarity," *The International Arab Journal of Information Technology*, vol. 15, no. 6, pp. 983-989, 2018.

[5] Chen X., Duan Y., Houthooft R., Schulman J., Sutskever I., and Abbeel P., "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[6] Dinh L., Krueger D., and Bengio Y., "NICE: Non-linear Independent Components Estimation," *ArXiv:14108516 [Cs]*, 2015.

[7] Dinh L., Sohl-Dickstein J., and Bengio S., "Density Estimation Using Real NVP," *ArXiv:160508803 [Cs, Stat]*, 2017.

[8] Donahue J., Krähenbühl P., Darrell T., "Adversarial Feature Learning," *ArXiv:160509782 [Cs, Stat]*, 2017.

[9] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., and Bengio Y., "Generative Adversarial Networks," *Communications of the ACM*, vol. 13, pp. 139-144, 2020.

[10] Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., and Courville A., "Improved Training of Wasserstein GANs," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[11] Heusel M., Ramsauer H., Unterthiner T., Nessler B., and Hochreiter S., "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] Jolicoeur-Martineau A., "The Relativistic Discriminator: A Key Element Missing From Standard GAN," *ArXiv:180700734 [Cs, Stat]*, 2018.

[13] Larsen A., Sønderby S., Larochelle H., and Winther O., "Autoencoding Beyond Pixels Using A Learned Similarity Metric," *in Proceedings of the 33rd International Conference on International Conference on Machine Learning*, New York, pp. 1558-1566, 2016.

[14] Liu Z., Luo P., Wang X., and Tang X., "Deep Learning Face Attributes in the Wild," *ArXiv:14117766 [Cs]*, 2015.

[15] Maas A., Hannun A., and Ng A., "Rectifier Nonlinearities Improve Neural Network Acoustic Models," *in Proceedings of the International Conference on Machine Learning*, Atlanta, 2013.

[16] Mao X., Li Q., Xie H., Lau R., Wang Z., and Smolley S., "Least Squares Generative Adversarial Networks," *in Proceedings of the IEEE International Conference on Computer Vision*, Venice, pp. 2794-2802, 2017.

[17] Mirza M. and Osindero S., "Conditional Generative Adversarial Nets," *ArXiv:14111784 [Cs, Stat]*, 2014.

[18] Miyato T., Kataoka T., Koyama M., and Yoshida Y., "Spectral Normalization for Generative Adversarial Networks," *ArXiv:180205957 [Cs, Stat]*, 2018.

[19] Odena A., Olah C., and Shlens J., "Conditional Image Synthesis with Auxiliary Classifier GANs," *in Proceedings of the 34th International Conference on Machine Learning*, Sydney, pp. 2642-2651, 2017.

[20] Van den Oord A., Kalchbrenner N., Espeholt L, Kavukcuoglu K., Vinyals O., and Graves A., "Conditional Image Generation with PixelCNN Decoders," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[21] Radford A., Metz L., and Chintala S., "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *ArXiv:151106434 [Cs]*, 2016.

[22] Rezende D. and Mohamed S., "Variational Inference with Normalizing Flows," *in Proceedings of the 32nd International Conference on Machine Learning*, Lille, pp. 1530-1538, 2015.

[23] Rifai S., Vincent P., Muller X., Glorot X., and Bengio Y., "Contractive Auto-Encoders: Explicit Invariance during Feature Extraction," *in Proceedings of the 28th International Conference on Machine Learning*, Bellevue, pp. 833-840, 2011.

[24] Salimans T., Goodfellow I., Zaremba W., Cheung V., Radford A., Chen X., and Chen X., "Improved

Techniques for Training GANs," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[25] Vincent P., Larochelle H., Bengio Y., and Manzagol P., "Extracting and Composing Robust Features with Denoising Autoencoders," *in Proceedings of the 25th International Conference on Machine Learning*, Helsinki Finland, pp. 1096-1199, 2008.

[26] Yu F., Seff A., Zhang Y., Song S., Funkhouser T., and Xiao J., "LSUN: Construction of a Large-Scale Image Dataset using Deep Learning with Humans in the Loop," *ArXiv:150603365 [Cs]*, 2016.

[27] Zhao J., Mathieu M., and LeCun Y., "Energy-based Generative Adversarial Network," *ArXiv:160903126 [Cs, Stat]*, 2017.

Li Chen is a PhD candidate at Universiti Sains Malaysia. His research interests include deep learning, generative model, and image processing and so on.

Huah Yong Chan is an Associate Professor at Universiti Sains Malaysia. His research interests include artificial intelligence, high performance computing.