Generative Adversarial Networks for single image with high quality image

Liquan Zhao[1*], and Yupeng Zhang[1]
[1] Key Laboratory of Modern Power System Simulation and Control and Renewable Energy Technology, Ministry of Education, Northeast Electric Power University
[e-mail: zhaoliquan@neepu.edu.cn]
[*]Corresponding author: Liquan Zhao

Received May 20, 2021; revised August 24, 2021; accepted October 19, 2021; published December 31, 2021

Abstract

The SinGAN is one of generative adversarial networks that can be trained on a single nature image. It has poor ability to learn more global features from nature image, and losses much local detail information when it generates arbitrary size image sample. To solve the problem, a non-linear function is firstly proposed to control downsampling ratio that is ratio between the size of current image and the size of next downsampled image, to increase the ratio with increase of the number of downsampling. This makes the low-resolution images obtained by downsampling have higher proportion in all downsampled images. The low-resolution images usually contain much global information. Therefore, it can help the model to learn more global feature information from downsampled images. Secondly, the attention mechanism is introduced to the generative network to increase the weight of effective image information. This can make the network learn more local details. Besides, in order to make the output image more natural, the TVLoss function is introduced to the loss function of SinGAN, to reduce the difference between adjacent pixels and smear phenomenon for the output image. A large number of experimental results show that our proposed model has better performance than other methods in generating random samples with fixed size and arbitrary size, image harmonization and editing.

Keywords: Artificial Intelligence, Generative adversarial networks, Attention mechanism, Random samples.

1. Introduction

In recent years, with the rapid development computing power, generative model has been one of the hottest topics in artificial intelligence [1]. Generative model is mainly used to train neural network model to generate images that are close to the training images. The early generative models are mainly explicit modeling of data distribution, such as Variable Auto-Encoder (VAE), Deep Belief Network (DBN), Deep Boltzmann Machines (DBM) and Autoregressive model (AR model) [2]. They can't generalize the generation results well. To solve these problems, Goodfellow, et al. firstly proposed the Generative Adversarial Networks (GANs) in 2014 [3]. It includes two models: generative model and discriminative model. In the training process, the goal of generative model is to generate a fake image as similar as the training image to deceive the discriminator. The goal of discriminative model is to estimate the probability that the input image of discriminator come from training image or generator. If the discriminator cannot judge whether the input image of discriminator is generated by generator or a training image, the generative model is optimal.

Recently, GANs has been successfully used in many fields, such as computer vision (CV), natural language processing (NLP) and so on [4-6]. In the field of computer vision, such as image generation, image editing, image inpainting, semantic segmentation, super-resolution and video generation. In the field of natural language processing, such as image generation from text, machine translation, dialogue generation and so on. It is also widely used in other fields, such as anomaly detection of medical images, 3D object generation and seismic denoising. Most of GANs methods are trained on many images. They consume much time and requires powerful computing. To make the GAN can be trained on a single image, the SinGAN method is proposed [7]. It can be trained on a single image, and suits many different image manipulation tasks. To further more improve the quality of generated image and reduce training time, we proposed AGAN on the basis of SinGAN. Our proposed method has better performance than other methods in the aspect of generating random samples, harmonization and editing. Our main contributions are summarized as followings:

1. We design a function to control the downsampling ratio, which increases the low-resolution images proportion in downsampled images.

2. We design a plug-and-play Residual Pixel Attention (RPA) module by introducing the attention mechanism to increase the effective information weight. It makes the network pay more attention to the learning of important features without causing the loss of information.

3. We design a new loss function by introducing the TVLoss function. It smoothes the generated image by minimizing the difference between adjacent pixel values and improves the naturalness of the generated image.

In Section 1, we introduce the generative adversarial networks and our contributions. In section 2, we describe the developments related to generative adversarial networks. In section 3, we explain our proposed method. In Section 4, we illustrate and discuss our simulation results. In section 5, we summarize the conclusions.

2. Related work

The improved GANs methods can be roughly divided into two categories: the improvement of loss function and the improvement of network structure. The typical improvements of loss function include Least Squares GANs (LSGAN), Wasserstein GANs (WGAN) and WGAN-GP [8-10], etc. The LSGAN is proposed by using the least square loss function instead of the cross-entropy loss function [8]. Although it improves the quality of generated image and

stability of training process, the problem of gradient disappearance still exists in the generator. Arjovsky et al. proposed WGAN by using Wasserstein distance instead of JS divergence to measure the distance between the real distribution and the generated distribution [9]. WGAN uses weight clipping directly to deal with Lipschitz constraints. The absolute values of all parameters of the discriminator are checked to monitor whether they exceed a threshold. If they exceed a threshold, they will be pulled back to the specified range. It solves the problems of training instability and mode collapse, and makes the generated results more diverse. WGAN-GP is proposed on the basis of WGAN [10]. It uses gradient penalty instead of weight clipping to carry out Lipschitz constraint. Lipschitz constraint requires that the gradient of the discriminator should not exceed K, while gradient penalty is to set an additional loss term to realize the connection between the gradient and K. It solves the problems of WGAN training difficulty and slow convergence.

The typical improvements of network structure mainly include Deep Convolutional GANs (DCGAN) [11], Conditional GANs (CGAN) [12], Laplacian Pyramid GANs (LAPGAN) [13], StyleGAN [14], etc. Radford et al. proposed DCGAN, which combines CNN with GAN. The principle of DCGAN is the same as GANs. It just replaces G and D with two convolutional neural networks (CNN) [15], cancels all pooling layers, removes FC layer, and uses ReLU fucntion as activation function in G network and LeakyReLU fucntion as activation function in D network to improve sample quality and convergence speed. It still does not solve the problem of GAN training instability. The CGAN introduces condition variable into both generative model and discriminative model [12], which can guide the data generation process. These conditional variables can be based on a variety of information, such as category labels, etc. It solves the problem of direction uncertainty of the general model generative GAN, and makes the generated output more in line with our requirements. Besides, Denton et al. proposed LAPGAN [13], which is based on Laplacian pyramid. Each layer of the pyramid learns the residuals between adjacent layers. In other words, the generation of high-resolution images is achieved by using low-resolution images as conditions to generate residuals, and then the low-resolution image is upsampled and then summed with the residual to obtain the high-resolution image. The essence is to stack CGAN to get the resolution we want. It solves the problem that GAN can only generate low pixel images, and can generate high pixel images. Karras et al. proposed StyleGAN [14], which uses a progressive layer network to learn low-resolution image generation first, and then to learn higher resolution image generation with the deepening of network layers. Progressive layer can control different visual features of the image. The lower the layer and resolution, the coarser the features it affects. This enables the network to generate high quality and realistic images. Tilon et al. used the generative adversarial network to observe the image and detect the damage of buildings after the disaster [16]. Because this method can be transferred to different types of damage or geographical locations, it has stronger applicability. Dong et al. designed an end-to-end generative adversarial network with fusion discriminator to remove haze from haze-affected images [17]. This method skips the estimation of intermediate parameters and directly performs end-to-end image dehazing, which can generate more realistic and natural dehazing images. Wu et al. use generative adversarial network to detect malicious social robots [18]. This method expanded the unbalanced data sets by generative adversarial networks, improved the detection of social robots, and achieved better detection results.

The above networks are trained on big data set. There are thousands of images in data set. It requires powerful hardware and consumes much time to train the network. To solve the problem, The GAN models that can be trained by using a single image are proposed. Chuan Li et al. proposed the Markovian GAN (MGAN) [19] by using a feed forward and interleaved

convolution network to capture the characteristic statistics of Markov slices and directly generate output of arbitrary size. It can decode noise into realistic texture, or decode photos into art painting. Although it can generate high quality images, but the effect of this method on non-texture data is not very good. Spatial GAN (SGAN) [20] is the first to apply fully unsupervised GAN to texture synthesis. It has good scalability and excellent results in texture. It can only deal with some types of texture. Therefore, Bergmann et al. proposed Periodic Spatial GAN (PSGAN) [21], which extends the structure of input noise distribution by constructing tensor with different dimensions. The PSGAN can deal with different textures flexibly, but it cannot learn too complex texture information. To deal with more complex textures, Yang Zhou et al. proposed Non-Stationary Texture Synthesis by Adversarial Expansion [22]. It cuts any texture block from the sample image, inputs it into the network for training, and then expands it to arbitrary size. Although it can deal with more complex textures, it cannot learn large global structure information well. Besides, Shocher et al. proposed the Internal GAN (InGAN) [23]. It achieves the expansion and stretching of the image by capturing the internal colorpatch distribution of the image, such as producing output with significantly different size, shape and aspect ratio. It solves the problem that previous methods cannot learn large global structure information, but it cannot do other tasks, such as image editing, image harmonization and so on. Except for InGAN, these methods are only used in the context of texture generation. They cannot generate meaningful image sample when the training image is non-texture image. Especially, they cannot generate random samples.

Tamar et al. firstly proposed an unconditional generative model (SinGAN) [7]. Comparing with other GAN methods for a signal image, it can be trained on a single general natural image, and suits many different image manipulation tasks. The goal of SinGAN is to learn an unconditional generative model. When a random noise is used as input, an image similar to the original image will be generated. SinGAN uses a pyramid model that is different from LAPGAN. SinGAN uses an unsupervised structure, which inputs the generated images of previous stage into the generator after they are upsampled. In addition, SinGAN adds a residual structure to improve the network performance. Besides, the ConSinGAN [24] is also proposed on the basis of SinGAN to improve the quality of generated image and training speed. The SinGAN and ConSinGAN are only two methods that are most relevant to our methods.

3. Proposed Method

3.1 Downsampling method

Low-resolution image training is useful to extract global image feature (such as the overall outline). The downsampling ratio is fixed in SinGAN method. If we want to extract more global image feature, we must increase the number of downsampling. This will consume much time to train the network. To extract global image feature information, we proposed a new method to control the downsampling ratio. This can make the low-resolution images obtained by downsampling have higher proportion in all downsampled images. In SinGAN, the size of downsampled image is expressed as following:

$$x_k = x_0 \times r^k, \quad k=1,2,...,N \tag{1}$$

where x_0 is the size of natural image, r is a downsampling factor ranging from 0 to 1. Increasing its value can increase the size of the downsampled image. During training, r is a constant, and the size of downsampled image is changed only by changing the value of k. In SinGAN, r is set to 0.75. N is the maximum number of downsampling, x_k is the size of

kth downsampled image. The downsampling ratio is constant in SinGAN.

To increase the proportion of low-resolution image without increasing the number of downsampling, we propose three methods to control the size of downsampled image. The sizes of downsampled image based on our three methods are expressed as followings:

$$x_k = x_0 \times r^{(N-1)*\cos(\frac{\pi}{2(N-1)}(N-k))+1}, \quad k = 1,\ldots,N \tag{2}$$

$$x_k = x_0 \times r^{-\frac{(N-k)^2}{N-1}+N}, \quad k = 1,\ldots,N \tag{3}$$

$$x_k = x_0 \times r^{\sqrt{(N-1)^2-(N-k)^2}+1}, \quad k = 1,\ldots,N \tag{4}$$

where x_0 is the size of natural image, r is a downsampling factor ranging from 0 to 1. Its value is obtained by calculating the ratio of the minimum image size to the input image size. N is the maximum number of downsampling, x_k is the size of kth downsampled image. Based on (1)-(4), we can deduce that

$$\log(x_k / x_0) = k\log(r) \tag{5}$$

$$\log(x_k / x_0) = [(N-1)*\cos(\frac{\pi}{2(N-1)}(N-k))+1]\log(r) \tag{6}$$

$$\log(x_k / x_0) = [-\frac{(N-k)^2}{N-1}+N]\log(r) \tag{7}$$

$$\log(x_k / x_0) = [\sqrt{(N-1)^2-(N-k)^2}+1]\log(r) \tag{8}$$

where (5)-(8) are deduced from (1)-(4), respectively. If we set the r=0.5 in (5)-(8), then we can get the relationship between the number of downsampling and size of downsampled image in (5)-(8). They are shown in **Fig. 1**. Equation (5) is the original method, and the (6)-(8) are our proposed methods. For the same number of downsampling, downsampled image size based on (8) is the smallest, followed by (7) and (6). The downsampled image size based on original method is the largest. This means that downsampled image sizes based on our proposed three methods are smaller than original method. It makes the low-resolution image obtained by downsampling has higher proportion in all downsampled images. It is useful to extract more global image feature information to improve the quality of generated image.

3.2 Generative Network

The generative network is used to generate expected image according input signal. The generative network contains several generators in SinGAN method. Due to that the generator is only composed of simple convolution layers, so it is limited to learn more important information. In generating arbitrary size image, it will lose some important information.

To solve the problem, we introduce the attention mechanism to the generator. It can make the generator pay more attention to learn important information and ignore the irrelevant information. The input of first generator is only noise, and there is no useful information about the training image to learn. Therefore, we keep the first generator unchanged. The network structure of other generators is proposed and shown in **Fig. 2**. Using the attention mechanism can improve the quality of the generated image and the performance of the network, so that the network pays more attention to the high-information features, and will not ignore the important details.

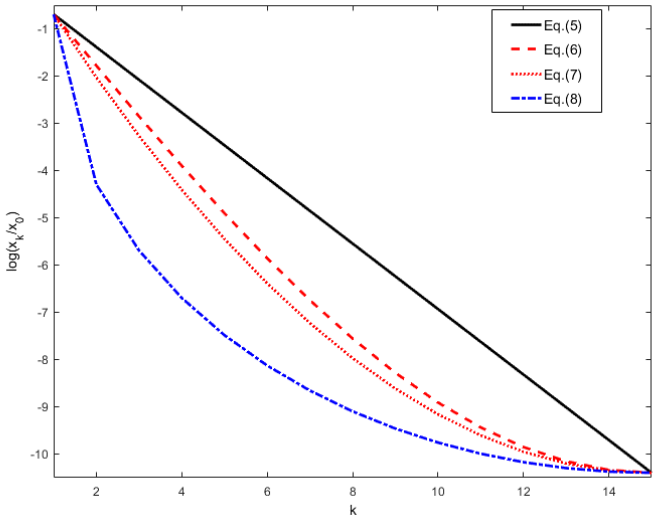

Fig. 1. Relationship between the number of downsampling and size of downsampled image.

For this network, we design a plug-and-play Residual Pixel Attention (RPA) module, and the network structure is shown in **Fig. 3**. The Residual Pixel Attention no longer treats each pixel equally, but focuses on more important pixels. The Residual Pixel Attention makes each pixel present different weights, and the more important information area will have higher weights. This makes the network pay more attention to the learning of important information. It can be expressed as:

$$PA = \sigma_1(Conv(\lambda_4(Conv(\lambda_3(Conv(\lambda_2(Conv(\lambda_1(Conv(x)))))))))) \tag{9}$$

$$\hat{x} = \sigma_2(Conv(\lambda_4(Conv(\lambda_3(Conv(\lambda_2(Conv(\lambda_1(Conv(x)))))))))) \tag{10}$$

$$F = \hat{x} \times PA + x \times (1 - PA) \tag{11}$$

Where x is the input of the RPA module. $Conv()$ is the convolution layer in the RPA module. $\lambda()$ is the LeakyRelu activation function. $\sigma_1()$ is the Sigmoid activation function and $\sigma_2()$ is the Tanh activation function. PA is the ouput of the attention part. And F is the output of the RPA module. When the pixel value of some unimportant features in the image approaches 0, pixel attention will destroy these features in the image. Therefore, we add residual structure to pixel attention, which can avoid feature loss and improve the stability of training.

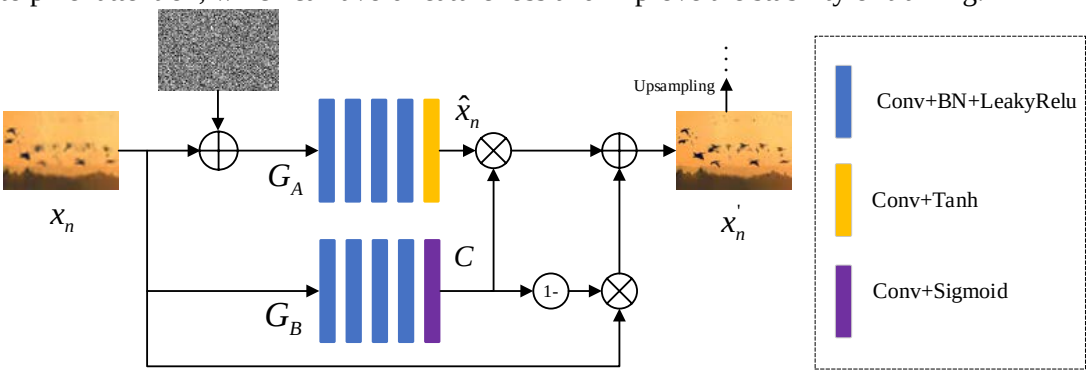

Fig. 2. Improved generative network structure by introducing attention mechanism

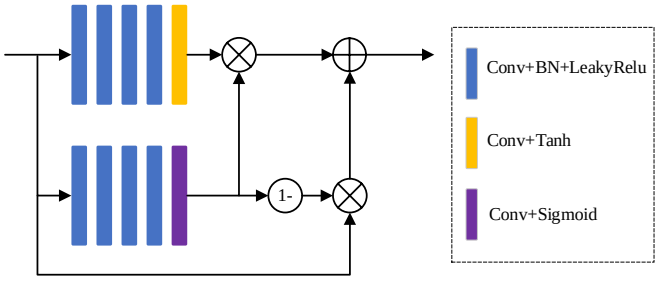

Fig. 3. The network structure of the Residual Pixel Attention module

The x_n is the upsampled image of the output of previous generator. The sum of upsampled image and noise is used as the input of network G_A. The network G_A has five layers. The first four layers have the same network structure that is composed of one convolution layer, one batch normalization layer and an activation layer with LeakyReLU function. The fifth layer is composed of a convolution layer and an activation layer with Tanh function. The output of network G_A is $\hat{x}_n$. To learn more important information from the input image, we design the new network G_B. The G_B is composed of five layers. The first four layers have the same network structure that is composed of one convolution layer, one batch normalization layer and an activation layer with LeakyReLU function. The last layer is composed of a convolution layer and an activation layer with Sigmoid function. The main differences between G_B and G_A are the input image and the last layer. The input of G_B is only upsampled image, and the output is C. The size of C is the same with the output image $\hat{x}_n$. Output C can be seen as the weight image. The pixel value of C is the pixel weight of each pixel in input image $\hat{x}_n$. The more information the feature contains, the larger weight of pixel that the feature is corresponding to is. Therefore, the product of C and $\hat{x}_n$ can improve the proportion of important information in image $\hat{x}_n$, and reduce the proportion of unimportant information. This makes the network pay more attention to the important information and ignore the unimportant information in the process of training.

The pixel value is between 0 and 1 in image C. If we directly use the product of C and $\hat{x}_n$ as the output of generator, it will destroy attributes of some feature when the pixel value of C is close to zero. This will increase difficulty of model training. Therefore, we use residual network structure to overcome the problem. The output of residual network is $x_n \times (1-C)$. We use the sum of output of residual network and product of C and $\hat{x}_n$ as the output of generative network. Therefore, the output of generator is expressed as:

$$x_n^{'} = \hat{x}_n \times C + x_n \times (1-C) \tag{12}$$

Where $x_n^{'}$ is output of nth generative network. Based on (12), if some pixel value of C are close to zero, the $x_n^{'}$ is close to x_n. On the contrary, the $x_n^{'}$ is close to $\hat{x}_n$. This avoids losing information and improves the quality of generated image.

3.3 Loss function

The shadowing phenomenon often happens in junction of the pasted image and background in the process of image fusion. This affects the quality of image fusion. To make the fusion part smoother, we introduce the TVLoss function to the loss function, to reduce the difference between adjacent pixel values. It can make the fusion image more natural. The TVLoss function is expressed as following:

$$L_{tv} = \sum_{i,j}[(x'_{i,j-1} - x'_{i,j})^2 + (x'_{i+1,j} - x'_{i,j})^2] \tag{13}$$

where x' is the image that generated by generator G, $x'_{i,j}$ is the pixel value of the ith row and jth column of x'. The more the adjacent pixel values are close to each other, the smaller the value of TVLoss function is, and the more natural the generated image is. Introducing the (13) to the original loss function, we can deduce the proposed loss function of proposed method. It is expressed as following:

$$Loss = \min_{G_n} \max_{D_n} L_{adv}(G_n, D_n) + L_{rec}(G_n) + L_{tv}(G_n) \tag{14}$$

where G_n and D_n are the generator and discriminator of nth stage, respectively. $L_{adv}(G_n, D_n)$ and $L_{rec}(G_n)$ are the loss function of SinGAN method. They are expressed as followings:

$$L_{adv}(G_n, D_n) = E[D_n(x'_n)] - E[D_n(y_n)] + \lambda E[(\left\|\nabla D_n(\overline{x}_n)\right\|_2 - 1)^2] \tag{15}$$

$$L_{rec}(G_n) = \begin{cases} \left\|G_n(0, x_n) - y_n\right\|, n < N \\ \left\|G_N(z^*) - y_N\right\|^2, n = N \end{cases} \tag{16}$$

where x' is the output image of G_n. y_n and y_N is the real image of nth stage and Nth stage, respectively. $\overline{x}_n = a \times x_n + (1-a) \times y_n$ ($a \in (0,1)$). z^* is random noise. Minimizing $L_{adv}(G_n, D_n)$ makes the model pay more attention to the details of the image. Minimizing $L_{rec}(G_n)$ is to make final output image close to the original image, which also improves the stability of training.

4. Simulation and discussion

In this section, we mainly make a comparison between SinGAN, ConSinGAN and our three methods (named AGAN(1) based on (2), named AGAN(2) based on (3), and named AGAN(3) based on (4)) in the aspect of random samples generation, random samples of arbitrary sizes generation, image harmonization and editing. The trained images are randomly selected from the internet and randomly selected from the LSUN dataset. The size of the image is arbitrary. All methods use the same model parameters. In this simulation system, we use Ubuntu 18.04 system and Intel Xeon e5-2678 V3 processor for simulation. The GPU is NVIDIA GeForce GTX 2080ti, and the deep learning framework is PyTorch.

4.1 Random samples with fixed sizes

In this section, we test the performance of random samples generation with the condition that the size of input image equals to the output image. We use stone tower image and twin towers

image that have simple structure as training image, respectively. The training images and the images generated by SinGAN, ConSinGAN and our three methods are shown in **Fig. 4** and **Fig. 5**. In **Fig. 4**, the most parts of the images generated by SinGAN are covered by background, and lose much important information. Although we can recognize the stone tower from the images generated by ConSinGAN, the details of stone tower are lost and the image of tower footing is also destroyed. Compared with SinGAN and ConSinGAN, the images generated by our three methods are closer to the training image. They contain more important and detailed information of training image. Among them, AGAN(3) has the best effect, followed by AGAN(2) and AGAN(1). In **Fig. 5**, the spire and parts of body are distorted in the twin tower images generated by SinGAN. For the twin towers images generated by ConSinGAN method, it generates more towers than the training images. Although the twin towers images generated by AGAN methods also are distorted, the distortion is limited, and we can still recognize the twin towers from the generated images. Among them, the distortion of AGAN(3) is the smallest, followed by AGAN(2) and AGAN(1). Based on **Fig. 4** and **Fig. 5**, they show that the images generated by AGAN methods are closer to the training images than other methods. This means that our proposed methods have better performance than others in random samples generation.

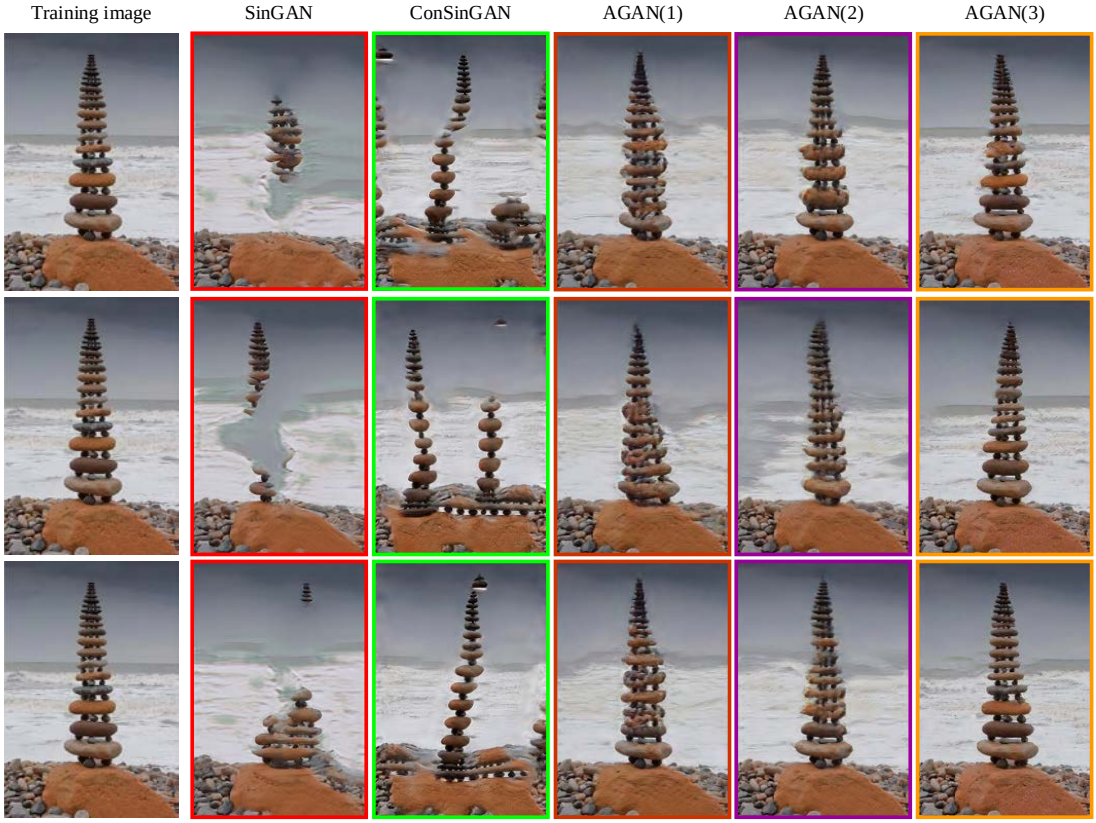

Fig. 4. Stone tower training image and generated images by SinGAN, ConSinGAN and our proposed methods (AGAN(1), AGAN(2) and AGAN(3))

We also use the evaluation index SIFID (Single Image FID) that used in SinGAN to compare the difference between the generated image and the original image, which calculates the average score of 50 generated images. The smaller the value is, the better the quality of the generated image is. We randomly generate 50 stone tower images and 50 twin towers images, and compute the average of their SIFID scores. The results are shown in **Table 1**. The SIFID values are 0.076, 0.051, 0.037, 0.032 and 0.028 for SinGAN, ConSinGAN, AGAN(1), AGAN(2) and AGAN(3), respectively. The AGAN methods have the smallest SIFID, followed by ConSinGAN. In addition, we also use another measure: structural similarity (SSIM) to calculate the correlation between image pixels. The larger the value is, the better the quality of the generated image is. The SSIM values are 0.6537, 0.7062, 0.7351, 0.7535 and 0.8097. The AGAN methods have the largest SSIM value. We also use RMSE to calculate the arithmetic square root of the expected value of the square of the difference between the generated image and the original image. The smaller the value is, the better the quality of the generated image is. The RMSE values are 0.2566, 0.2042, 0.1864, 0.1787 and 0.1543. The AGAN methods have the minimum RMSE value. The Trainning times are about 80min, 35min and 40min for SinGAN, ConSinGAN and AGAN, respectively. The ConSinGAN has the fastest training speed, followed by AGAN. These show that the AGAN has the best performance than others in generating image, and faster training speed than SinGAN method.

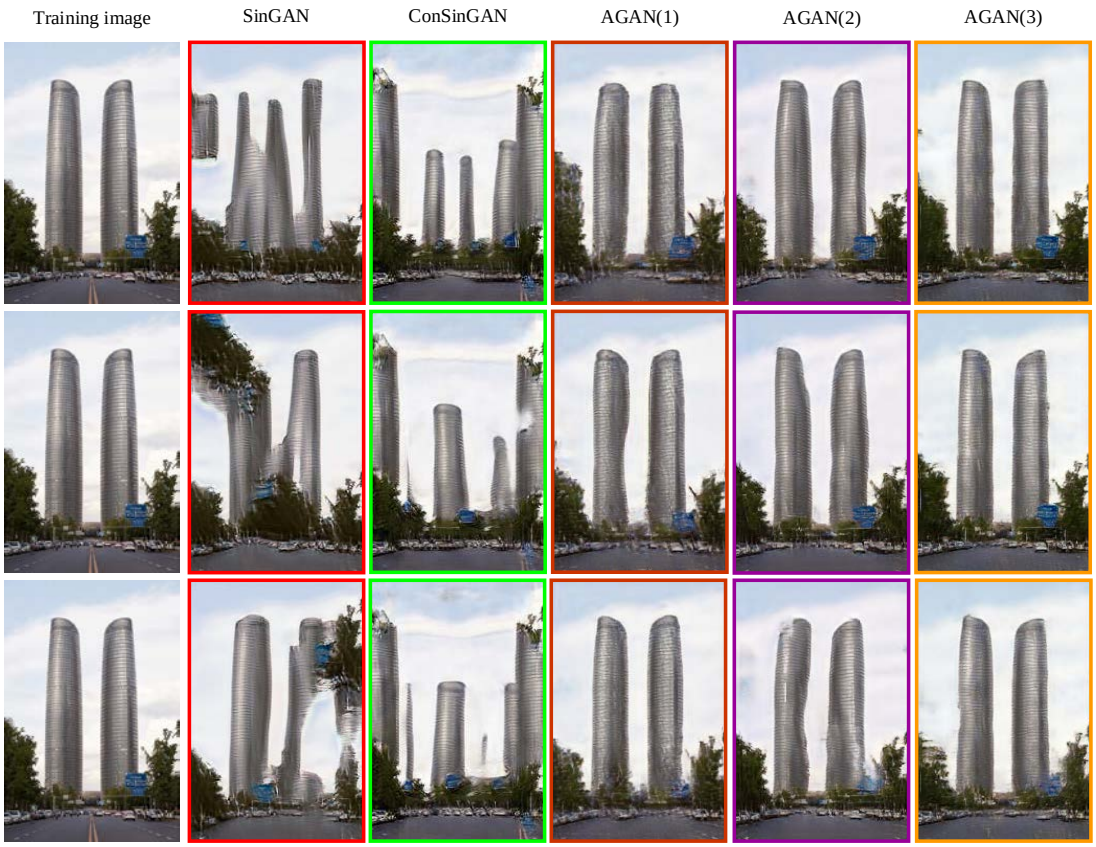

Fig. 5. Twin towers training image and generated images by SinGAN, ConSinGAN and our proposed method (AGAN(1), AGAN(2) and AGAN(3))

Table 1. Comparison of SIFID and running time for stone tower and twin towers images

Model	SIFID	SSIM	RMSE	stages	Train times
SinGAN	0.076 ± 0.004	0.6537	0.2566	8-10	$\approx 80min$
ConSinGAN	0.051 ± 0.007	0.7062	0.2042	6	$\approx 35min$
AGAN(1)	0.037 ± 0.008	0.7351	0.1864	6	$\approx 40min$
AGAN(2)	0.032 ± 0.007	0.7535	0.1787	6	$\approx 40min$
AGAN(3)	0.028 ± 0.005	0.8097	0.1543	6	$\approx 40min$

In **Fig. 6**, we use a complex building that is Alcázar of Segovia as training image to train the network and generate images. The images generated by AGAN methods are closer to the training image than others. AGAN can learn more global characteristics of the training image, so AGAN methods have better results than others. The corresponding average SIFID, SSIM, RMSE and running time is shown in **Table 2**. The AGAN methods have the smallest SIFID (0.069, 0.063 and 0.061), followed by ConSinGAN (0.081) and SinGAN methods (0.089 and 0.104). The AGAN methods have the smallest RMSE (0.1822, 0.1798 and 0.1662), followed by ConSinGAN (0.2133) and SinGAN methods (0.2573 and 0.5832). The AGAN methods have the largest RMSE (0.7763, 0.7859 and 0.8521), followed by ConSinGAN (0.7522) and SinGAN methods (0.7038 and 0.5121). The training time are about 85min, 52min, 39min and 45 min for SinGAN, SinGAN with 6 stages, ConSinGAN and AGAN, respectively. The ConSinGAN has the fastest training speed, followed by AGAN. When the stage of SinGAN equals to 6, its SIFID is 0.104. To generate higher quality image, the SinGAN requires more stages. This is also why it consumes more time to train the network.

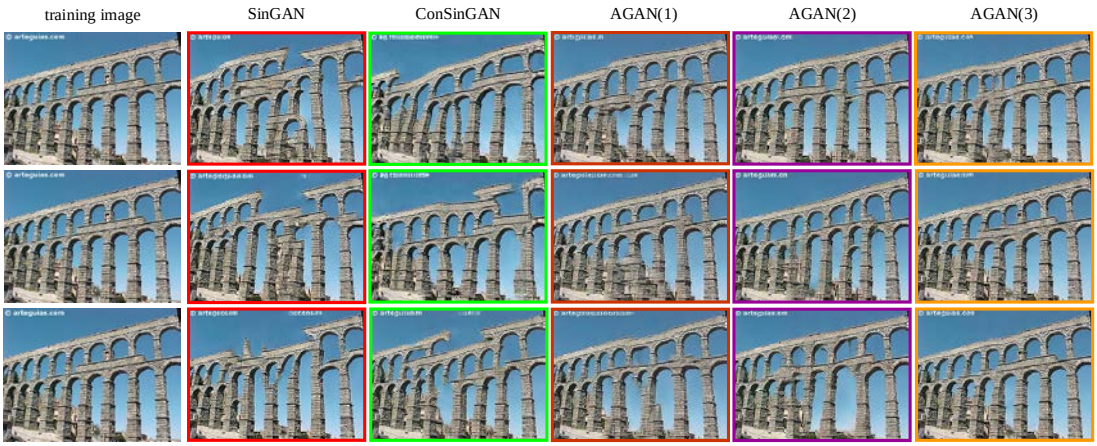

Fig. 6. Alcázar of Segovia image and generated images by SinGAN, ConSinGAN and our proposed method (AGAN(1), AGAN(2) and AGAN(3))

Table 2. Comparison of SIFID and training time for Alcázar of Segoviaimage

Model	SIFID	SSIM	RMSE	stages	Train times
SinGAN	0.089 ± 0.005	0.7038	0.2573	8-10	$\approx 80min$
SinGAN(6 stages)	0.104 ± 0.009	0.5121	0.5832	6	$\approx 52min$
ConSinGAN	0.081 ± 0.005	0.7522	0.2133	6	$\approx 39min$
AGAN(1)	0.069 ± 0.007	0.7763	0.1822	6	$\approx 45min$
AGAN(2)	0.063 ± 0.003	0.7859	0.1798	6	$\approx 45min$
AGAN(3)	0.061 ± 0.006	0.8521	0.1662	6	$\approx 45min$

We also compare the images generated by ASinGAN(2) when the number of downsampling are 4, 5, 6, 7 and 8. The results are shown in **Fig. 7**. In the figure, the first column, the second column, the third column, the fourth column and the fifth column show the generated images with the number of downsampling of 4, 5, 6, 7 and 8 respectively. We can see that when N = 4 and N = 5, the content of the generated image is chaotic. When N = 6, N = 7 and N = 8, the content of the generated image has a better structure. But with the increase of the number of downsampling, the diversity of the image is decreasing. Therefore, when N = 6, the model has the best effect.

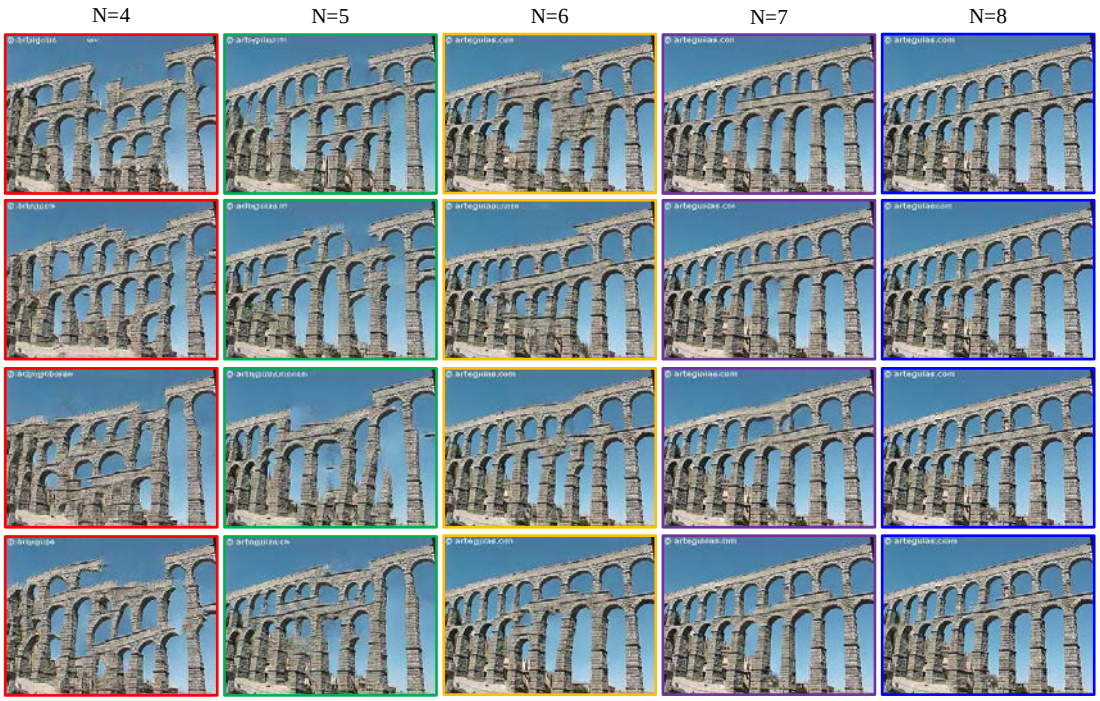

Fig. 7. The generated images by ASinGAN(2) when the number of downsampling are 4, 5, 6, 7 and 8

4.2 Random samples with arbitrary size

In this section, we test the performance of random samples generation to generate arbitrary size image. The generative network is fully convolution, so we can change the size of the input noise and generate an image with the arbitrary size. The two training images marked with red box and their corresponding generated images by AGAN with arbitrary size are shown in **Fig. 8**. We can see that the output image can well retain the overall information of the original image. In addition, the details of the original image are retained.

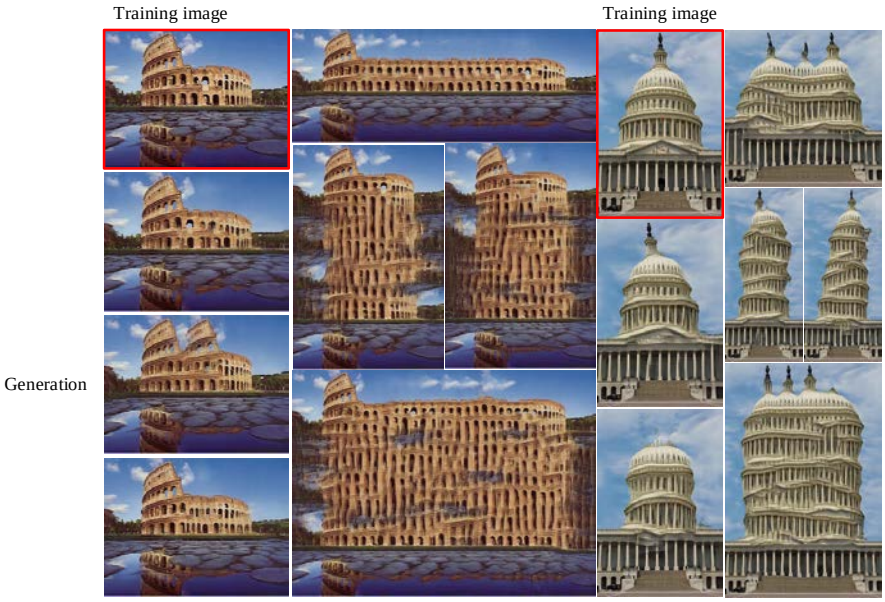

Fig. 8. Training images and generated images by AGAN with arbitrary size

To compare the performance of different methods for generating different size image, we adjust the size of input noise to make that the size of generated image is 1.5 times of the training image. The training images use the two images in the red box in **Fig. 8**. The training images and the images generated by SinGAN, ConGAN and AGAN methods are shown in **Fig. 9** and **Fig. 10**. We can see that the images generated by sinGAN lose the global features, and the generated images are chaotic. The images generated by ConSinGAN and AGAN can retain the global features well, which is obviously better than SinGAN. However, in the generated images of ConSinGAN, the background image often appears in the building, which affects the final result. Comparing with SinGAN and ConSinGAN, the images generated by our proposed methods are more natural. In **Fig. 9**, AGAN(2) gets the best results, followed by AGAN(1) and AGAN(3). In **Fig. 10**, AGAN(3) gets the best results, followed by AGAN(2) and AGAN(1).

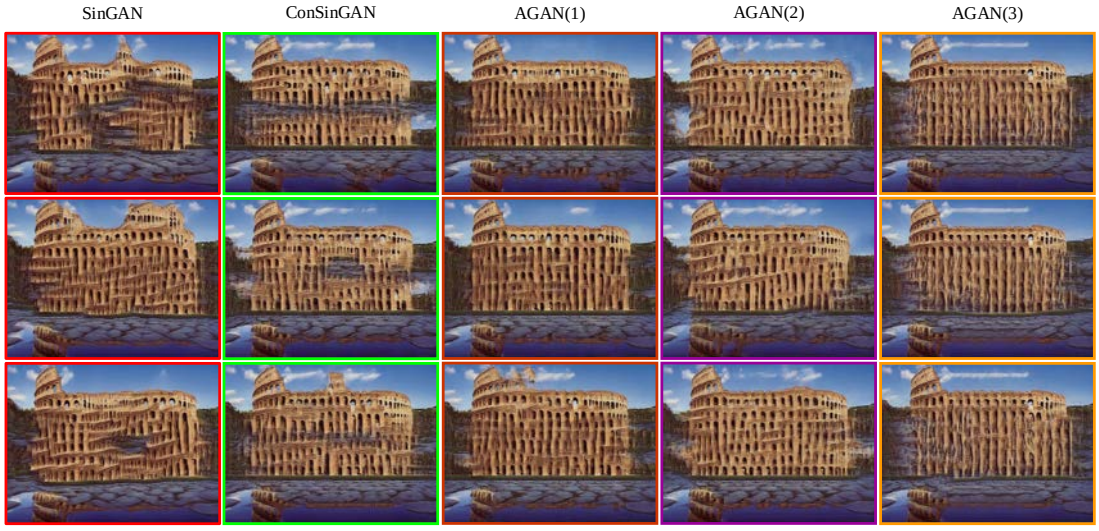

Fig. 9. Training image (colosseum) and generated images with 1.5 times the size of training images

4.3 Harmonization and Editing

Harmonization is usually used to merge different style images into the same style images. Editing merges a certain part of the image into the original image and makes it more coordinated to increase the diversity of the image. It is usually used for image enhancement to expand the data set. We firstly train the generative model to generate the original image, and the network only needs to input noise to generate the original image. Then we use the PS tool to paste the part that needs to be fused onto the original image to get a rough image and we call it as 'naive image'. Finally, the naive image is used as input image of generator to get the same style image with original image. The original images, naive images and generated images are shown in **Fig. 11**. In the first row of **Fig. 11**, cyclist image is pasted to original image. In the generated image of DPH [25], we can recognize the front wheel of the bicycle and upper body of the cyclist. In the generated image of SinGAN, cyclist and bicycle image are blurred. In the generated image of ConSinGAN, the whole body of the cyclist is clearer than DPH generated image. In the generated images of AGAN(1) and AGAN(2), the whole body of the cyclist and bicycle are clearer than the images generated by other methods. AGAN (3) has slightly worse results than ConSinGAN, but stronger than other methods. In the second row of **Fig. 11**, clock image is pasted to original image. In the generated image of DPH, the contour of clock is clear, but the numbers on the clock are mixed and cannot be recognized. In the image generated by SinGAN, the contour of clock is not clearer than that of DPH, but the numbers on the clock is clearer than that of DPH. Although the whole clock image generated by ConSinGAN is clearer than clock images generated by DPH and SinGAN, the clock image generated by AGAN(1) has clearer contour and number than the clock image generated by other methods, followed by AGAN(2). AGAN (3) has slightly worse results than ConSinGAN, but stronger than other methods. In the third row of **Fig. 11**, it also pastes a clock image to original image. The clock images generated by DPH and SinGAN are blurred. The clock images generated by AGAN methods are the clearest, followed by the generated image of ConSinGAN. The AGAN method retains more information of the pasted object than other methods. The AGAN can smooth the connection between the pasted object and the background image, and makes the fused image more natural.

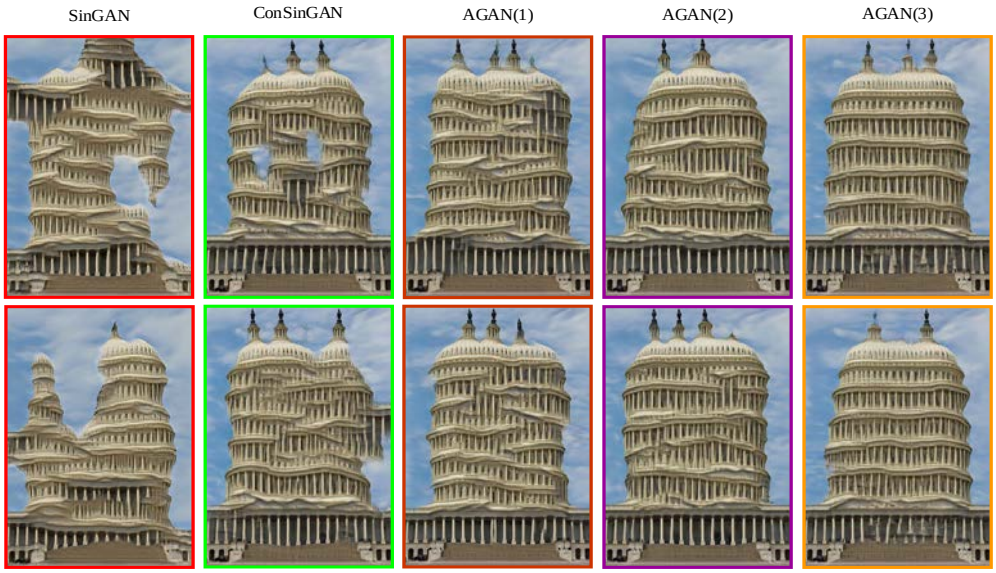

Fig. 10. Training image (capitol building) and generated images with 1.5 times the size of training images

In order to quantitatively compare the results of each method, we calculated the SIFID between the generated image and naive image for each method. The results are shown in **Table 3**. The SIFID values are 0.108, 0.113, 0.091, 0.087, 0.078 and 0.082 for DPH, SinGAN, ConSinGAN, AGAN(1), AGAN(2) and AGAN(3), respectively. The AGAN methods have the smallest SIFID, followed by ConSinGAN. In order to further verify the effectiveness of our model, we also conducted a user preference study. A total of 50 people participated in the user study. We made twenty groups of data, each group of data is composed of images generated by each method. Each person selects the most satisfactory image from each group of data, and finally calculates the average satisfaction of each method. The UPS (user preference study) values are 3.4%, 5.6%, 8.8%, 25.4%, 30.2% and 26.6%. These can also show that our method has the most real, natural and excellent effect.

Table 3. Comparison of SIFID and UPS in the aspect of harmonization

Model	DPH	SinGAN	ConSinGAN	AGAN(1)	AGAN(2)	AGAN(3)
SIFID	0.108	0.113	0.091	0.087	0.078	0.082
UPS	3.4%	5.6%	8.8%	25.4%	30.2%	26.6%

Fig. 11. Performance comparison of different methods in the aspect of harmonization

We also compare the editing performance of SinGAN, ConSinGAN, DPH and AGAN. We cut a part of original image and paste the cut image to different location of the original image, to generate the naive image. We use the generated naive image as input image of network model to generate more natural image. In the first row of **Fig. 12**, we cut three parts of original image to generate the naive image. The tree parts are marked in red number. Two parts are stone image, and one part is the sky image. From the images generated by different methods in the first row, we can see that there are obvious shadows around the part 1 and part 2 in the images generated by DPH and ConSinGAN method, which makes the generated images be unnatural. The images generated by SinGAN and AGAN are clearer and more natural. Among them, the differences between AGAN(1), AGAN(2) and AGAN(3) are small. In the second row of **Fig. 12**, we can see that there are obvious shadows and fuzzy edges around the stone hole in the images generated by DPH and ConSinGAN method. The images generated by SinGAN and AGAN have better coordination and consistency. So, they look clearer and more

natural. In the third row of **Fig. 12**, the shape of pasted object is square in the images generated by DPH, SinGAN and ConSinGAN. The shape of pasted object is close to circular in the images generated by AGAN. Therefore, the images generated by AGAN have a smoother effect on the pasted connection, which makes them look more natural and more integrated. It also can be seen that the images generated by AGAN look more like a natural image in the fourth and fifth rows than images generated by other methods. In other words, our proposed method can make the pasting part fully integrate into the original image. Based on above analysis, the AGAN method has better performance than other methods in the aspect of editing.

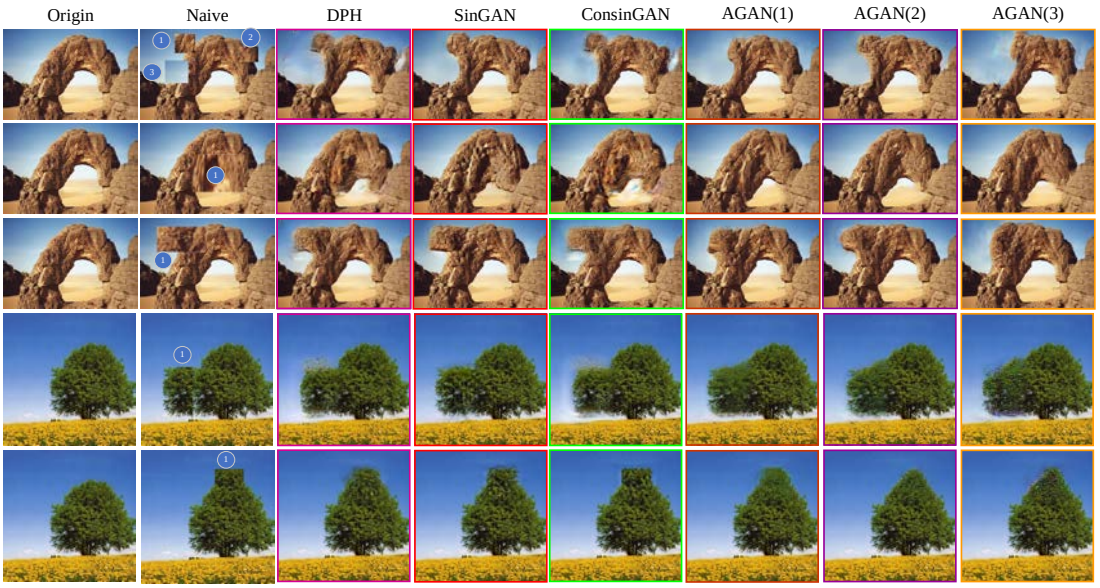

Fig. 12. Performance comparison of different methods in the aspect of editing

We also evaluated each method quantitatively in the aspect of editing. The results are shown in **Table 4**. The SIFID values are 0.097, 0.088, 0.108, 0.072, 0.081 and 0.077. The UPS values are 4.2%, 5.6%, 2.8%, 31.2%, 26.4% and 29.8%. These also show the excellent performance of our method.

Table 4. Comparison of SIFID and UPS in the aspect of editing

Model	DPH	SinGAN	ConSinGAN	AGAN(1)	AGAN(2)	AGAN(3)
SIFID	0.097	0.088	0.108	0.072	0.081	0.077
UPS	4.2%	5.6%	2.8%	31.2%	26.4%	29.8%

5. Conclusion

This paper proposes a new generative adversarial network on the basis of SinGAN method. We propose a downsampling strategy to increase proportion of low-resolution images, to obtain more global information. To increase the weight of effective image information, a new generative network is designed by introducing attention mechanism. We also propose a new loss function by adding TVLoss function to original loss function, to smooth the generated image. Firstly, to test the performances of different methods in the generating random samples with fixed sizes, two simple images and one complex image are used as training image, respectively. The images generated by our proposed method are closer to training images, and

have the smaller SIFID than the SinGAN and ConSinGAN methods. The training time of our proposed method is also smaller than SinGAN method. Secondly, to test the performances of different methods in the generating random samples with arbitrary sizes, colosseum image and capitol building image are used as training image, respectively. The images generated by our proposed method retain more information of the training image and natural, than the SinGAN and ConSinGAN method. Thirdly, we test the performance of different methods in harmonization. The images generated by our method can smooth the connection between the pasted object and the background image, and makes the pasted image clearer than SinGAN, ConSinGAN and DPH methods. In the end, we also test the performance of editing images of our method are more natural than SinGAN, ConSinGAN and DPH methods.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (61271115), Research Foundation of Education Bureau of Jilin Province (JJKH20210095KJ), Doctoral Scientific Research Foundation of Beihua University (20171424), Scientific Research Foundation of Jilin City (20190104124).

References

[1] C. G. Turhan and H. S. Bilge, "Recent Trends in Deep Generative Models: a Review," in *Proc. of International Conference on Computer Science and Engineering (UBMK)*, pp. 574-579, 2018. Article (CrossRef Link)

[2] G. Bombuwala and G. Poravi, "A Review of Generative Image Modeling Techniques," in *Proc. of IEEE 5th International Conference for Convergence in Technology (I2CT)*, pp. 1-5, 2019. Article (CrossRef Link)

[3] Goodfellow I J, Pouget-Abadie J, Mirza M, et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672-2680, 2014.

[4] Li Y, Wang H, and Dong X, "The Denoising of Desert Seismic Data Based on Cycle-GAN With Unpaired Data Training," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, pp. 2016-2020, 2020. Article (CrossRef Link)

[5] Yi X, Walia E, and Babyn P, "Generative adversarial network in medical imaging: A review," *Medical image analysis*, vol. 58, 2019. Article (CrossRef Link)

[6] Ledig C, Theis L, Huszár F, et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 105-114, 2017. Article (CrossRef Link)

[7] Shaham T R, Dekel T, and Michaeli T, "Singan: Learning a generative model from a single natural image," in *Proc. of IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4569-4579, 2019. Article (CrossRef Link)

[8] Mao X, Li Q, Xie H, et al., "Least Squares Generative Adversarial Networks," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 2813-2821, 2017. Article (CrossRef Link)

[9] Arjovsky M, Chintala S, Bottou L, "Wasserstein generative adversarial networks," in *Proc. of the 34th International Conference on Machine Learning, PMLR 70*, pp. 214-223, 2017. Article (CrossRef Link)

[10] Gulrajani I, Ahmed F, Arjovsky M, et al., "Improved training of wasserstein GANs," in *Proc. of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, pp. 5769-5779, Dec. 2017. Article (CrossRef Link)

[11] Radford A, Metz L, Chintala S, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. Article (CrossRef Link)

[12] Mirza M, Osindero S, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. Article (CrossRef Link)
[13] Denton E, Chintala S, Szlam A, et al., "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. of the 28th International Conference on Neural Information Processing Systems (NIPS'15)*, vol.1, pp. 1486-1494, 2015. Article (CrossRef Link)
[14] Karras T, Laine S, Aila T, "A Style-Based Generator Architecture for Generative Adversarial Networks," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396-4405, 2019. Article (CrossRef Link)
[15] Xin R, Zhang J, Shao Y, "Complex network classification with convolutional neural network," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 447-457, Aug. 2020. Article (CrossRef Link)
[16] Tilon S, Nex F, Kerle N, et al., "Post-Disaster Building Damage Detection from Earth Observation Imagery Using Unsupervised and Transferable Anomaly Detecting Generative Adversarial Networks," *Remote Sensing*, vol. 12, p. 4193, 2020. Article (CrossRef Link)
[17] Dong Y, Liu Y, Zhang H, et al., "FD-GAN: Generative Adversarial Networks with Fusion-Discriminator for Single Image Dehazing," in *Proc. of the AAAI Conference on Artificial Intelligence*, pp. 10729-10736, 2020. Article (CrossRef Link)
[18] Wu B, Liu L, Dai Z, et al, "Detecting Malicious Social Robots with Generative Adversarial Networks," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 11, pp. 5594-5515, 2019. Article (CrossRef Link)
[19] Li C, Wand M, "Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks," *Lecture Notes in Computer Science*, vol. 9907, pp. 702-716, 2016. Article (CrossRef Link)
[20] Jetchev N, Bergmann U, Vollgraf R, "Texture synthesis with spatial generative adversarial networks," *arXiv preprint arXiv:1611.08207*, 2016. Article (CrossRef Link)
[21] Bergmann U, Jetchev N, Vollgraf R, "Learning texture manifolds with the Periodic Spatial GAN," in *Proc. of the 34th International Conference on Machine Learning, PMLR*, vol. 70, pp. 469-477, 2017. Article (CrossRef Link)
[22] Zhou Y, Zhu Z, Bai X, et al., "Non-stationary texture synthesis by adversarial expansion," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1-13, 2018. Article (CrossRef Link)
[23] Shocher A, Bagon S, Isola P, et al., "InGAN: Capturing and Remapping the "DNA" of a Natural Image," *arXiv preprint arXiv:1812.00231*, 2018. Article (CrossRef Link)
[24] Hinz T, Fisher M, Wang O, et al., "Improved Techniques for Training Single-Image GANs," in *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1300-1309, 2021. Article (CrossRef Link)
[25] Luan F, Paris S, Shechtman E, et al., "Deep painterly harmonization," *Computer graphics forum*, vol. 37, no. 4, pp. 95-106, 2018. Article (CrossRef Link)

Liquan Zhao was born in Heilongjiang province in 1982. He received the B.S degree in Electrical & Information Engineering from Harbin University of Science and Technology, Harbin, China, in 2005 and the Ph.D. degree in Communication and Information System at Harbin Engineering University, Harbin, China, in 2009. From 2009, he was an associate professor at Northeast Electric Power University, Jilin, China. His research interests include deep learing and blind source separation

Yupeng Zhang was born in Jilin province in 1997. He received the bachelor's degree in electronic information engineering from the Northeast Electric Power University, Jilin, China, in 2019. He is working toward themaster's degree in School of Electrical Engineering, Northeast Electric Power University, Jilin, China. His research interests include deep learning and Generative Adversarial Networks