Image Synthesis and Editing with Generative Adversarial Networks (GANs): A Review.

Wanwan Li

George Mason University, Fairfax VA 22030, USA

Abstract. Recently, as many deep learning models are emerging, deep learning has achieved great success in the field of artificial intelligence(AI). Especially, the Generative adversarial networks (GANs) based on zero-sum game theory has become a new research hot spot in the field of deep learning. The significance of the GAN model is that it can generate realistic data through unsupervised learning. Based on the conceptual and theoretical framework of the generative adversarial network, GANs models and their application result in tremendous success among different areas, especially in image synthesis and editing. This paper visualizes the data structures of various kinds of GANs models in 3D and discusses the variational GAN models with respect to their improvements in the applications. As the GANs have superior learning ability, strong plasticity, great potential for improvement, and a wide application range, this paper prospects the possible applications of the GANs in the near future.

1 Introduction

Generative adversarial networks (GANs) have become a hot research direction in the field of artificial intelligence(AI). The basic idea of GANs is derived from the two-person zero-sum game in game theory. It consists of two different neural networks: a generator and a discriminator. Typically, it is trained by means of adversarial unsupervised learning. The purpose of GAN is to estimate the potential distribution of data samples and generate new data samples. In the fields of image processing and visual computing, speech and natural language processing, information security, chess games, etc, GANs have been widely studied and got tremendous success in different areas. This paper firstly introduces what is GAN, illustrate its structure with 3D visualizations, and discusses the advantages and disadvantages of the original GAN model. Subsequently, this paper introduces some derivative models of GANs, their new features, and their improvements compared with the original model. Finally, this paper summarizes the application fields of GANs, the performance of existing models, some representative works of GAN in image synthesis and editing area, and prospect the possible applications and extensions of GANs in the near future.

1.1 Introduction to Generative Adversarial Networks(GANs)

Generative adversarial networks (GAN) is a generative model originally proposed by [6]. GAN is one kind of structured learning, it is inspired by the two-person zero-sum game in game theory (i.e. the sum of two-person interests is zero, and the gain of one side is the loss of the other side). Generally, the GAN system consists of a generator and a discriminator(See Figure 1). The generator captures the potential distribution of real data samples and generates new data samples, namely, fake data. The discriminator is a classifier to distinguish whether the input is real data or fake data.

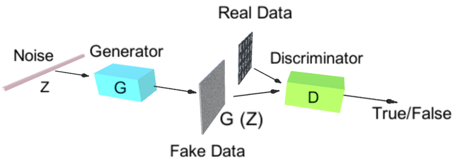


Fig. 1. The structure of GAN.

As shown in the Figure 1, the task for the generator G is to train a neural network that is able to convert arbitrary randomly distributed noise, typically called as a latent vector Z, into a synthesized fake data $G(Z)$, and try to make the fake data approach real data x, that is, the training data, as much as possible. At the same time, the discriminator is trained simultaneously as a classifier. Ideally, the data generated by the generator is classified as fake data while the data from the training set as the real data. Therefore, the discriminator is trained to be good at deciding whether the data is real or fake.

In the beginning, the generator generates fake data randomly, therefore, it is easy for the discriminator to identify the fake data. While the generator is optimizing and improving, more and more fake data looks like real data, until the discriminator can't tell which one is real and which one is fake. That means, in the end, the generator can generate realistic data being able to "fool" the discriminator. Therefore, the optimization is done through a min-max loss function, that the "worse" performance of the discriminator results in the "best" performance of the generator. The loss function for discriminator $L_D(D, G)$ is defined as:

$$\min_G \max_D L_D(D, G) = \log D(x)|_{x \sim \text{data}} + \log(1 - D(G(z))|_{z \sim \text{noise}} \tag{1}$$

where data x is from the real data distribution while latent vector z is from the random distribution (noise). The discriminator D wants to maximize the classification between the distribution from real data $x \sim$ data and the one from the fake data $x \sim G(z)$. After the optimizations, we hope that $D(x)$ increases which means the discriminator can identify real data more accurately. That is why we maximize D. At the same time, the discriminator hopes $D(G(z))$ decreases, as it wants to identify the fake data more accurately. Therefore, we minimize G. On the other side, as a generator G, we hope that $D(G(z))$ increases which means the fake data $G(Z)$ looks like real data after the optimizations. Therefore, the goal of the generator is to maximize the loss function for generator $L_G(D, G)$:

$$\max_G L_G(D, G) = \log(D(G(z))|_{z \sim \text{noise}} \tag{2}$$

where similarly, z is a latent vector from a random distribution (noise).

Generator and discriminator can be implemented through deep neural networks which are proved to be robust enough to achieve the goals through a series of experiments. As the trained model is able to generate fake data that conform to the sample distribution of real data without any prior knowledge, therefore, generative models take an important role in unsupervised deep learning to capture the high-order correlation of data without target class label information. By learning the semantic features of the real data, the GAN model can estimate the distribution of training data and generate new data similar to training data.

1.2 Advantages and Disadvantages of Original GAN

Before the appearance of the original GAN model, there are some existing unsupervised learning-based generative models proved to be efficient for image synthesis. For example, Autoencoder (AE) proposed by [9] is able to convert an input image into a code layer through a neural network (Encoder) and convert such code layer through another neural network (Decoder) back to an image as similar as possible to the original input. During the training process, both encoder and decoder are trained simultaneous and the difference between the input images and the decoded images are backpropagated to the optimizer as the loss function to be minimized. AE and its variational version called Variational Auto-Encoder(VAE)[11] have been successfully applied to image reconstruction and content-based image retrieval[12]. AE overcomes the traditional bottleneck of pixel-based image retrieval approaches by directly comparing the code generated from the encoder instead of comparing the images pixel-by-pixel. Through this approach, AE is able to catch the semantic features of the retrieved images while pixel-by-pixel-based approaches cannot. Another application of AE is image denoising [25]. Before sending the original high-quality input image to the AE, the input image is preprocessed by adding white noises. Then train the AE to generate images as close to the original high-quality input images as possible. Through such a training process, AE is able to reconstruct high-quality images from low-quality noised images input. As another representative generative model, PixelRNN[21] trains the neural network by estimating the conditional distribution of each individual pixel in a given adjacent pixel (left or upper). In the PixelRNN model, the input of the current pixel and its adjacent previous pixel are both sent into the recurrent neural network as a sequence, the previous pixel is the condition of the current pixel input. Through such a mechanism, the PixelRNN is able to predict another part of an image from one part of the image. Therefore, PixelRNN is proved to be a powerful tool to deal with image completion problems.

Although the existing generative models (such as AE, VAE, etc.) are able to generate new images by itself, the output images are seriously restricted to the training dataset as they are trying to mimic the input image from the training set. Therefore, the synthesized data are very similar to the original input from the existing dataset, and sometimes, such models even directly retrieve images from the dataset. On the other side, PixelRNN is trying to complete the images without considering the global image semantics and trying to combine different images from different catalogs as new images that are semantically meaningless. For example, a car image is completed with a cat face. Therefore, a more intelligent generate model needs to have two advantages: 1) the generated images are semantically meaningful and can be identified as realistic images. 2) the generated images are not retrieved from the dataset and not too similar to the given training samples. With such two advantages, the original GAN model made itself a breakthrough in the state-of-art of machine learning. On one hand, as the GAN has a discriminator to identify the image as real or fake, therefore, if well-trained, GAN is theoretically able to synthesize realistic images, this explains the first advantage. On the other hand, as the generator in the GAN has no input directly from the dataset, therefore, the generator doesn't know how does the realistic data look like and there is no chance for the generator to generate new realistic images by copying the existing images, this explains the second advantage.

However, the original GAN model, inspired by the Nash equilibrium in game theory, still has some inevitable disadvantages. For example, there is no explicit function to evaluate how good the current status of the GAN is. Typically, loss functions can be used to evaluate whether the current neural network is well-trained. If the loss function is close to zero, then the neural network looks "good" enough. But in the original GAN model, the generator hopes to maximize the discriminator's loss function, which means it hopes to fail the discriminator by "fooling" it "smartly". But on the other hand, the discriminator wants to be good at punishing the generator by minimizing its own loss function while maximizing the generator's loss function. Only in that way, the discriminator can be trained to identify the fake data well. Therefore, as a consequence, the optimization will not be guaranteed to converge as their loss functions are adversarial. Also, it is not guaranteed whether the discriminator or the generator is over-trained and it is very hard to be balanced. In the end, there will be either harmonious consistency between discriminator and generator or there is one side over-performed than another side. For example, when the generator is trying to "fool" the discriminator in some "tricky" ways such as only generate some specific images that can be identified as real images. In that case, the discriminator is not well-trained and the generator will not be able to generate realistic images diversely. For solving such instability, oscillations and divergences issues that potentially exist during the training process of the original GAN model, some variations of the GAN models change the original JS-divergence[4] into other types of divergence such as the f-divergence in the f-GAN[19] or change the distance measure function into Wasserstein distance in the WGAN[1].

2 Variations of Generative Adversarial Networks

The Generative Adversarial Networks (GANs) was proposed in 2014 and has achieved great influence on the machine learning community. However, the original GAN model is unpractical for most of the challenging real-world tasks. Therefore, different variations of GANs models were proposed. For example, Conditional Generative Adversarial Networks (CGAN) [18], Deep Convolutional Generative Adversarial Networks (DCGAN)[24], Information Maximization Generative Adversarial Networks (InfoGAN)[2], Wasserstein Distance-based Generative Adversarial Network (WGAN)[1], etc. Most of these GAN models are milestones that push the original GAN model a great step forward as a powerful tool to solve the real-world problem. In this section, some milestones of the variations of the GAN models will be introduced.

2.1 DCGAN

DCGAN, short for deep convolutional GAN, firstly proposed by [24], is a milestone in the development of improved GANs model. It combines CNN in supervised learning and the GAN in unsupervised learning, which is robust and convenient for engineering implementation (See Figure 2). In order to enhance the original GAN model, the DCGAN model removes the fully connected hidden layers to construct a deeper neural network. Details of the implementation such as scale the synthesized image with the Tanh function, add batch normalization layer, all parameters initialization is randomly obtained from the normal distribution, add the ReLU activation function in the generator, and add the Leaky ReLU activation function in the discriminator, etc. Besides, DCGAN used Adam optimizer with a learning rate = 0.0002.

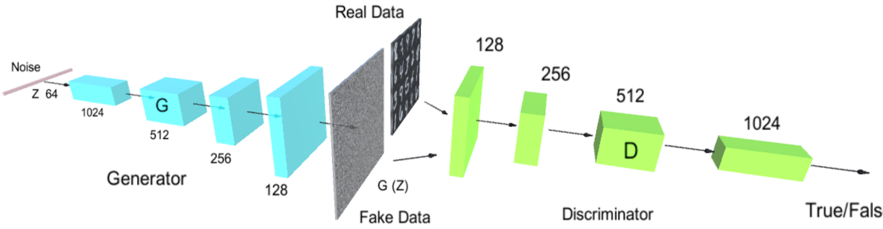


Fig. 2. The structure of DCGAN.

The contributions of the DCGAN model include (1) Batch normalization (BN): BN is used both in the generator and the discriminator to alleviate the training collapse problem and effectively avoids oscillation and instability of the model. (2) The output layer of the generator uses the Tanh activation function, which is appropriate for the adam optimizer and results in more satisfying results[22]. (3) Replacing the pooling layer with stride convolutional networks in discriminator and with fractional-stride convolutions in the generator, which is a more suitable sampling kernel function for unsupervised learning. Because of the improved training stability of the DCGAN model, it has been widely accepted and used for academic purposes.

2.2 WGAN

Although DCGAN tries to use multi-layer CNN to solve the problem of training collapse in engineering, theoretically, instability of training the original GAN model still exists. As the original GAN model plays a min-max game, which wants the discriminator to separate the real data from fake data as much as possible. However, during the training, if the discriminator is over-trained and very "alert" to the fake data, even the fake data are getting more and more real, it will still be "punished" as fake data. Therefore, the gradient towards moving to real will be unstable so that there is no suitable fitting function for fast convergence of loss function, in the end, the fitting process in the training process is shaking seriously with large gradients and not promised to be converging[20]. As a great step in improving the original GAN model, WGAN, short for Wasserstein GAN, propose by [1], devised a Wasserstein distance-based optimization approach to theoretically solve such problem.

Wasserstein distance, also known as earth mover distance, between two distributions, is defined by a metric measured by the minimum steps left for moving one distribution towards another. However, accurate calculation of the Wasserstein distance requires an optimizations approach which increases the computational time complexity of problem-solving. Therefore, according to the proof that if the derivatives of the discriminators D are smoother enough through approximating a Lipschitz function, it is equivalent to solve a Wasserstein distance as the measure of two distributions. Therefore, in the WGAN model, the loss function for discriminator $L_D(D, G)$ is defined as:

$$\max_{D \in 1-\text{Lipschitz}} L_D(D, G) = \log D(x)|_{x\sim\text{data}} - \log D(G(z))|_{z\sim\text{noise}} \tag{3}$$

where z is a random latent vector. $1-$Lipschitz functions are the functions whose disturbance of input is always greater or equal to the disturbance of their output. According to the existing techniques, $1-$Lipschitz function is implemented through two different methods: (1) weight clipping method (WGAN-WC) and (2) gradient penalty method (WGAN-GP). WGAN-WC method enforces the weights clipped in a pre-assigned range. And the WGAN-GP method, also known as Improved WGAN, proposed by [7], penalizes the gradient with a soft regularization term if the length of the gradient is larger than 1.

The contribution of WGAN theory lies in: (1) Defining a smoother loss function that can avoid the over-large step gradient problem; (2) Solving the problem of GAN training instability in an innovative way, make it unnecessary to carefully balance the training of generator and discriminator, (3) Theoretically solving the problem of model collapse and ensuring the diversity of generated samples [8]. (4) The Wasserstein-based method provides a better mathematical definition of the distance between the real data distribution and the fake data distribution[28].

2.3 CGAN

Inspired by the conditional probability, feeding the GANs with additional conditions results in a novel variation of the GAN, namely, Conditional GAN (CGAN), which is proposed by [18]. Generally, we separate unsupervised learning from supervised learning by judging whether the neural network can extracting rules from unlabeled data. Through the CGAN, the generated model is trained by the joint probability distribution of the training data samples and user-specified additional conditions (such as its labels), it successfully extends the original GAN model from unsupervised learning into supervised learning. In order to solve the problem of how to generate a model given specific requirements, the improved method feeds such conditions into both of the discriminator and the generator networks (See Figure 3). For example, if the user specifies the CGAN to generate the handwritings for number 3, then the networks of CGAN can be trained with the prior knowledge of classification information (namely, the labels of 1, 2, 3, ..., 10, etc.) together with the image pixels information. During the training, the generator takes both the label and the latent noise vector z as the input, and pass the label to the discriminator. For discriminator, either the generator generates a poor-quality image or it generates an image not matching with its input label, it will be detected as fake data. In such a way, the discriminator is able to improve both the quality of the generated image and the correlations between the generated image and its label through the feedback for the generator. As shown in experimental results, CGAN improves the original GAN by adding controllable conditions into the model it is able to generate user-desired results given to their requirements.

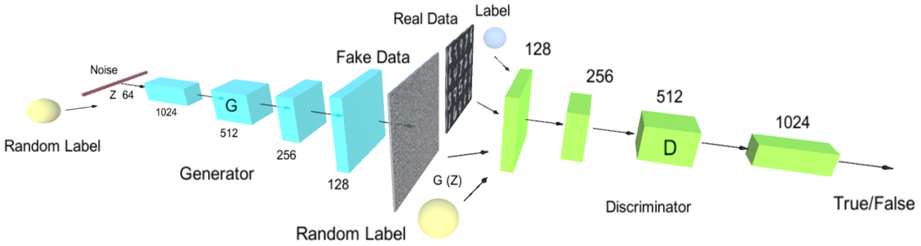


Fig. 3. The structure of CGAN.

2.4 InfoGAN

As proposed by [2], InfoGAN is the GAN extended with information maximizing theory. The output of the original GAN model is $G(z)$, where z is a totally unstructured random vector. During the training, there are no additional restrictions on z to generate data, therefore, within the neural network, there are complex relations and connections between z and $G(z)$, namely, output $G(z)$ is highly entangled with input z, and make the individual elements in vector z loss their correspondence to the specific semantic features of the data. InfoGAN has made improvements by adding a highly structures latent code c, changing the output to $G(z, c)$ and splitting the original input of the generator model into two parts: (1) the random noise z, and (2) a random latent code c which is used to targeting the semantic features of the data distribution[28]. (See Figure 4).

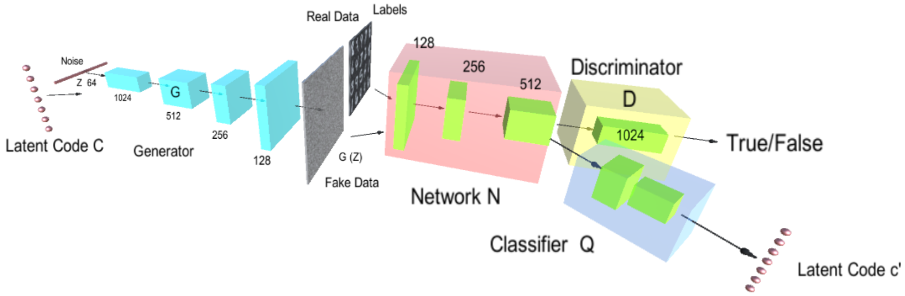


Fig. 4. The structure of InfoGAN.

According to the hypothesis from [2], there should be high mutual information between latent codes c and generator distribution $G(z, c)$, therefore, in order to maximize the mutual information between the c and $G(z, c)$, it incorporates an assistant classifier $Q(c, G(z))$ to classify the latent code c from the synthesized fake data $G(z)$. Therefore, only if c has connections to the salient features of data so that it can be classified correctly by the classifier Q. Put this in another way, InfoGAN can be generalized as an unsupervised version of conditional GAN as it plays a role as both an encoder and a decoder himself at the same time. For simplicity, some of the implementations of InfoGAN shares part of deep neural network N for both the classifier and discriminator. The discriminator has a boolean output true or false for identifying the real data from fake data, while the output latent code c' are classified through the classifier Q, which is used to be compared with the input latent code c fed into the generator G. If the classified output latent code c is closer to the original input latent code c, it will result in a higher score for the generator G as the generator is able to extract the semantic information from latent code better.

Specifically, when learning to generate images, images have many controllable meaningful dimensions. For example, In the MINIST dataset, the latent code c can be a continuous value to represent the thickness or rotation of the handwritten letters. Similarly, in the celebrity face dataset CelebA, c can be a variable to control the eye size, smiling degree, and hair length [8]. The rest parts of the latent code are not obvious to observe, or there doesn't exist any salient feature, therefore, they are kept as the random part of the latent vector

z. InfoGAN extends the original GAN model with the ability to synthesize more controllable data representation by tuning the values in c[20]. Besides, the user can also control the dimension of c so that the InfoGAN can adjust the generated images in a specific semantic dimension [22]. Using this latent code-based variable modeling mechanism, InfoGAN has taken another step forward in the development of GANs.

2.5 EBGAN

According to the original GAN model, in the beginning, the performance of the discriminator is poor and it is improved step by step as more and more real data and fake data are fed in. Therefore, the generator improved extremely slowly in the case that the discriminator is not well trained. However, EBGAN (Energy-based GAN) first proposed by [28] which extends the GANs with an auto-encoder/decoder which is pre-trained (See Figure 5). As the auto-encoder is pre-trained through the real data by minimizing the energy which measures the difference between the decoded images and the input images. Therefore, when the fake images are fed into the autoencoder, they will not be decoded correctly until the generated fake images look real.

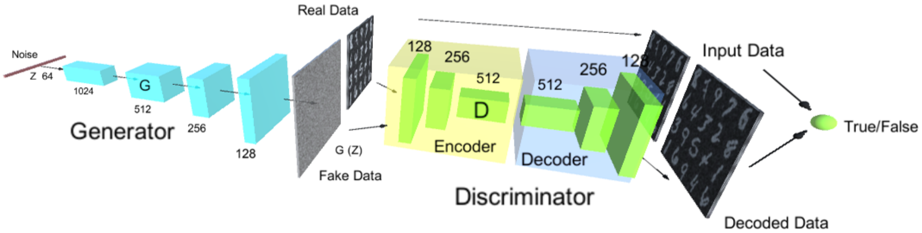


Fig. 5. The structure of EBGANs

By minimizing the energy value, extracted from the well-trained discriminator, which is smaller in the region near the real data domain but higher in the non-real data domain [20], the generator will be improved much faster at the beginning than the original GAN model. As shown in the success of the experiments, EBGAN gives GANs an explanation of the energy model, that is, the generator aims to produce the samples with the lowest energy according to the decoded data from the discriminator, while the discriminator aims to give the samples with higher energy to reject the fake data from the generator. From the point of view of the energy model, EBGANs extend the GANs structure with a more and wider range of generalizing the loss function in a novel way. For example, VAE-GANs proposed by [14], which uses variational autoencoders (VAE) as discriminators within the GANs framework and some other variations of EBGANs take a further step along with this research direction.

3 Application of GANs

As a powerful tool to generate realistic data, GANs are widely used in different areas. The most general application of GANs is to synthesize a fake image that looks like a real one and editing the existing image into a synthesized image without distortion. According to the strengths from diverse variations of GANs, GANs has been widely used among different areas to solve real-world tasks and problems. In this section, some of the recent representative works of GANs are selected and introduced.

3.1 Image Synthesis

Image synthesis is the most general application of the variations of GAN models. There are several famous image datasets (See Figure 6) serving as benchmarks for evaluating the efficiency of a GAN model. For example, the MNIST database of handwritten digits proposed by [15], CelebA dataset of a large-scale celebrity faces attributes proposed by [17], and Fashion-MNIST proposed by [26].

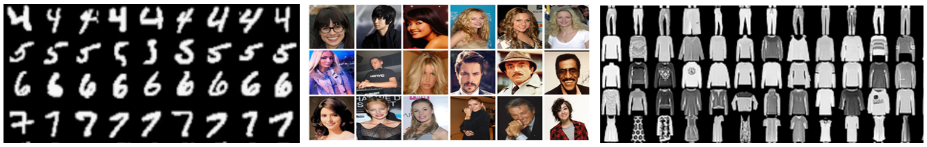
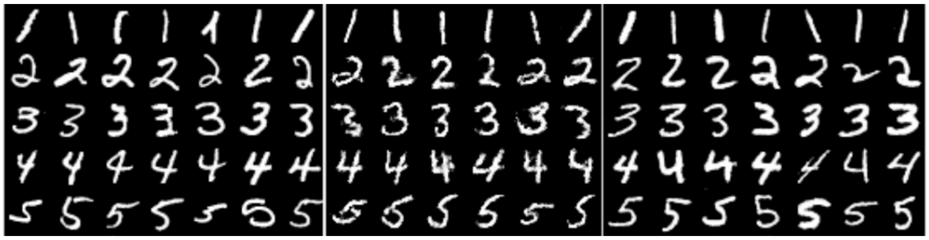

(a) MNIST samples (b) CelebA samples (c) Fashion-MNIST samples

Fig. 6. This figure displays the sample data from there benchmark database (a) samples from the MNIST database[15], (b) samples from the CelebA database [17], and (c) samples from the Fashion-MNIST database [26].

Different variational GAN models demonstrate the improvements in their performances on the benchmark database. [24] devised DCGAN to generate different ranges of images including bedroom and face images, and prove that the synthesized images are not retrieved from the existing image database. Its performance compared with the ground truth of MNIST and the original GAN are shown in Figure 7. As shown in Figure 8, WGANs demonstrate their capability in generating images with high-qualities. Beyond this baseline, CGAN is moving a step forward which is able to generate the samples with specific conditions as shown in Figure 9. As shown in Figure 10, InfoGAN is able to extract the semantic information from the sample images and find the relations between the latent code c and such semantics. This provides with users a more controllable interface to generate the desired images. Based on the performance of DCGAN, EBGAN makes further improvements in generating high-quality images. As shown in the LSUN bedroom dataset (See Figure 11), the left images are generated with the DCGAN, the right images are generated with the DCGAN, some noise can still be found in the DCGAN, but in EBGAN, the noises are less obvious.



(a) MNIST (b) GAN (c) DCGAN

Fig. 7. Performance of DCGAN: (a) samples from the MNIST dataset, generations from the original GAN, and generations from the DCGAN. Image is cited from [24].

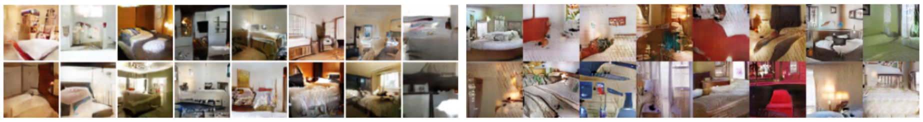


Fig. 8. Images synthesis from the WGAN generator. Left: WGAN model. Right: original GAN model. Given the bedroom figure dataset, both models produce high-quality samples. Image is cited from [1].

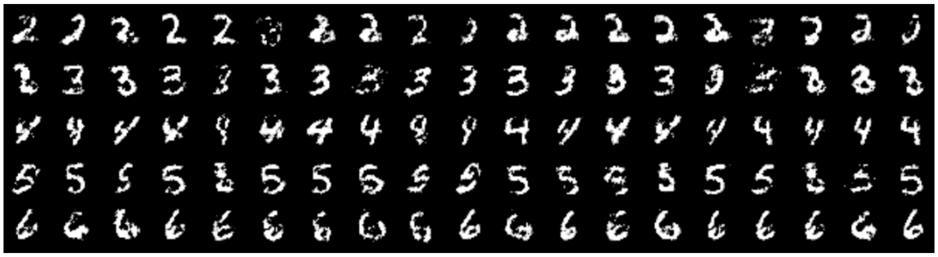
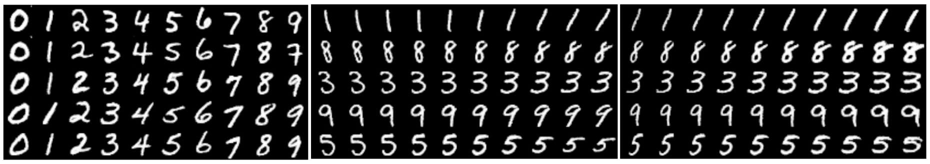
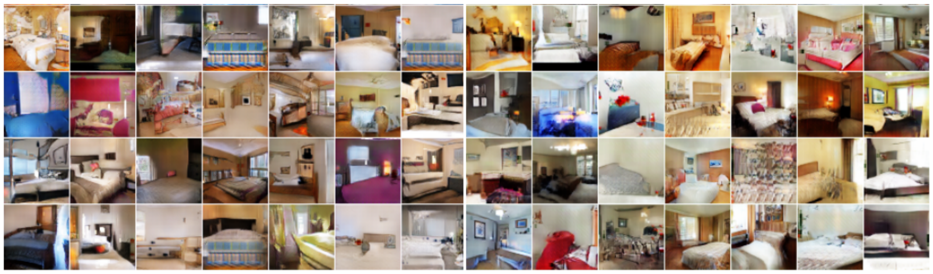


Fig. 9. Generated MNIST digits with CGAN, rows (from top to bottom) represents conditioned labels from "2", "3", "4", "5", to "6". Image is cited from [18].



(a) Digit type(c_1) (b) Rotation(c_2) (c) Width(c_3)

Fig. 10. InfoGANis able to correspond the latent code c with the image semantics: In (a), varying c_1 corresponds to different digit types; in (b), a small value of c_2 denotes left-leaning digit whereas a high value corresponds to right-leaning digit; in (c), c_3 smoothly controls the width of the strokes. Image is cited from [2].



(a) DCGAN (b) EBGAN

Fig. 11. This figure compares the generator performance between DCGAN (a) and EBGAN (b) Image is cited from [28].

3.2 Image Editing

As GAN models and technologies are getting mature, the extensions of the GANs are increasing rapidly, especially in the area of image editing. As traditional ways to edit the image, the popular image processing software, such as Photoshop developed by Abode company, is widely used among multimedia artists and designers. However, even the software is getting more powerful and provides more easy-to-use functions, there still need lots of manual efforts from the designers. Therefore, GANs' impressive synthesis ability attracts more and more researchers to develop image auto-editing tools through deep neural networks. Typically, image editing is divided into two types: local editing and global editing. The most famous application of deep learning on global editing is the image style transfer technique [5]. Most of the existing global editing algorithms are built on image-to-image translation networks and result in surprising results [23]. However, unsupervised image-to-image translation and image local editing remain challenging before the appearance of the GANs models. As local image editing techniques require more conditions than the global image editing (such as whether the stitching is natural, whether the edited part is consistent with the full image content, etc.), this challenges of the local image editing techniques provide a perfect stage for the GANs to demonstrate their specialty.

As the GANs model can repair and complete the missing area of the image according to its surrounding area from the semantic level, it provides more natural and acceptable results than the nearest neighbor stitching method. Typically, GANs work through the Context Encoder (CE) [27], which are widely used for high-resolution image patching under specific circumstances. CE includes two parts: Encoder and Decoder, which typically incorporate DCGAN applied with ADAM optimizer to achieve image auto-completing with highly satisfying quality. For example, [3] successfully change the status of the people's eyes in an elegant manner using the Exemplar GAN, a variation of CGAN. As shown in Figure 12, Exemplar GAN can achieve photo-realistic, high-quality, and personalized in-painting function to edit the image in an astonishing way. As another impressive work for image completion with high consistency between local image and global image, high image resolution, and extremely photo-realistic image content, [10] successfully achieved the real-time and interactive auto-completing of local image editing skills using GAN model (See Figure 13). At the same time, the face editing skills also improve. As shown in the Figure, [23] it successfully devised a conditional DCGAN-based approach to automatically edit the human face and hair image in a time-saving manner simply by sketching on the face image with simple curves. This is a great invention as it opens the eyes for people towards photo editing through sketching, the most convenient, and straightforward way to edit the images. Also, there are many other types of applications of the GANs that are not listed here, which will be briefly discussed in the discussion sections.

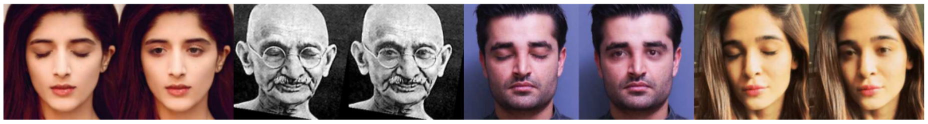


Fig. 12. Eye editing using Exemplar GAN. Image is cited from [3].

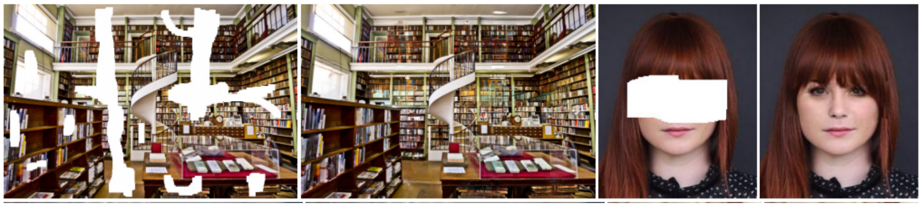


Fig. 13. Auto-image completion using GAN. Image is cited from [10].

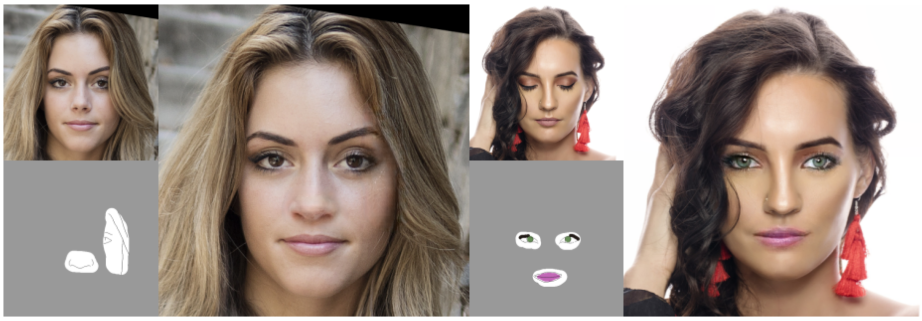


Fig. 14. Sketch-based interface for face image auto-editing using CDCGAN. Image is cited from [23].

4 Discussions and Conclusion

Besides image synthesis and editing, GANs has more various kinds of applications among different research areas, for example, GANs have deep influences on image enhancement, super-resolution. The same kind of image, such as face image enhancement from low-resolution face image to high-resolution image, can be better implemented by GANS. The idea is to use a low-resolution image as a constraint condition to generate the realistic high-resolution images[24]. As another example, the super-resolution energy adversarial network (SRGAN) [16], without depending on refined data sets, it can enhance and denoise various types of images. The features of SRGAN includes: (1) completing the missing part of images with the constraint on the global context of the generated image to ensure the smoothness of the result; (2) SRGAN feed the generated data and real data into the VGG-19 network respectively, define the loss items according to the difference of the feature map and adds normalization to the output[20]. Finally, after combining these loss items including mismatching loss, image smoothing loss, and feature map difference loss, the super-resolution image is generated by feeding into the GANs framework through minimizing the loss functions.

Besides, the GANs have been extended into the sketch restoration and colorization techniques. Sketch restoration refers to the technique that can convert the drawing of sketches into realistic color images. Its special case is portrait restoration. A DCGAN can complete the restoration of draft drawings [13]. it gives the training methods of two kinds of image databases with corresponding relations, finds the relationship between database images, and gives a general solution of sketch-to-image conversions, such as the conversion of the street scenes, the sketch of buildings, the terrain elevation from a contour map. Traditional image coloring methods do not use massive data, and image coloring can not be applied to all types of pictures. [28] use Patch GAN network to complete the picture coloring work, and make full use of massive data to achieve better results. This proves that GAN plays an active role in sketching techniques.

In conclusion, generative adversarial network(GANs) has become one of the most important and influencing methods in deep learning. It has the advantages of fully fitting data, faster synthesis speed, and realistic data generation. The academic research of the GANs model is progressing rapidly. The original GANs model is trained by MinMax optimization. The conditional generative adversarial network(CGAN) adds preconditions to input data in order to provide the controllability of the output. DCGAN, a deep convolution generation adversarial network, proposes a stable training network structure to prevent training collapse. InfoGAN controls semantic change through latent code and extracts the relations between the latent code and its corresponding semantic feature of the training data. EBGAN explains the adversarial network from the perspective of the energy model. WGAN defines a more smooth distance measure for the loss function and gives a better mathematical definition of the distance between real data distribution and fake data distribution, which theoretically solves the problem of training collapse and instability. As a powerful, robust, and reliable tool, the GANs model has been widely used in image synthesis, image editing image repair, image denoising, sketch restoration, sketch colorization, and other image processing area. At present, however, there is no quantitative standard for evaluating the realism of synthetic images, that is, it is hard to quantitatively measure how realistic an image is. It can be only judged subjectively whether the synthesized images look natural and realistic. This remains an open topic for the GANs community. In the near future, GANs will be applied to more general applications not only within the image processing domain. For example, music synthesis using GANs to generate different styles of music composed by different composers or even create innovative music styles in a ”masterpiece” level like famous musicians such as Beethoven and Mozart. The GANs can also be extended to write articles and poems with different styles and cultural backgrounds. When the GANs are getting more mature, even the GANs can take over human jobs such as movie directors who are directing new movies including story design and character artistic design, choreography artists who are designing new dance for a group of dancers and so forth. In one last word, GAN is going to change human lives from different aspects in the near future.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
2. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in neural information processing systems. pp. 2172–2180 (2016)
3. Dolhansky, B., Canton Ferrer, C.: Eye in-painting with exemplar generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7902–7911 (2018)
4. Fuglede, B., Topsoe, F.: Jensen-shannon divergence and hilbert space embedding. In: International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings. p. 31. IEEE (2004)
5. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
6. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
7. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. pp. 5767–5777 (2017)
8. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: ACM Transactions on Graphics (TOG). vol. 26, p. 4. ACM (2007)
9. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science **313**(5786), 504–507 (2006)
10. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (ToG) **36**(4), 1–14 (2017)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

12. Krizhevsky, A., Hinton, G.E.: Using very deep autoencoders for content-based image retrieval. In: ESANN. vol. 1, p. 2. Citeseer (2011)
13. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. arXiv preprint arXiv:1512.09300 (2015)
14. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: International conference on machine learning. pp. 1558–1566 (2016)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
16. Lian, S., Zhou, H., Sun, Y.: Fg-srgan: A feature-guided super-resolution generative adversarial network for unpaired image super-resolution. In: International Symposium on Neural Networks. pp. 151–161. Springer (2019)
17. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
18. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
19. Nowozin, S., Cseke, B., Tomioka, R.: f-gan: Training generative neural samplers using variational divergence minimization. In: Advances in neural information processing systems. pp. 271–279 (2016)
20. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2642–2651. JMLR. org (2017)
21. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759 (2016)
22. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2536–2544 (2016)
23. Portenier, T., Hu, Q., Szabo, A., Bigdeli, S.A., Favaro, P., Zwicker, M.: Faceshop: Deep sketch-based face image editing. arXiv preprint arXiv:1804.08972 (2018)
24. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
25. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. pp. 1096–1103 (2008)
26. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
27. Yelamarthi, S.K., Reddy, S.K., Mishra, A., Mittal, A.: A zero-shot framework for sketch based image retrieval. In: European Conference on Computer Vision. pp. 316–333. Springer (2018)
28. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016)