IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 6, 2018, accepted December 8, 2018, date of publication December 14, 2018, date of current version February 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2886814

Recent Advances of Generative Adversarial Networks in Computer Vision

YANG-JIE CAO[1], LI-LI JIA[1], YONG-XIA CHEN[1], NAN LIN[1], CONG YANG[1], BO ZHANG[1], ZHI LIU[2], (Member, IEEE), XUE-XIANG LI[1], AND HONG-HUA DAI[3], (Member, IEEE)

[1]School of Software Engineering, Zhengzhou University, Zhengzhou 450000, China
[2]Department of Mathematical and Systems Engineering, Shizuoka University, Shizuoka 432-8561, Japan
[3]Institute of Intelligent Systems, Deakin University, Geelong, VIC 3220, Australia

Corresponding authors: Li-Li Jia (jialilics@163.com) and Nan Lin (linnan@zzu.edu.cn)

This work was supported in part by the Nature Science Foundation of China under Grant 61602230, in part by the Research Foundation Plan in Higher Education Institutions, in part by the Education Bureau of Henan Province, China, under Grant 17A520016, and in part by the Outstanding Young Talent Research Fund of Zhengzhou University under Grant 1521337044.

ABSTRACT The appearance of generative adversarial networks (GAN) provides a new approach and framework for computer vision. Compared with traditional machine learning algorithms, GAN works via adversarial training concept and is more powerful in both feature learning and representation. GAN also exhibits some problems, such as non-convergence, model collapse, and uncontrollability due to high degree of freedom. How to improve the theory of GAN and apply it to computer-vision-related tasks have now attracted much research efforts. In this paper, recently proposed GAN models and their applications in computer vision are systematically reviewed. In particular, we firstly survey the history and development of generative algorithms, the mechanism of GAN, its fundamental network structures, and theoretical analysis of the original GAN. Classical GAN algorithms are then compared comprehensively in terms of the mechanism, visual results of generated samples, and Frechet Inception Distance. These networks are further evaluated from network construction, performance, and applicability aspects by extensive experiments conducted over public datasets. After that, several typical applications of GAN in computer vision, including high-quality samples generation, style transfer, and image translation, are examined. Finally, some existing problems of GAN are summarized and discussed and potential future research topics are forecasted.

INDEX TERMS Deep learning, generative adversarial networks (GAN), computer vision (CV), image generation, style transfer, image inpainting.

I. INTRODUCTION

Generative Adversarial Network (GAN), a generative approach proposed by Goodfellow *et al.* in 2014 [1], has become one of the most discussed topics in machine learning. GAN has made significant improvements in most computer vision tasks, as demonstrated by its remarkable achievements in image processing [2], [3], image style transfer [4], [5], classification [6], [7], and image generation [8]–[10]. Up until now, many variants of GAN models have been proposed in different fields. GAN has become one of the most active algorithms in deep learning in recent years.

Generative approaches are used to model simulated observations drawn from a probability density function, and can obtain plentiful samples. Generally speaking, generative models can be divided into two categories: traditional machine learning based algorithms, and deep learning based algorithms. Examples of the former include Restricted Boltzmann Machine(RBM) [11], Gaussian Mixture Model (GMM) [12], Naive Bayes Model (NBM) [13], Hidden Markov Model (HMM) [14] and so on. These algorithms build generative models, which use specific functions to approximate true distributions, and thus constitute the desirable interpretability and remarkable achievements of the generative models. However, each traditional generative model needs a certain functional form, which has complex expression and is hard to be designed. In addition, traditional generative models do not perform well on large scale datasets such as ImageNet [15].

To address these issues, researchers have been looking for solutions from deep learning algorithms. The deep generative models include variational autoencoders (VAE) [16] and GAN, the two most promising methods for unsupervised learning on complex distributions. The goals of VAE and GAN are to generate distributions from input data

distributions. Both VAE and GAN construct models to generate target data from hidden variables, but they are different in implementation methods. VAE generates data distributions based on the variational Bayesian inference, while GAN generates data distributions through the adversarial process. The variational approach of VAE introduces a deterministic bias to optimize the lower bound of log-likelihood rather than the likelihood itself and results in blurred generation. In contrast, GAN gradually improves the quality of generations by adversarial training process. GAN exhibits the following advantages over VAE based models: 1) GAN belongs to the type of non-parametric production-based modeling methods, which does not require prior approximate distributions of training data. 2) GAN works on the whole image and takes less time to generate samples by directly using global information.

What makes GAN so outstanding is its special structures. GAN is a deep adversarial framework consisting of a generative network named generator and a discriminative network called discriminator. The generator captures the data distributions, which wish to pass through the test of the discriminator, and the discriminator estimates the probability whether the sample is from true distributions. The GAN framework is inspired by minimax two-player game, and the competition between the generator and discriminator forces them to improve their methods until the counterfeit is undistiguishable from the true samples [1]. Both the quality of samples generated and the identification ability of the discriminator are improved interactively during the training process [17]. It is notable that the generator can be any algorithm as long as it can learn distribution of training data, and the discriminator needs to extract features and train a binary classifiers using these features. For example, convolutional neural networks (CNN) [18]–[20], recurrent neural networks (RNN) [21], [22], and long-short-term memory (LSTM) [23], [24] could be used to extract features. While the generator needs to produce detailed distributions, and as an opposite operation of CNN to produce detailed distributions, deconvolutional neural networks are generally used as generators. Combined with other models, GAN has developed rapidly in recent years, which will be briefly introduced below.

Recent development of GAN can be divided into three stages. From the time when GAN was proposed until the appearance of DCGAN [20], is the initial stage (2014.06-2015.11). The second is exploration stage (2015.11-2017.01) from the appearance of DCGAN to the appearance of WGAN [25]. From WGAN to present is rising stage (2017.01-present). When GAN was initially proposed, it didn't receive much attention, because the original GAN is difficult to control, the model is easy to collapse and the result is not satisfactory. Then a landmark model DCGAN was proposed in the exploration stage. Researchers began to find solutions to make GANs more stable by improving the structure and training skills. Meanwhile, the applications of GAN began to appear and have achieved good results, such as high-quality image generation [8], image style conversion [4], etc.. Despite wide application, theoretical explanation of why GAN has the above problems is rarely seen. In the rising phase, WGAN has provided detailed explanation of GAN's poor control and easy collapse. It also proposes a solution to improve the quality of generated results. Accordingly, many new models with better results have been proposed from different angles. In addition, GANs are applied in many new fields, such as text-image mutual generation [26], image in painting [27], etc., showing strong vitality via combination with other approaches. GAN is still in an ascendant stage towards deeper explorations and more extensive applications, indicating a wide developing prospect.

TABLE 1. Abbreviations and corresponding full names appearing in the paper.

Abbreviations	Full names	Authors
GAN	Generative Adversarial Network	Goodfellow et al. [1]
CNN	Convolutional Neural Networks	Lecun et al. [19]
RNN	Recurrent Neural Networks	Graves et al. [21]
LSTM	Long Short Term Memory	Hochreiter et al. [23]
BP	Back Propagation	Lecun et al. [18]
CGAN	Conditional GAN	Mirza et al. [28]
DCGAN	Deep Convolutional GAN	Radford et al. [20]
InfoGAN	Information GAN	Chen et al. [29]
ACGAN	Auxiliary Classifier GAN	Odena et al. [6]
EBGAN	Energy-Based GAN	Zhao et al. [30]
WGAN	Wasserstein GAN	Arjovsky et al. [25]
WGAN-GP	WGAN with Gradient Penalty	Gulrajani et al. [31]
LSGAN	Least Squares GAN	Mao et al. [32]
BEGAN	Boundary Equilibrium GAN	Berthelot et al. [8]
DRAGAN	Degenerate Avoided GAN	Kodali et al. [33]
SNGAN	Spectral Normalization GAN	Miyato et al. [34]
SAGAN	Self-Attention GAN	Zhang et al. [35]
JR-GAN	Jacobian Regularization GAN	Nie et al. [36]
CapsGAN	Capsule GAN	Saqur et al. [37]
BWGAN	Banach Wasserstein GAN	Adler et al. [38]
DEGAN	Decoder-Encoder GAN	Zhong et al. [39]
VAE	Variational Autoencoders	Kingma et al. [40]
CVAE	Conditional VAE	Kingma et al. [41]
FID	Frechet Inception Distance	Heusel et al. [42]
IS	Inception Score	Salimans et al. [43]
StackGAN	Stacked GAN	Zhang et al. [44]
LAPGAN	Laplacian Pyramid of GAN	Fergus et al. [45]
Pix2pix	Pixels to Pixels	Isola et al. [4]
CycleGAN	Cycle-consistent GAN	Zhu et al. [46]
DiscoGAN	Discover Cross-domain GAN	Kim et al. [47]
DTN	Domain Transfer Network	Taigman et al. [48]
Sem-GAN	semantically consistent GAN	Cherian et al. [26]
PGGAN	Global patch GAN	Demir et al. [49]
RTT-GAN	Recurrent Topic-Transition GAN	Liang et al. [50]
TP-GAN	Two-pathway GAN	Huang et al. [27]
MoCoGAN	Motion and Content GAN	Tulyakovet et al. [51]

The rest of this paper is organized as follows. The mechanism, advantages, and disadvantages of the generator and discriminator are introduced in section II. Evolutions of typical GAN models are listed in section III. Several variants of GAN and their applications and improvements are described in section IV, followed by summaries and future trends of GAN in section V.

II. GENERATIVE ADVERSARIAL NETWORKS

In this section, we will introduce the principle and architecture of GAN, with a discussion of its advantages and disadvantages. The key idea of GAN is inspired by the minimax two-person zero-sum game in which one player benefits only at the equal loss of the other. In GAN, the two players correspond to the generator and the discriminator. The goal of the generator is to deceive the discriminator, and the goal

of the discriminator is to determine whether a sample is from real distribution. The output of the discriminator is a probability that the input sample is a true sample. A higher probability indicates that the sample is more likely from real data. Conversely, the closer the probability is to 0, the more likely the sample is fake. When the probability infinitely approaches to 1/2, the optimal solution is obtained, as the discriminator finds it is hard to check fake samples.

A. GAN NETWORK STRUCTURE

GAN consists of two networks: the generator (G) and the discriminator (D). Essentially, both G and D are implicit function expressions that are usually implemented by deep neural networks [52]. Fig. 1 shows the model structure of GAN, where G captures the data distribution from real sample and maps it to a new space. The generated data is recorded as $G(z)$, whose distribution is recorded as $p_g(z)$. The aim of GAN is to make $p_g(z)$ as similar as the distribution of training sample $p_r(\boldsymbol{x})$. The input of D can be either real data $\boldsymbol{x}$ or generated data $G(z)$. The result of D is a probability or a scalar predicting whether the input of D is from real distribution.

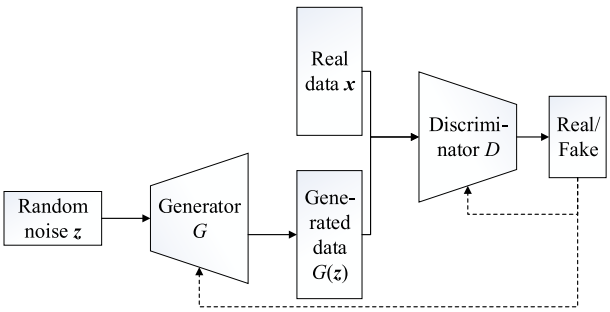


FIGURE 1. The basic framework of GAN, which includes a generator G and a discriminator D, trained through back-propagation algorithm.

1) THE GENERATOR

The generator is represented by a differentiable function G. G collects random variables z from the prior distribution and maps them through a neural networks to pseudo-sample distributions $G(z)$, which is an upsampling process. The input z generally uses Gaussian noise which is a random variable or a random variable in latent space. During the training of GAN, the parameters of G and D are updated iteratively. When G is being trained, the parameters of D are fixed. The data generated by G is labeled as fake and is input into D. Between the output of the discriminator $D(G(z))$ and the sample label, the error is calculated, and the error back propagation algorithm is used to update the parameters of G.

G only imposes a few constraints on input variables which can not only be entered to the first layer but also be entered to the last layer. Moreover, the noise can be added to hidden layers, in the way of either summation product or mosaic. GAN does not limit the input dimension of z, which is usually a 100-dimensional vector. Furthermore, G must be differentiable because feedback passed through the discriminator will return gradients to update parameters of G and D.

2) THE DISCRIMINATOR

The goal of discriminator D is to determine whether the input is from real sample and provides a feedback mechanism that refines weight parameters of G. When the input is real sample $\boldsymbol{x}$, the output of D approaches to 1. Otherwise, the output of D approaches to 0.

When the discriminator is trained, the G is fixed. D obtains the positive sample $\boldsymbol{x}$ from the real dataset and the negative sample $G(z)$ generated by the generator. Both of them are input into D, and the output of D and sample labels are used to calculate the error. Finally, the error back propagation algorithm is used to update the parameters of the discriminator.

B. LOSS FUNCTION

The loss function of GAN is based on a two-player minimax game, which contains two neural networks that compete against each other in a zero-sum game framework [53]. Two players are represented by two functions which are differentiable with respect to their inputs and parameters. The discriminator function is denoted by D, whose inputs and parameters are $\boldsymbol{x}$ and $\boldsymbol{\theta}^{(D)}$, and whose loss function is

$$V(D, \boldsymbol{\theta}^{(D)}) = -E_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}[logD(\boldsymbol{x})] - E_{z\sim p_g(z)}[log(1 - D(G(z)))] \quad (1)$$

the p_r represents real data distribution and the p_g represents generated data distribution. The generator function is denoted by G, whose input and parameters are z and $\boldsymbol{\theta}^{(G)}$, and whose loss function is

$$V(G, \boldsymbol{\theta}^{(G)}) = E_{z\sim p_g}[log(1 - D(G(z)))] \quad (2)$$

Both players have their own loss functions. D needs to maximize $\boldsymbol{V}^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)})$ by updating $\boldsymbol{\theta}^{(D)}$, and G needs to minimize $\boldsymbol{V}^{(G)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)})$ by updating $\boldsymbol{\theta}^{(G)}$. Both players' loss functions depend on parameters of each other. They cannot update the other parameter and will not stop training until a Nash equilibrium is achieved [54]. GAN is actually a minimax optimization problem, whose loss function is defined as:

$$\min_G \max_D V(D, G) = E_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}[logD(\boldsymbol{x})] + E_{z\sim p_g(z)}[log(1 - D(G(z)))] \quad (3)$$

The first part of formula (3) represents that D makes objective function as large as possible when real data is input. The latter represents that when the generated data is input to D, D makes the output $D(G(z))$ approach 0, while the goal of G is to make the output D close to 1. When two models have been sufficiently trained, the game eventually reaches a Nash equilibrium. Ideally, D cannot tell whether the input is real data or generated data, that is, when the minimum $p_r(\boldsymbol{x})/(p_r(\boldsymbol{x}) + p_g(z))$ equals to 1/2, the model is optimal. It means that D cannot distinguish real samples and generated samples.

C. THE CHALLENGES OF GAN

Though GAN has achieved great success, there are still some problems such as gradient disappearance, difficulty in training, and poor diversity. The reasons of these problems will be explained.

According to the loss function (3), the input of D includes two parts: the real data set distribution p_r and the generated data distribution p_g. The loss function can be also written as:

$$\min_G \max_D V(D,G) = E_{\boldsymbol{x}\sim p_r}[logD(\boldsymbol{x})] + E_{\boldsymbol{x}\sim p_g}[log(1-D(\boldsymbol{x}))] \quad (4)$$

The core problem of GAN is to measure and minimize the distance between the two distributions p_r and p_g. When the generator is fixed, the training of discriminator is also a process of minimizing the cross entropy, and the loss function of D is:

$$V(D) = -p_r(\boldsymbol{x})[logD(\boldsymbol{x})] - p_g(\boldsymbol{x})[log(1-D(\boldsymbol{x}))] \quad (5)$$

Let the derivative of equation (5) be 0, then the optimal discriminator $D(x)$ has the following shape

$$D^*(\boldsymbol{x}) = \frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x}) + p_g(\boldsymbol{x})} \quad (6)$$

There are two ways to express the loss function of the generator, one of which can be written as:

$$V(G) = E_{\boldsymbol{x}\sim p_g}[log(1-D(\boldsymbol{x}))] \quad (7)$$

In the case where the discriminator is optimal, a generator-independent item is added to equation (7) to make a new equation $V(D,\boldsymbol{\theta}^{(G)})$ (8), where $\boldsymbol{\theta}^{(G)}$ is a neural network parameterized by θ.

$$V(D,\boldsymbol{\theta}^{(G)}) = E_{\boldsymbol{x}\sim p_r}[logD(\boldsymbol{x})] + E_{\boldsymbol{x}\sim p_g}[log(1-D(\boldsymbol{x}))] \quad (8)$$

In the following, merging Eq.(6) into Eq.(8), we obtain

$$\begin{aligned} V(D,\boldsymbol{\theta}^{(G)}) &= E_{\boldsymbol{x}\sim p_r} log\frac{p_r(\boldsymbol{x})}{\frac{1}{2}[p_r(\boldsymbol{x})+p_g(\boldsymbol{x})]} \\ &\quad + E_{\boldsymbol{x}\sim p_g} log\frac{p_g(\boldsymbol{x})}{\frac{1}{2}[p_r(\boldsymbol{x})+p_g(\boldsymbol{x})]} - 2log2 \\ &= KL(p_r||\frac{p_r+p_g}{2}) + KL(p_g||\frac{p_r+p_g}{2}) - 2log2 \end{aligned} \quad (9)$$

where KL is the Kullback-Leibler divergence (KL divergence) defined by the following equation

$$KL(p_r||p_g) = E_{\boldsymbol{x}\sim p_r}[log\frac{p_r(\boldsymbol{x})}{p_g(\boldsymbol{x})}] \quad (10)$$

KL divergence is a non-symmetric measurement of similarity between two distributions. There is another similar measurement named Jensen-Shannon divergence (JS divergence), which is defined by Eq.(11).

$$JSD(p_r||p_g) = \frac{1}{2}KL(p_r||\frac{p_r+p_g}{2}) + \frac{1}{2}KL(p_g||\frac{p_r+p_g}{2}) \quad (11)$$

Then, the Eq.(9) can be rewritten as

$$V(D,\boldsymbol{\theta}^{(G)}) = 2JSD(p_r||p_g) - 2log2 \quad (12)$$

The optimization of the original loss function is equivalent to minimizing the JS divergence $JSD(p_r||p_g)$. The closer the two distributions are, the smaller the $JSD(p_r||p_g)$ is. By optimizing the JS divergence, the generated samples look more and more like real ones. However, when p_r and p_g have no or little overlapped parts, the $JSD(p_r||p_g)$ is a constant log2, implying that its gradient with respect to p_r and p_g is zero, which makes it hard to train the model. Actually, the possibility that no overlap between p_r and p_g is 1, when the support set of p_r and p_g is low-dimensional manifold in high dimension. This is the main cause of gradient vanishing and disappearance.

Moreover, note that the probability of p_r and p_g overlap is too low to be calculated, because the input of the generator is generally a low-dimensional coding vector (such as 100). But the dimension of real sample is usually much larger than 100, so the JS divergence is a constant, resulting in 0 of the gradient of generator and thus its disappearance.

Another challenge of GAN is that it is difficult to train the network. When the discriminator is trained optimally, the feedback from D is significantly close to 0, leading to the decrease of convergence rate. When the discriminator is not well trained and then the gradient of generator is not accurate. GAN can only work properly when the discriminator is well trained, but there is no indicator to show whether the discriminator is properly trained or not.

Poor diversity of GAN generation is another problem and will be addressed in the following. Firstly, we introduce another expression of generator loss function [55].

$$V(G) = E_{\boldsymbol{x}\sim p_g}[-logD(\boldsymbol{x})] \quad (13)$$

Making the $KL(p_g||p_r)$ to the following transformations

$$\begin{aligned} KL(p_g||p_r) &= E_{\boldsymbol{x}\sim p_g}[log\frac{p_g(\boldsymbol{x})}{p_r(\boldsymbol{x})}] \\ &= E_{\boldsymbol{x}\sim p_g}[log\frac{p_g(\boldsymbol{x})/(p_r(\boldsymbol{x})+p_g(\boldsymbol{x}))}{p_r(\boldsymbol{x})/(p_r(\boldsymbol{x})+p_g(\boldsymbol{x}))}] \\ &= E_{\boldsymbol{x}\sim p_g}[log\frac{1-D^*(\boldsymbol{x})}{D^*(\boldsymbol{x})}] \\ &= E_{\boldsymbol{x}\sim p_g}[log(1-D^*(\boldsymbol{x}))] - E_{\boldsymbol{x}\sim p_g}[logD^*(\boldsymbol{x})] \end{aligned} \quad (14)$$

The equivalent equation (13) can be obtained by the formulas (12) and (14)

$$E_{\boldsymbol{x}\sim p_g}[-logD^*(\boldsymbol{x})] = KL(p_g||p_r) - 2JSD(p_r||p_g) + 2log2 + E_{\boldsymbol{x}\sim p_r}[logD^*(\boldsymbol{x})] \quad (15)$$

The last two terms in Eq.(15) are not functions of the generator, and therefore, to minimize (15) is equivalent to minimize Eq.(16).

$$KL(p_g||p_r) - 2JSD(p_r||p_g) \quad (16)$$

However, there are two problems with Eq.(16). Firstly, to minimize (16), the KL divergence should be minimized and the JS divergence should be maximized, which are contradictory and cause unstable BP process. Secondly, the preceding $KL(p_g||p_r)$ divergence term is much different from the previous $KL(p_r||p_g)$. Note that the KL divergence is asymmetrical. Taking $KL(p_g||p_r)$ as an example.

When $p_g(x) \to 0$ and $p_r(x) \to 1$, then $KL(p_g||p_r) \to 0$; When $p_g(x) \to 1$ and $p_r(x) \to 0$, then $KL(p_g||p_r) \to +\infty$.

The punishment is different for the above two cases. In the first case, the generated sample lacks diversity and the penalty is small. Whereas in the second case, the generated sample lacks accuracy and the punishment is very huge. Accordingly, the generator tends to generate some repetitive but safe samples rather than diverse samples to avoid penalties, because of the huge punishment for the latter. The above phenomenon is called mode collapse.

All of the above analysis shows that under (approximate) optimal discriminator of the original GAN, the first kind of generator loss Eq.(7) faces the problem of gradient disappearance and difficulties in training, and the second kind of generator loss Eq.(13) suffers from the optimization goal absurdity, gradient instability and mode collapse. The fundamental causes of these problems can be attributed to two points. Firstly, the distance measurement (such as KL divergence and JS divergence) of the equivalent optimization is unreasonable. Secondly, the generated distribution is difficult to overlap with the real distribution [55].

D. THE ADVANTAGES OF GAN

Since the emergence of GAN, it has been applied in different fields with modifications either structurally improved or theoretically developed. Its advantages include the following aspects: 1) There are few prior assumptions and hardly any hypothesis about data sets which almost could be any distribution in original GAN proposed by Goodfellow. 2) The final goal is that GAN has infinite modeling power and can fit all distributions. 3) The design of GAN model is simple and it is not necessary to pre-design complex function models. 4) GAN provides a powerful method for unsupervised deep learning models, and it subverts traditional artificial intelligence (AI) algorithms which are limited by human thinking. 5) GAN uses machines to interact with machines through the continuous confrontation which can learn inherent laws in the real world after adequate data training.

III. THE EVOLUTION OF GAN MODEL

To solve problems of the original GAN, such as gradient disappearance, unstable training, and poor diversity, many new GAN models have been proposed to increase the stability and to improve qualities of generated results [56], [43]. In this section, we will introduce the evolution of GAN models, including deep convolutional generative adversarial network (DCGAN) [20], conditional GAN (CGAN) [28], Wasserstein GAN (WGAN) [25], WGAN with gradient penalty (WGAN-GP) [31], Energy-Based GAN (EBGAN) [30], Boundary Equilibrium GAN (BEGAN) [8], Information GAN (InfoGAN) [29], Least Squares GAN (LSGAN) [32], Auxiliary Classifier GAN (ACGAN) [6], Degenerate avoided GAN (DRAGAN) [33], Spectral Normalization GAN (SNGAN) [34], Jacobian Regularization GAN (JR-GAN) [36], CapsGAN [37], Banach Wasserstein GAN (BWGAN) [38], Decoder-Encoder GAN (DEGAN) [39].

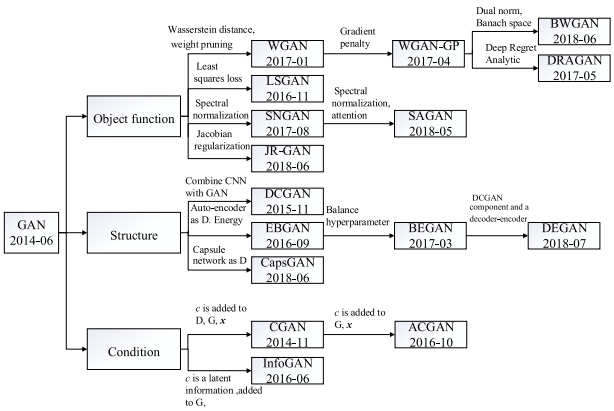


FIGURE 2. A classification of GAN models.

To be specific, we classify GANs into different types in terms of objective functions, structures and conditions, as shown in Fig. 2. The objective functions belong to the improvements of loss function, the structures and conditions belong to the developments of architecture. The time of each model proposed is also marked in the figure to show the pedigree relationships between them.

The evolution of GAN models are classified from two aspects: the development of the architecture and the improvement of loss function. The structural improvement combined GANs with other models or other algorithms. For example, the structure includes different models that combined GAN with CNN [20], capsule network [37] and encoder decoder [39]. The conditional GANs are to enhance the control of GAN. In this section three different GAN models CGAN, ACGAN and InfoGAN are introduced according to the way that conditions work. On the other hand, objective functions refer to the improvement of loss function in respect of the modifications of theory. Four models, WGAN for example, use Wasserstein distance to improve their loss function. LSGAN introduces least squares method as its loss function. SNGAN and SAGAN use spectral normalization, and JR-GAN proposes Jacobi regularization method. Their evolutionary relationships are shown in Fig. 2. Furthermore, some models show both theoretical and structural innovations, but are classified only to one type based on their distinctive features. For example, even though EBGAN changes and uses reconstruction loss as object function, its structural improvements are more typical. In a word, we evaluate classical models quantitatively and qualitatively through experiments and analyze each model fairly and comprehensively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

A. THE EVOLUTION OF GAN STRUCTURE

The improvement of GAN structure is mainly from stabilizing the model and reducing convergence delay. To better control the models, different conditional GANs that can carry more information (such as CGAN, InfoGAN, ACGAN) are proposed.

1) CONDITIONAL GENERATIVE ADVERSARIAL NETWORK (CGAN)

As a method of unsupervised learning, GAN learns the law of probability distribution from unlabeled datasets and expresses it during a slow and free process. However, when the dataset is complex or large-scale, it is difficult for GAN to control generated results. To solve this problem, a natural idea is to add constraints and set targets for the generator. This forms CGAN. It takes random variable $\boldsymbol{z}$ and real data $\boldsymbol{x}$, together with a conditional variable $\boldsymbol{c}$ to guide the data generation process. Thus, the convergence speed has been greatly accelerated. The structure of CGAN is shown in Fig. 3.

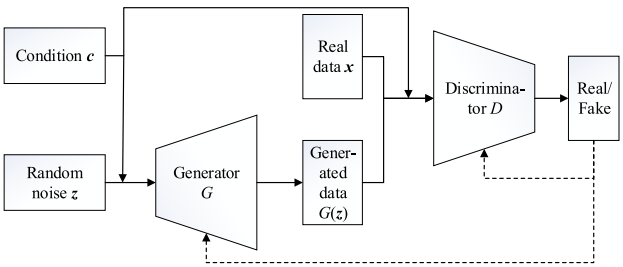


FIGURE 3. The basic framework structure of CGAN.

The conditional variable $\boldsymbol{c}$ of CGAN can be a category label which turns the unsupervised GAN into a supervised model; $\boldsymbol{c}$ can be texts, such as sentences which describe the corresponding images; $\boldsymbol{c}$ can also be a particular generated target, which proposes a goal to learn. CGAN can not only generate images with specified categories and labels, but also use image features as a conditional variable to generate word vector for the image. This straightforward improvement proves to be very effective and is widely used in subsequent works.

2) DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORK (DCGAN)

One milestone in the history of GAN is DCGAN [20], whose structure of generator is shown in Fig. 4. DCGAN combines GAN with CNN, which performs well in the field of computer vision. DCGAN sets a series of limitations on CNN's network topology, so that it can be stably trained and use learned feature representations to classify images. DCGAN improves the quality of generated images by making the following improvements. Firstly, DCGAN uses strided convolutions on the discriminator and fractional-strided convolutions on the generator to replace pooling layers [57]. Generally, CNN is used to extract features, but the CNN structure in DCGAN needs to generate samples, which is opposite with feature extraction. The strided convolutions and fractional-strided convolutions can transmit most of the information to next layer to ensure the completeness and clarity of generated samples. Secondly, DCGAN uses Batch Normalization algorithm to solve the problem of gradient disappearance. The BN algorithm solves poor initializations, conveys the gradient to each layer and prevents the generator from converging all samples to the same point [58]. Thirdly, different activation functions are used in DCGAN, such as Adam optimization [59], ReLU activation function [60], leakyReLU activate function [61]. The results show the good performance of DCGAN in practice, and confirm the capability of GAN structure in generating samples. DCGAN is generally regarded as the standard when compared with different GAN models.

Other structural improvement models will also be introduced, such as Information GAN (InfoGAN) [29], Auxiliary Classifier GAN (ACGAN) [6], Energy-Based GAN (EBGAN) [30], CapsGAN [37] and Decoder-Encoder GAN (DEGAN) [39]. InfoGAN adds latent information $\boldsymbol{c}$ as the control condition of the generator, and $\boldsymbol{c}$ does not enter the discriminator, but the control information $\boldsymbol{c}$ will be output by the discriminator, which allows InfoGAN to learn more information.

The recent ACGAN model adds a category label on the basis of CGAN, which requires the discriminator to output both the probability and the category. EBGAN interprets GAN from an energy perspective, which uses an automatic encoder as discriminator and reconstruction loss as loss function. In addition, EBGAN introduces a pull-away item to prevent generators from focusing on one or a few modes. EBGAN shows better training stability and enhanced robustness, which can reduce human work for regulating GAN. CapsGAN combines GAN and Capsule network [62] by replacing the CNN of the discriminator in DCGAN with a capsule network. CapsGAN uses dynamic routing algorithm that occurs between the main capsule layer and the output digital capsule. In addition, CapsGAN uses binary cross entropy as a loss function that allows the model to converge without any pattern collapse. DEGAN consists of two parts: a DCGAN component and a pre-trained decoder-encoder structure, combining both adversarial training and variational Bayesian inference to improve image generation performance. Furthermore, the hidden space loss function is added to the adversarial loss function to enhance the model stability.

B. THE IMPROVEMENT OF GAN LOSS FUNCTION

Gradient disappearance is one of the most common problems in the training of GAN, because the generator is generally an encoding vector sampled from random distribution of low dimensions (ie, $\boldsymbol{z}$ is usually taken as 100 dimensions), but it is then used to generate high-dimensional samples via neural networks. Even if the sample dimensions of the generator are definite, the probability distribution of generated samples is defined in a space of 4096 dimensions. All possible changes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

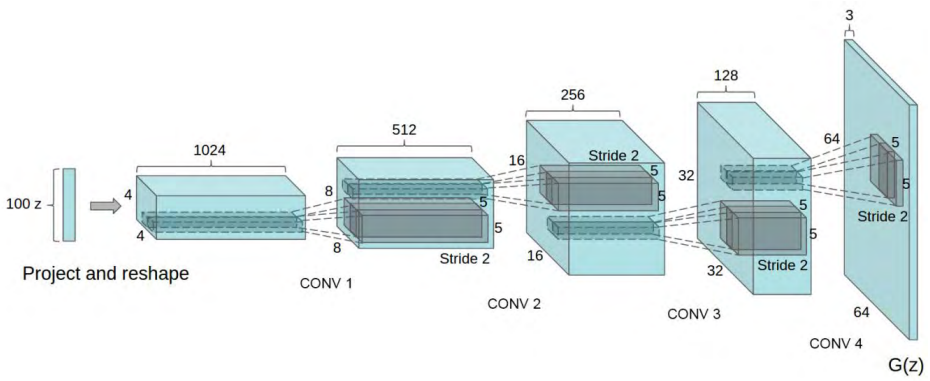


FIGURE 4. The structure of the generator of DCGAN [20].

have been defined by the 100-dimensional random distribution, but the practical dimension is still 100. Therefore, the support set for generated sample distributions constitutes a low-dimensional manifold with up to 100 dimensions in a 4096-dimensional space. Hence, the probability of the overlapping between the generated distribution and the data distribution is close to 0. Then the divergence of Jensen-Shannon (JS), which is a measure of similarity between generated distributions and true distributions, will become a constant, and the gradient will disappear so the model training is unable to continue [63].

1) WASSERSTEIN GENERATIVE ADVERSARIAL NETWORK (WGAN)

To solve the problem that JS distance is unable to, Wasserstein distance (also called Earth-Mover (EM) distance) (Eq.(17)) was firstly used in WGAN [25] to measure the distance between real samples and generated ones instead of JS divergence.

$$W(p_r, p_g) = \inf_{\gamma \sim \prod(p_r, p_g)} E_{(\boldsymbol{x},\boldsymbol{y})\sim\gamma}[\|x - y\|] \tag{17}$$

$\prod(p_r, p_g)$ is the set of all possible joint distributions that p_r and p_g combine, and γ is every possible distribution. Wasserstein distance has the following advantages: It can measure distance even when two distributions do not overlap; it has excellent smoothing properties; and it can solve the problem of gradient disappearance to some extent. In addition, WGAN solves the problem of instability in training and makes generated samples diverse. There is no need to carefully balance the training of G and D [64]. However, using Wasserteion distance needs to satisfy strong conditional lipschitz continuity, so WGAN limits the weight to a range to enforce the continuity of lipschitz. However, the forced cutting weights can easily cause the gradient to disappear or explode.

2) WGAN WITH GRADIENT PENALTY (WGAN-GP)

To solve the problem of disappearance or explosion of the gradients and find a suitable way to meet the lipschitz continuity, a gradient penalty method termed WGAN with gradient penalty (WGAN-GP) is proposed by Gulrajani *et al.* [31]. WGAN-GP replaces the weight pruning in WGAN to implement Lipschitz constraint method. The Lipschitz constraint requires the gradient of the discriminator no more than K. The discriminator tries to widen the score gap of the true and false samples as much as possible. After the discriminator is fully trained, the gradient norm will get close to K. K takes 1 to simplify the calculation, the new discriminator loss of WGAN-GP is

$$V(D) = -E_{(\boldsymbol{x})\sim p_r}[D(\boldsymbol{x})] + E_{(\boldsymbol{x})\sim p_g}[D(\boldsymbol{x})] + \lambda E_{(\boldsymbol{x})\sim p_x^*}[\|\nabla_x D(\boldsymbol{x})\|_p - 1]^2 \tag{18}$$

In the Eq.(18), $x^* = \epsilon x_r + (1-\epsilon)x_g$, $x_r \sim p_r$, $x_g \sim p_g$ and $\epsilon \sim Uniform[0, 1]$, x^* is a random interpolation sample on the line of x_r and x_g. Experiments show that the quality of samples generated by WGAN-GP is better than that of WGAN. WGAN-GP provides stable training without hyperparameters and trains a variety of generating tasks successfully. However, experiments also show that the convergence rate of WGAN-GP is slower, as it takes more time to converge under the same dataset. WGAN and WGAN-GP have improved the GAN on optimization methods and constraint approaches without changing the structure of it. In essence, they improve the original GAN by increasing constraints to generate better samples. The comparison of four landmark models of GANsis shown in Table 2. DCGAN and CGAN are representatives of structural improvements of GAN. They design more stable network structures or add conditions. WGAN and WGAN-GP improve the objective function theoretically, and make the training more stable.

Some other methods that modify the loss function in different ways will be introduced. LSGAN replaces cross-entropy loss of objective function with the least squares loss, and it partly repaires two defects of low quality and instability of training process. BEGAN proposes a balance concept to balance the abilities of the discriminator and provides a hyperparameter that can make a balance between the diversity of image and the generation quality. DRAGAN uses a gradient penalty to avoid degraded local equalization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

TABLE 2. Comparisons of typical GAN models.

GAN model	Improvements	Advantages	Disadvantages	Scenarios
DCGAN [20]	Combined with CNN; stride convolution, fractional-strided convolutions, batchnorm, ReLU, LeakyReLU.	More stable, easy convergence, generate diverse samples.	Training different data needs to adjust parameters, model collapse, gradients disappear or explode.	Suitable for most scenarios, is one of the highest usage models.
CGAN [28]	Add a conditional variable c to constrain model and guide data generation.	Add a label to generate the specified target convergence is faster.	More requirements for data set, data set need to have a tag or marked.	Supervised, semi-supervised learning, generate the scene for specified target.
WGAN [25]	Weight pruning.	The training is more stable, theoretically solving the problem of gradient disappearance.	Due to inappropriate pruning of weights, gradients may disappear or explode.	Other GAN models do not converge and models collapse.
WGAN-GP [31]	Replace gradient pruning with gradient penalties.	No need to balance the generator and discriminator, the training is stable, out of box, can directly process text.	Slow convergence, less diversity of generated samples than DCGAN.	When model parameters are uncertain or need to deal with text directly.

TABLE 3. The loss functions of discriminators and generators.

Model	Discriminator Loss	Generator Loss
GAN [1]	$L_D^{GAN} = -E_{x\sim p_{data}}[log(D(x))] - E_{z\sim p_z}[log(1-D(G(z)))]$	$L_G^{GAN} = -E_{x\sim p_z}[log(D(G(z)))]$
CGAN [28]	$L_D^{CGAN} = -E_{x\sim p_{data}}[log(D(x,c))] - E_{z\sim p_z}[log(1-D(G(z),c))]$	$L_G^{CGAN} = -E_{x\sim p_z}[log(D(G(z),c))]$
WGAN [25]	$L_D^{WGAN} = -E_{x\sim p_{data}}[(D(x))] + E_{z\sim p_z}[(1-D(G(z)))]$	$L_G^{WGAN} = -E_{x\sim p_z}[(D(G(z)))]$
WGAN-GP [31]	$L_D^{WGAN-GP} = L_D^{WGAN} + \lambda E_{x\sim p_{data}}[(\parallel \bigtriangledown D(\alpha x + (1-\alpha G(z))) \parallel -1)^2]$	$L_G^{WGAN-GP} = -E_{z\sim p_z}[(D(G(z)))]$
EBGAN [30]	$L_D^{EBGAN} = E_{x\sim p_{data}}[(D_{AE}(x))] + E_{z\sim p_z}[\max(0, m - D_{AE}(G(z)))]$	$L_G^{EBGAN} = \mathrm{E}_{x\sim p_z}[(\mathrm{D}_{AE}(\mathrm{G}(\mathrm{z})))]$
BEGAN [8]	$L_D^{BEGAN} = E_{x\sim p_{data}}[(D_{AE}(x))] - k_t E_{z\sim p_z}[D_{AE}(G(z))]$	$L_G^{BEGAN} = \mathrm{E}_{x\sim p_z}[(\mathrm{D}_{AE}(\mathrm{G}(\mathrm{z})))]$
InfoGAN [29]	$L_D^{InfoGAN} = L_D^{GAN} - \lambda L_1(c,c')$	$L_G^{InfoGAN} = L_G^{GAN} - \lambda L_1(c,c')$
LSGAN [32]	$L_D^{LSGAN} = -E_{x\sim p_{data}}[(D(x)-1)^2]$	$L_G^{LSGAN} = -E_{x\sim p_z}[(D(G(z))-1)^2]$
ACGAN [6]	$L_D^{ACGAN} = L_D^{GAN} + E_{x\sim p_{data}}[P(class=c \mid x)] + E_{z\sim p_z}[P(class=c \mid G(z))]$	$L_G^{ACGAN} = L_G^{GAN} + E_{z\sim p_z}[P(class=c \mid G(z))]$
DRAGAN [33]	$L_D^{DRAGAN} = L_D^{GAN} + \lambda E_{z\sim p_z+\mathcal{N}(0,c)}\left[(\parallel \bigtriangledown D(G(z)) \parallel -1)^2\right]$	$L_G^{DRAGAN} = E_{z\sim p_z}[log(1-D(G(z)))]$

SNGAN proposes a new weight regularization method called spectral normalization, which can stabilize training process of the discriminator. The spectral normalization allows parameter matrix to use as many features in images as possible and satisfy local 1-Lipschitz constraints. SNGAN could even fit all 1000 classes of ImageNet. BWGAN replaces the l^2 norm in WGAN-GP with dual norm and generalizes the WGAN theory with gradient penalty on Banach space. Furthermore, BWGAN extends WGAN-GP to any separable complete normed space. JR-GAN suggests a new Jacobian regularization method, which can simultaneously alleviate the Phase Factor and the Conditioning Factor to ensure good convergence behavior of GAN. While the gradient-based regularization methods can only avoid one factor, the other factors are more serious. JR-GAN makes the GAN architecture more robust. Table 3 shows loss functions of different models.

C. THE EXPERIMENTAL RESULTS OF DIFFERENT MODELS

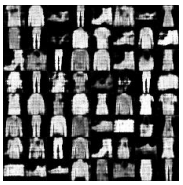
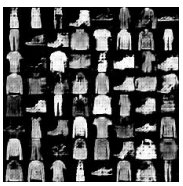
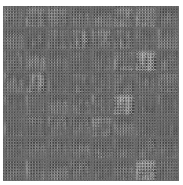
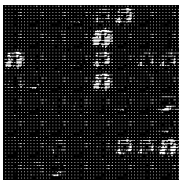
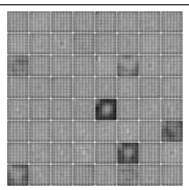
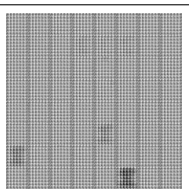
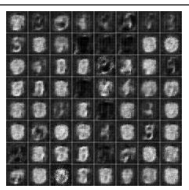
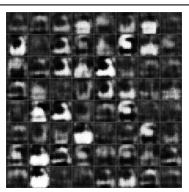
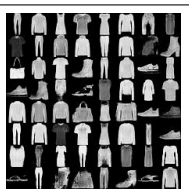
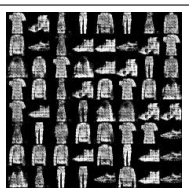
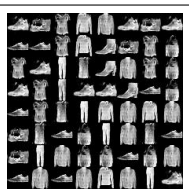
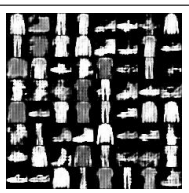
To evaluate the performance of different GAN models, DCGAN, CGAN, WGAN, WGAN-GP, EBGAN, BEGAN, INFOGAN, LSGAN, VAE [40], Conditional Variational Autoencoders (CVAE) [41], ACGAN, DRAGAN are compared on the same datasets. The loss functions of generators and discriminators are shown as in Table 3. The code is from [65] and [66], with the generated results compared when relevant parameters are consistent (see Table 4 and Table 5). In this section, we will introduce the basic setups and network structures of the experiment, and analyze experimental results from qualitative and quantitative comparisons.

1) EXPERIMENTAL SETUP

The experiments are conducted on two common public datasets: MNIST [19] and Fashion-MNIST [67]. The latter is a new image database like the MNIST and includes 10 categories of frontal images of 70,000 different products, including T-shirts, pants, pullovers, skirts, jackets, sandals, sweatshirts, sneakers, bags and ankle boots, labeled from 0 to 9. In addition, the name, size, format and training set of Fashion-MNIST are exactly the same as the original MNIST that contains 60,000 training pictures and 10,000 test pictures, with the size of 28 × 28 grayscale. The listed results of different models are to verify the performance of generation from human vision. The experimental results are shown in Table 4 and Table 5. The images in Table 4 generated in Epoch1 reflect the convergence rate of different models. Those models with fast convergence

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

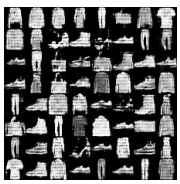
TABLE 4. Comparisons of images generated by different models.

The type of GAN	Epoch1	Epoch40	Epoch1	Epoch40
DCGAN [20]				
CGAN [28]				
WGAN [25]				
WGAN-GP [31]				
EBGAN [30]				
BEGAN [8]				
InfoGAN [29]				
LSGAN [32]				

include WGAN, CGAN, DCGAN, LSGAN and InfoGAN. The images generated in Epoch40 reflect the generation quality of different models. Those models with high generation quality are BEGAN, WGAN, LSGAN, WGAN-GP, InfoGAN, EBGAN, DCGAN, CGAN. The images in Table 5 generated in Epoch1 reflect the convergence rate of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

TABLE 5. Comparisons of images generated by different models.

The type of GAN	Epoch1	Epoch40	Epoch1	Epoch40
VAE [40]				
CVAE [41]				
ACGAN [6]				
DRAGAN [33]				

different models. The models with fast convergence include ACGAN, CVAE, DRAGAN, VAE. The images generated in Epoch40 reflect the generation quality of different models. Those models with high generation quality are CVAE, DRAGAN, ACGAN, VAE.

Table 4 and Table 5 show generated results of different GAN models, whose network architectures of generator and discriminator are the same as InfoGAN based on DCGAN. To compare the key ideas of different GAN models fairly, their parameters and settings are kept consistent except VAE, EBGAN and BEGAN. The number of output nodes in encoder is different for VAE and GAN. Small modification is made for EBGAN and BEGAN, since they adopt auto-encoder structure for the discriminator. The experiments use unified setup: a discriminator includes an input layer, an output layer, two convolutional layers and a fully connected layer. The generator includes an input layer, an output layer, two fully connected and deconvolutional layers, and identical activation functions and batch standardization operations.

2) QUALITATIVE COMPARISONS

In terms of the quality of generated images from the first iteration in TABLE 4 and TABLE 5, WGAN converges the fastest and generate clearer images than others. The edges of images generated by WGAN are easy to distinguish from the background, since WGAN uses weight penalty which can learn image distributions more quickly. In contrast, WGAN-GP has the slowest convergence rate in all models. The final generative results also show that its generated images in 40 iterations are not clear, while converged under the same conditions with other models. Instead of using weight penalty of WGAN, WGAN-GP calculates the weight gradient according to the input of the discriminator, then it penalizes the gradient norm [31]. WGAN-GP needs to train more parameters, so the convergence rate is slow. But WGAN-GP is out of box and needs no adjustment, in other words, parameter changes have little effects on its learning rate and the model is pretty stable. Therefore, WGAN-GP can produce high-quality images after 60 iterations. DRAGAN is similar to WGAN-GP with slow speed but high-quality images. It uses gradient clipping to avoid local equilibrium and obtains more stably training.

In the above tables, CGAN, ACGAN, InfoGAN and CVAE add controllable conditions in their generators, and they can generate samples in specified category with faster convergent rate. The differences are that CGAN directly adds conditional information $\boldsymbol{c}$ together with random variable to the input of the generator to control output mode. While ACGAN adds information of category $\boldsymbol{c}$ to both D and G, telling G how to generate better category simulations. InfoGAN explores new ways to generate samples of the same categories by maximizing mutual information. CVAE is a variation of VAE which adds controllable information $\boldsymbol{c}$ to encoders and decoders.

TABLE 6. The FIDs of different models trained on MNIST AND fashion-MNIST, the smaller the FIDs, the better the performance (* represents the model easily crashed during training).

Method	Time on MNIST(second)	Time ON fashion-MNIST(second)	FID on MNIST	FID on fashion-MNIST
DCGAN	3519.02	3607.20	11.51	8.22
CGAN	3814.65	3730.80	20.98	11.92
WGAN	3632.44	3460.58	13.40	28.17
WGAN-GP	4793.22	4928.91	1.92	5.76
EBGAN	2125.10	2126.27	17.05	41.32
BEGAN	2571.38	2473.87	19.05	15.90
InfoGAN	4859.36	4555.40	15.96	12.93
LSGAN	3584.98	3577.26	3.84	14.72
VAE	1902.47	1941.14	61.95	69.84
CVAE	2024.56	1984.63	18.37	36.64
ACGAN	5453.77	5370.58	62.24	49.11*
DRAGAN	4534.20	4594.89	11.07	6.40

Experiments show the quality of images generated by CVAE is lower than that of GAN, since VAE uses KL distance and relies on a hypothetical loss function without adversarial training. While GAN directs G and D to compete with each other without assuming a single loss function. In the experiment, ACGAN generates clear images on epoch 11, but it tends to fall into mode-collapse in latter training. Its stable training requires suitable hyper-parameters. Instead of inputting noise $\boldsymbol{z}$, the generator of InfoGAN inputs a control variable $\boldsymbol{c}$, so the internal texture generated by InfoGAN is not good and the external shapes are similar. This poor diversity is a result of the variable $\boldsymbol{c}$, as it contains interpretable information on the data to control generated result.

EBGAN interprets GAN from the view of energy. Generated results show that this model can learn the probability distribution of images, but with a low convergence rate. When other models have been able to roughly express the outline of images, samples that EBGAN produced are still disorganized. But BEGAN generates rich and diverse results with the sharpest edges, because its discriminator draws lessons from EBGAN and its generator refers to the definition of WGAN loss. BEGAN also proposes a hyperparameter which can measure the diversity of generated samples to balance D and G and stabilize the training process [8]. LSGAN generates high quality images, dues to the replacement of cross-entropy loss of objective function by the least squares loss, which partly solves two defects of low quality and instability of training process. Generally, images generated by DCGAN are more diverse, especially inside textures and details.

3) QUANTITATIVE COMPARISONS

The above results are judged by human visual observation, which is only one aspect of evaluating GAN models. Comprehensive evaluations of the performance of GANs are being investigated. However, no single indicator could fully evaluate GAN, because which indicator to use depends on what researchers want to do with GAN. Quantitative indicators commonly used to automatically estimate the GAN include inception score (IS) [43] and frechet inception distance (FID) [42]. IS offers a method to evaluate the quality of generated examples. When calculating FID, the real data and the generated data are firstly embedded into a feature space through Inception Net. Then, the embedding layers are treated as a continuous multivariate, and the mean and covariance of real data and generated data are estimated to obtain FID [42]. IS has a good correlation with scores of human annotators, but with one disadvantage of insensitivity to the prior distribution of labels. FID is considered to be the best evaluation criteria currently, because it is robust in pattern dropping and coding network selection. These indicators are still an ongoing important research area, if GAN is used to generate high-quality examples, and thus requires human raters to evaluate the texture, details, diversity and rationality of generated examples; and if GAN is used for semi-supervised learning, and the accuracy of test set needs to be used as the evaluation criteria.

Table 6 shows different models measured from the time required for each model to run 40 iterations on both data sets. And the FID indicators calculated on 32000 generated and real examples. The time index can measure the number of parameters and rate of convergence. In experiments, VAE, CVAE, EBGAN and BEGAN spend less time and have faster convergence rate. From Table 6, it can be observed that VAE and CVAE show better performance on time consumption. This is because an encoder or decoder is employed in the generator of discriminator. EBGAN and BEGAN use automatic encoders for their discriminators, and both of them have fewer parameters than others that use neural networks. EBGAN treats the discriminator as an energy function. It has a smaller energy value when near the real data region and a higher energy value in other regions. Therefore, EBGAN interprets GAN from the perspective of energy. Its generator aims to produce samples with minimal energy, while its discriminator tries to assign higher energy to these generated samples. In this way, GAN can be trained with more extensive structures and loss functions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

The discriminator of BEGAN borrows structure from EBGAN, and the generator draws lessons from Wasserstein GAN to define loss, which derives from Wasserstein distance to match the self-encoding loss distribution. BEGAN also introduces a hyperparameter γ, the ratio between expected loss of generated samples and real ones, to measure the diversity of generated samples. This hyperparameter γ balances D and G to stabilize training process. If the generator performs too well, then the discriminator is focused. This hyperparameter also provides a measurable indicator for judging the convergence rate and the quality of images, resulting in a bit slower training speed than that of EBGAN.

Retaining the same conditions, the convergence speed ranking of other models from fast to slow is DCGAN, LSGAN, WGAN, CGAN, DRAGAN, WGAN-GP, InfoGAN and ACGAN. DCGAN is a benchmark model, other models add different constraints or improvements on it, for example, WGAN uses weight pruning to reduce the number of parameters to converge quickly. Its speed of convergence is directly proportional to the quality of images. In addition, the above typical GAN models have been quantitatively evaluated by FID, which is consistent with human judgment and more robust to noise. There is a negative correlation between FID and the visual quality of generated samples. However, one disadvantage of the indicator is that over-fitting cannot be detected, in other words, when GAN stores all training samples, the FID will still score perfectly [66].

As shown in Table 4, table 5 and Table 6, there is a trade-off between the quality of generated samples and the time consuming. For example, WGAN-GP generates higher quality samples but takes more time. When a model shows good performance in one aspect, it may not be able to perform well in another. For example, WGAN-GP has good robustness and can produce high quality samples, but its convergence rate is relatively slow. Therefore, basic theories of GAN to develop more stable and easily trained models are in demand. For example, a combination of global and local regularization in generators and discriminators could be used to capture more features. The latest Self-Attention generative adversarial network (SAGAN) [35] has unified attention mechanism and spectral normalization and achieved the best results. Currently, there is still a lack of a systematic evaluation system that can evaluate each model fairly and neutrally. Therefore, if you intend to use GAN, you need to choose appropriate model according to your purpose.

IV. APPLICATIONS OF GAN IN COMPUTER VISION

Computer vision is a simulation of biological vision using computers and related devices. It seeks to automate tasks that human visual system can do and deals with how computers can be made for gaining high-level understanding from digital images or videos [68]. GAN has performed exceptionally well in many fields of computer vision. Due to characteristics of adversarial mechanism and constant self-improvement, GAN stands out at learning features from existing distributions and capturing reasonable visual characteristics. From image generation to a series of applications, more and more new GAN models have been proposed, and they have produced significant results than traditional methods in different computer vision fields. In this section, we will introduce several representative cutting-edge applications of GAN, including high quality samples generation, style transfer and image translation, text-image mutual generation, image in painting and restoration and others.

A. GENERATE HIGH QUALITY SAMPLES

The most extensive application of GAN as a generative model is sample generation. GAN learns the distribution of real data in ways of supervised learning, semi-supervised learning or unsupervised learning [20], [69]. Compared with traditional machine learning algorithms that design a certain functional expression, GAN works in a way of end-to-end. GAN learns feature distributions or mappings of real data and generates new samples through artificial neural networks. The most basic GAN application is to imitate distributions of real samples and to generate the same samples. For example, GAN is trained on MNIST and generates new handwritten numbers [70]. The quality of generated samples is one of the indicators to measure a model. Several models that can generate high quality images would be introduced in this section.

Researchers have been devoting themselves to make generated image look like a real one, and they have developed some successful models, such as DCGAN, WGAN and Laplacian Pyramid of Adversarial Networks (LAPGAN) [45]. DCGAN combines GAN with deep CNN and adds constraints to stabilize the model, and it has obtained inspiring results on several data sets. In addition, its generator can perform interesting vector arithmetic, proving that generated pictures are not the memory of picture elements in database, but that these pictures are drawn by particular filters. Currently, DCGAN is the model with the highest usage rate and is suitable for most generation tasks. However, one of its disadvantages is that the resolution of generated sample is low. Since the number of images in general datasets is large, the pixel of images is low. For instance, ImageNet contains tens of millions images of low resolution. It can satisfy the requirement of training a model, but the generated images will not be clear. Then, how to improve the quality of generated images with the dataset has become a focus of research.

The LAPGAN proposed in literature [45] is a tandem network. A set of images is arranged hierarchically according to their resolution from low to high. Based on a low-resolution sample, LAPGAN first generates a low-resolution image, which will be input together with another higher resolution image to the next phase. The generator of each phase corresponds to a discriminator which judges whether input image is fake or real. The model is shown in Fig. 5. One of the advantages of LAPGAN is that the generators in each stage can learn different distributions and pass them to next layer as supplementary information. After several times of feature extraction, the resolution of the final generated image will be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

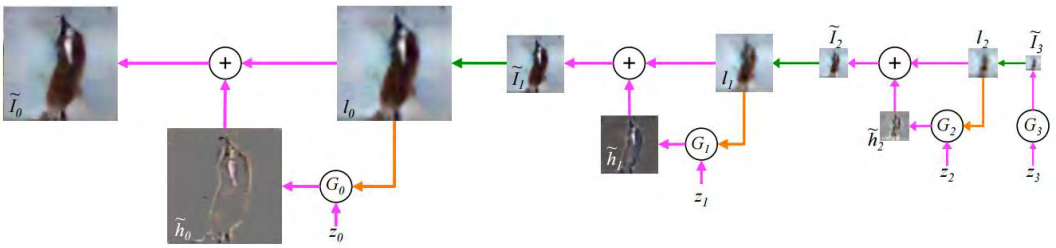


FIGURE 5. The process of generating samples of LAPGAN, which is divided into multiple stages, a short period of output is used as the input for next stage [45].

TABLE 7. Comparisons of different models used in generating high quality examples.

GAN model	Improvements	Advantages	Disadvantages	Scenarios
DCGAN [20]	Combined with CNN; stride convolution, fractional-strided convolutions, batchnorm, ReLU, LeakyReLU.	More stable, easy convergence, generate diverse samples.	Training different data needs to adjust parameters, model collapse.	Suitable for most scenarios, is one of the highest usage models.
LAPGAN [45]	Laplacian Pyramids with up sample;Gaussian Pyramids with down sample; Cascaded Convolutional.	Easy to approach and learn residuals; step-by-step independent training; increase ability of GAN.	Must be trained under supervision.	Scenes that need to generate high resolution images.
SAGAN [35]	Introduced self-attention mechanism, spectral normalization and two-timescale update rule(TTUR).	Stable and speed up training, generate realistic images.	Attention is not extended to larger (k $\times$ k) convolution.	Large-scale classification of conditional image generation tasks.

greatly improved and more authentic. Apart from adopting the above methods, LAPGAN also incorporates CGAN to transform unsupervised approaches into supervised learning with significant efficiency improvements. Since LAPGAN must be trained under supervised learning, it is more suitable for scenarios where high-resolution images need to be generated.

Then, Self-Attention generative adversarial network (SAGAN) is proposed in the literature [35], which adopts self-attention mechanism to improve the quality of generated images. Unlike traditional convolution GANs, which focus on local features of images and are difficult to capture geometric patterns or structural patterns, SAGAN allows attention-driven, long-range dependency modeling for image generation tasks and uses clues from all feature locations to generate details. In addition, the generator of SAGAN also adopts the spectral normalization which was previously only used in the discriminator to enhance its adjustment. In addition, it uses two-timescale update rule (TTUR) to accelerate the training of the regularization discriminator. SAGAN has greatly improved the training dynamics of GAN as well as the quality of generated images. It is compared with DCGAN, LAPGAN in Table 7.

B. STYLE TRANSFER AND IMAGE TRANSLATION

Style transfer is another interesting application of GAN, it also called image translation, which transforms images from one style to another. Traditional methods of image generation can only solve one certain task, such as converting an image into a corresponding semantic label map [71], or translating the outline into a real image [72], and it requires different systems for each task. GAN can solve different tasks of style transfer and image translation, because it provides a unified framework for different tasks by adversarial training.

One typical GAN based style transfer model is pix2pix, which is a one-to-one image style migration model [4]. Pix2pix uses two datasets A and B, one being the collection of images of one style, and the other being the collection of the same set of images but in another style. For example, dataset A is outlines of shoes, and correspondingly dataset B is real shoe images. In training, one dataset is used as input, and the other is used as a conditional input, also called target. Pix2pix learns mappings between the two datasets and generates images. The errors between generated images and targets are calculated by loss function, which further adjusts the parameters to generate better images that are as similar as the target image. To make generated images more authentic, pix2pix has been optimized in the following ways: firstly, its generator uses U-NET architecture, which adds skip connections between each layer i and $(n-i)$ (i.e. n is the total number of layers) [73]. In style transfer, normal convolutional and pooling operations are replaced by U-NET while many pixel permutations remain fixed, then images could be passed directly to the next layer in skip connections to ensure the content of images unchanged. Secondly, its discriminator uses PatchGAN architecture, which has been proved effective in classifying effect with a local classifer rather than a global one, to pay attention to local image. PatchGAN also reduces

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

the number of parameters, which improves the speed and efficiency of training. Pix2pix model is sufficiently trained to achieve realistic artistic style transitions between pairs of images, such as styles transfer of different maps, objects and their contour maps. However, the model also has its disadvantage, to be specific, it requires a one-to-one paired data set.

Another method based on GAN is employed by Kim *et al.* [47] to discover cross-domain relationships, known as DiscoGAN. With the discovered relationship, DiscoGAN retains the main features of images while successfully transferring the style of one domain to another and withhold key attributes, such as the changeover of apple and orange, cat and dog, etc. DiscoGAN improves the quality of generated image by implementing one-to-one bidirectional mappings, which requires one-to-one paired data sets, therefore, DiscoGAN also has limitations as pix2pix. On the other hand, unsupervised cross-domain image generation can be obtained by the domain transfer network (DTN) [48]. DTN employs composite loss functions which include several GAN losses and specification components to produce a convincing new image and to maintain original identities of the entities. For example, it generates visual appealing facial emoticons symbols that capture more facial features than human-created, and transfers photos to emojis (Fig. 6 Shown). Therefore, DTN is suitable for generating anime images from photographs of people. However, due to the asymmetry of input function and lower information content in new source domain, the results produced are less attractive.

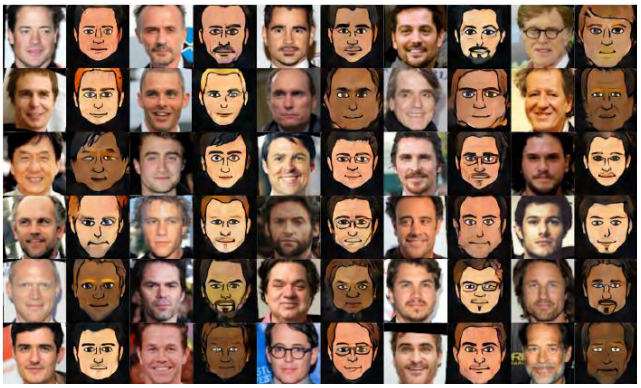


FIGURE 6. Transfer photos to emoji by DTN model [48].

To break the constraints of paired data sets, the cycle-consistent adversarial network (CycleGAN) proposed by literature [46] introduces cycle loss function, whose core idea is to generate samples for twice. CycleGAN consists of two-step transformations to realize self-constraint. Firstly, it maps original images to the target domain, then returns first-generation images to original domain to get second-generation images and eliminates requirements for matching images in target domain. The *G* network maps initial images to target domain through the matching generator and discriminator to improve the quality of generated images. It can be assumed that the first-generated output images are reasonable when the secondary generated samples are the same with the original images. However, CycleGAN has its limit due to strict constraints of the model. For instance, when two datasets to be converted differ greatly, the generated images are not alike; in other words, it tends to perform better when the two datasets are similar. CycleGAN can be applied to many aspects, such as painting style conversion, seasonal migration, two-dimensional drawings to three-dimensional image conversion, and the conversion of historical celebrity images to real people and so on. However, the quality of its generated image is lower than that of pix2pix, since the latter is easier to learn the exact mappings than the former.

Sem-GAN proposes a semantically consistent GAN framework in which semantics are defined by the class identities of image fragments generated by semantic segmentation algorithms in the source domain [26]. Sem-GAN combines GAN loss and cycle constraints to make the images generated from source images inherit the appearances of the target domain, and it also introduces semantic loss to improve consistency loss. Sem-GAN is suitable to handle unpaired image-to-image transformations, but requires datasets to be semantically identified.

GAN is good at image style transfer, since two networks of GAN are able to check and balance each other. Five GAN based models are compared and contrasted in Table 8 in terms of their advantages, disadvantages and usage scenarios in style transfer and image translation. Based on GAN, it is convenient to realize mutual conversion between two styles or to create some works with a certain artistic style. The future directions of GAN in image style transfer and image translation are to generate more accurate and desired images through semantic control and to realize the real-time interaction with the model. For instance, StarGAN represents the generation of semantically controlled multi-domain transformations [74], while most current models learn a fixed mapping, and generate results randomly and cannot be intervened during the generation process. In addition, another future direction is to add more constraints to the model to increase the quality of generated images in the absence of constraints on the datasets.

C. TEXT-IMAGE MUTUAL GENERATION

The most common applications of GAN are to generate images on image datasets. For example, GAN is trained to generate handwritten numerals and face images that cannot be differentiated with real ones. All these applications learn features and generate distributions directly. A creative application is to generate corresponding images based on input text. It is more difficult than simply generating an image, because it does not only involve semantic understanding of texts, but is also a many-to-many task. Compared with traditional approaches that describe an object by attributes and encode the features into corresponding vectors, but difficult to obtain a large number of tags [75], [77]. GAN can learn the feature mappings between characters and image pixels

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

TABLE 8. Comparison of different GAN models used in style transfer and image translation.

GAN model	Improvements	Advantages	Disadvantages	Scenarios
pix2pix [4]	CGAN; generator using U-NET architecture; discriminator using PatchGAN classifier, common framework.	Large-scale reduction of parameters, improve training speed and efficiency; generated images are realistic.	Data sets must be a one-to-one paired images.	The style transfer between pairs of images, such as different styles of maps, objects and its contours, etc.
CycleGAN [46]	Cycle loss, self-constraint. The two-step transformation between original domain images and target domain.	Low data set requirements; ability to convert two image styles randomly.	The quality of generated target image is lower than pix2pix.	Most of the style conversion scenes, such as seasonal transfer, migration of artwork styles, etc.
DiscoGAN [47]	Two GANs couple together to discover cross-domain relationships; style transfer and retaining key attributes.	Implement bidirectional one-to-one mapping to avoid model collapse and improve image quality.	Data sets must be one-to-one paired images.	Inter-generation between different domains, such as gender conversion, bags and shoes, cars and chairs, etc.
DTN [48]	The generator contains a learning function, it integrates several complex loss functions to generate emoji from a facial image.	Create visually appealing facial emoticons and capture more facial features than human-created emoticons.	Due to the asymmetry input function and lower information content, the result is less attractive.	Create a cartoon image from real photos.
Sem-GAN [26]	Introduced semantic loss, combined GAN loss and cycle constraints.	Ensures semantic consistency and realizes better translation.	Requires datasets to be semantically identified.	Handle unpaired image-to-image transformations.

FIGURE 7. StackGAN generates an image in two phases [44].

directly and can process large quantity of data to generate new visual images.

Reed [75] proposes a model that combine the deep convolutional structure and GAN, which bridge text and images to transform visual concepts from characters to pixels. It can generate natural pictures of birds and flowers from a detailed text description. The implementations of this model are as follows: Gaussian noise is embedded in G network together with the input of text descriptions to learn mappings and to generate images as described by the texts. Generated images, real images and corresponding descriptions and false descriptions of real image are entered together into D. Through training with these pictures and descriptions, the discriminator's ability is increasing, and accordingly it drives the generator to produce more authentic pictures. By producing simple objects with some simple descriptive words, such as ''flowers with overlapping pink pointy petals and surrounded by yellow stamens'', this model [75] realizes image generation from text descriptions to images and generates authentic image representations in simple scenes. It is suitable for simple text-to-image generation, such as flower generation and bird generation. The disadvantage of this model is that it works well only for simple descriptions of image generation, and fails to generate clear images for complex descriptions, unless with extensive training.

A better improvement is StackGAN [44] which is different from general GAN models. StackGAN takes two stages to generate images. In the first stage, generated images are rough, while in the second stage, higher resolution images are generated. If needed, more stages can be added to generate images with rich details and delicate textures. GAN can generate higher quality images after changing structure or adding more constraints. For example, the improved StackGAN-v2 model is a multi-stage generative adversarial network architecture. It contains multiple generators and discriminators arranged in a tree-like structure [78]. Fig. 7 shows generated images described by ''this flower has white petals with a yellow tip and a yellow pistil''. The upper row images are generated in the first stage and the bottom row is the results of the second stage. Different stages of image generation improve the resolution of images generated. Despite the advantage of clear image generation from texts, it is possible that StackGAN will fail generating sample because it takes multiple phases to finish complex generation tasks, and each phase may not be clear about what to do.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

TABLE 9. Comparison of different GAN models used in text and image transforming to each other.

GAN model	Improvements	Advantages	Disadvantages	Scenarios
Literature [75]	Learn mappings between text and picture; interpolation; inverse style migration analysis.	Realize generation from text descriptions to images; Generate realistic image.	Only a simple description can be generated. If the description is complex, the generated image is not clear.	Simple text-to-image generation, for example, flower and bird generation.
StackGAN [44]	It includes two GANs, first generates a relatively coarse image, then corrects previous image and adds details.	The final image sharpness is improved by phasing generation.	Generation task contains two stages, it may result in failure of each task to find the focus, then the whole task is failed.	Used for generating text to clear images.
GAN-CLS [76]	Revised the objective function of original GAN-CLS, judge whether text and image match.	Efficient, could generate images based on the given text in two data sets.	Generated results do not have clearly boundaries, sensitive to initialization of hyperparameters and parameters.	Generating images from corresponding text descriptions.
RTT-GAN [50]	Sentences are generated sequentially by introducing attention mechanisms in each step.	To generate paragraphs and to synthesize semantic coherent paragraphs.	Unable to work under unsupervised learning.	Let the machine look like a human to read and write articles.

The modified GAN-CLS model is proposed by Gong and Xia [76] to generate images from corresponding text descriptions. The modified GAN-CLS revises the objective function of original GAN-CLS [75] to make the discriminator matching sensing ability. In other words, the discriminator can judge whether the input text and the image match with each other. In addition, a pre-trained deep convolutional-recurrent text encoder is used to encode the text. The modified model is very efficient and can generate corresponding images based on a given text in two datasets. However, this model is sensitive to the initialization of hyperparameters and parameters, and in some cases, the generated results do not have clear boundaries.

It is easier to generate textual descriptions for given images than to generate images from texts. The computer has already been able to describe image content, but how to make a computer speak and write articles like a human being remains a question. The literature [50] proposes a semi-supervised generative framework Recurrent Topic-Transition GAN (RTT-GAN) for paragraph generation. RTT-GAN constructs a confrontational framework between a structured paragraph generator and a multi-level paragraph discriminator. Paragraph generator synthesizes sentences sequentially by introducing regional-based visual and verbal attention mechanisms at each step. RTT-GAN reasons over local semantic regions and exploits linguistic knowledge to synthesize diverse and semantic descriptions of paragraphs. It realizes machine speaking and writing as human beings do, on the condition of supervised learning. Table 9 shows comparisons for mutual generation of text and image.

D. IMAGE INPAINTING AND RESTORATION

Image inpainting and restoration are processes of image reconstruction, which restores an incomplete image to a complete one, or obtains a global image from a local image. For example, local or non-local information are often used to recover images in traditional methods. Local approaches rely on prior distribution of input image [82], but it is less effective when the missing content is different from the surrounding region. Non-local approaches predict missing region by training massive complete images [83], but it is difficult to repair images when the missing image is not in the training set. Unlike traditional methods, GAN reconstructs an image by generating a similar missing image and finding the closest vector.

At present, the results of face recognition are getting more and more accurate. Different face recognition algorithms have been applied in crowded areas such as subways, train stations and airports. However, it is still a bit difficult to detect each pedestrian accurately in such dense crowds. At the same time, people appear with various forms and expressions, especially when only one side or part of them appearing in lens, they will not be identified by the existing face recognition techniques. How to use scientific and technological methods to obtain overall information from the locality is an urgent issue that needs to be solved. The literature [27] proposes a two-pathway generative adversarial network (TP-GAN) inspired by human visual recognition process to quickly identify human faces. TP-GAN combines global structure and local details to generate photo-like images and retains original identity features of people. It can synthesize an image of a front face based on the information of partial faces shot from different viewpoints or under different lightening conditions, or with different postures. Fig. 2 shows frontal images synthesized from different angles. To get these images, TP-GAN has made the following changes: it has two paths for the generator; it has one global network focusing on processing global structure and four landmark located patch networks attending to local textures around four facial landmarks to obtain two feature maps; then the two feature maps are merged together for final synthesis [27]. Not only the synthetic front view and real photo are input to the discriminator, but also the distribution information of front faces are incorporated to GAN, thus restoring the process under a very good constraint. In addition, TP-GAN combines multiple types of loss functions to synthesize missing

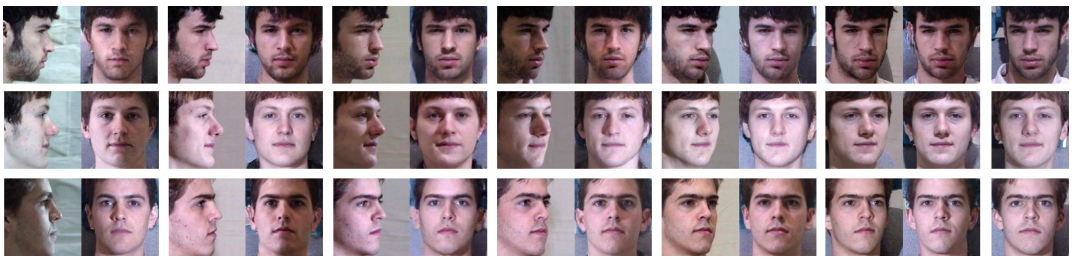


FIGURE 8. TP-GAN synthetizes frontal images from different angles [27].

parts to preserve facial features. Therefore, the synthesized images are authentic and have well-preserved identity features. Trough this way, a large number of different postures could be dealt with. Furthermore, TP-GAN can be applied to face analysis or in scenes that require identification of identity information through side faces, such as airport, train station and so on. However, when the rotation angle is too large, the generated facial details are different real photos.

GAN has also been used in the field of image restoration. Literature [79] proposes a novel method for semantic image restoration. It generates missing content by adjusting available data. It search the closest encoding of corrupted images in latent images with the context and previous losses, then the encoding is used to infer missing contents. This method has successfully predicted a large amount of missing region information and achieved pixel-level fidelity, and it is used to recover the occlusion area and generate missing content. However, in the case of large area loss, the generated results are not authentic.

Literature [80] proposes an unsupervised feature learning algorithm driven by a context-based pixel prediction. A context encoder is proposed. It is a pre-trained convolutional neural network that can generate contents of any image area with its surroundings. When a context encoder is trained, usages of standard pixel reconstruction loss and adversarial loss can complement images and produce clearer results. This model [80] indicates that when the context encoder learns to acquire features, it does not only capture characteristics of appearance but also captures semantics of visual structure, so that it can also be used for semantic repair tasks. It is suitable for unsupervised visual feature learning, as outer conditions may influence its results. Therefore, it can generate better results than in supervised training.

Li *et al.* [81] proposes a face completion algorithm using a depth generative model. The algorithm is based on neural network to directly generate contents of missing regions. It uses reconstruction loss as well as the combination of two adversarial losses and semantic parsing loss to train the model to ensure the consistency of pixel loyalty and local global content. It is able to handle large areas of missing pixels of any shape and produce realistic faces. It can also generate the missing area directly. The problem of this algorithm is that it uses too many loss functions, and each loss function contains some parameter settings; once the loss function is not properly chose, the results of training may not be satisfactory.

PGGAN is an image restoration model that combines the global GAN (G-GAN) architecture and the patchGAN method to construct a discriminator network to capture global and local information [49]. The discriminator of PGGAN has two paths, which share a weighting architecture at first few layers, where they learn common low-level visual functions and separate after a certain layer. The first path decides whether the output image is real, and the other evaluates local details. PGGAN combines reconstruction loss, adversarial loss and joint loss and has made considerable progress in both visual and quantitative assessments. It is suitable for repairing high resolution images.

Table 10 shows comparisons of different models for image restoration. Compared with other generative models, GAN is more flexible and effective. The generative models need to be further developed to complete inpainting. In contrast, the superior performance of GAN in modeling of two-dimensional data distribution has solved many unreasonable low-level visual problems. GAN makes generated images more authentic by cooperating with encoders, CNN, contextual semantics, and a combination of multiple losses.

E. OTHER APPLICATIONS OF GAN TO COMPUTER VISION

In addition to the applications mentioned above, GAN has also shown great potential in other areas of computer vision [84], [85]. For instance, GAN combines simulated and unsupervised algorithms to generate synthetic images as samples for training, which is also a promising direction [86]. Mathieu [87] proposes a GAN network for video prediction. It can reasonably predict next frame of scene with spatio-temporal convolutional architecture by distinguishing the scene's foreground from background [88]. The motion and content GAN (MoCoGAN) maps random noise vectors to video frames one by one, and generates video clips to achieve future frame predictions [51]. GAN has achieved great success in super resolution, which is to promote low resolution images to higher resolution [89], [90]. GAN is also used for road detection [91] and object detection. For example, perceptual generative adversarial network [92] is for small object detection by reducing the representation difference between small and large objects. GAN can learn

TABLE 10. Comparison of GAN models used in image inpainting and restoration.

GAN model	Improvements	Advantages	Disadvantages	Scenarios
TP-GAN [27]	Combining data distribution with human face, two-path, one focusing on global structure, the other on local texture.	The synthetic images of front face view are realistic and well preserved, handle a large number of different positions.	Too large angle of rotation is, too different details of generated faces from the real pictures.	Face analysis or face identification looking for suspects, etc.
Literature [79]	Use contexts and previous losses to search for closest encoding of corrupted image.	No need camouflage training, produce sharper images, achieve pixel-level fidelity.	The result is not realistic when loss area is large.	Generate missing content, such as recovering occluded parts.
Literature [80]	A context encoder; pixel reconstruction loss, adversarial loss, generate clearer results.	Context encoder captures the appearance, semantics and used for semantic repair tasks.	Because of the unsupervised training, the results are not realistic as the supervision.	Unsupervised visual feature learning driven by pixel prediction for context
Literature [81]	Consistency of pixel and local global content, combination of two adversarial losses and semantic parse losses.	Process large-area missing pixels of any shape, produce realistic facial finishes.	Too many loss functions, difficult to select the weight of each loss.	Facial complement, it can directly generate contents of the missing region.
PGGAN [49]	Combine global GAN with patchGAN, two paths shared a weighting architecture at first layers.	Made considerable progress in both visual and quantitative assessments.	The inpainting problem that is tightly coupled to the generative modeling is need progress.	Repairing high resolution images.

potential probability of object shape through 3D modeling, and translate real images to new 3D views by using a generative network [93], [94]. It can also detect multi-spectral image changes [95], generate realistic results in limited training data [96], and generate time series such as music waveform [97], ICU record in intensive care units [98], electronic health records [99], and be applied in medical image segmentation and so on [100], [101].

V. CONCLUSION AND PROSPECT

A. CONCLUSION

In this paper, we reviewed GAN and its typical applications in computer vision. Both state-of-the-art and classical GAN models have been evaluated in detail from the perspectives of principles adopted, visual results of generated examples and so on. GAN is not only novel in algorithm, but also capable of achieving good results in practice. Furthermore, GAN provides a better solution for solving problems of insufficient samples, poor quality of generations and difficulties in extracting features as a generative model. We have showed the ability of different GAN models, and generated samples from theoretical and experimental results. However, the experiments also show a complete scientific evaluation system is absent to evaluate the model objectively, comprehensively and fairly. We have summarized advantages, disadvantages and scenarios of different models to provide suggestions for further improvement. In addition, we have analyzed those applications of GAN in different fields that have achieved remarkable achievements in computer vision, and proposed solutions to the problems in each field. In a word, GAN is an inclusive framework that can be combined with many deep learning models to solve problems that traditional machine learning algorithms cannot solve.

B. PROSPECT

To develop faster and better GAN, it is necessary to do the following: 1) Improve the GAN theory, which must be provable and guaranteed. For example, to look for a suitable distance, which can work well under various conditions to replace JS divergence. Although Wasserstein distance can measure the distance when two distributions are not overlapping, certain constraints must be met, otherwise gradient disappearance or explosion will still occur. 2) Find a more suitable evaluative indicator and prove why it is the right indicator. Commonly used indicators such as IS and FID can only reflect the performance of GAN in some aspects. Since GAN supervises itself, its boundary is unclear, then it is difficult to evaluate it comprehensively. 3) More killer application scenarios of GAN need to be found. At present, GAN has not been applied to a certain scene on a large scale.

Furthermore, GAN has a more extensive application prospect when combined with other machine learning algorithms. It is expected to make progresses in the following areas:

1) Theory breakthroughs. The incompleteness of GAN basic theory is a barrier for GAN models to produce high quality generated examples. Therefore, the most important direction for future research is to make breakthroughs in theoretical aspects to solve problems such as non-convergence, model collapse and training difficulties [43]. In spite of some commonly improved methods such as weights pruning [25], weights regularization [31], [34], [35], new loss functions [32] and Nash equilibrium [33], further improvement is still necessary.

2) Algorithm evolutions. GAN can entail the latest theories and research results in machine learning, for instance, attention mechanism can be introduced into GAN to capture global features [35]. GAN and adversarial samples are applied to solve security problems of deep learning systems [102]. GAN works with reinforcement learning to solve weaknesses in dealing with discrete variables, where policy gradient algorithms of reinforcement learning are used, so that GAN could work in discrete scenarios to further widen its scope of application [103].

3) Performance evaluations. As a new generative model, GAN has no relevant indicators that can evaluate different models from their performance, accuracy, over-fitting degree, and visual quality of generated samples and other aspects comprehensively. Therefore, a scientific and uniform performance evolution standard needs to be developed. It is urgent to establish a standardized and universal scientific evaluation system [66].

4) Special killer applications. Transform the current solution to one type of problem to one specific practical application problem and develop killer applications based on existing problems. That is, GAN should solve more specific application problems, such as a system of specific scene generation [104] or a system that enhances the resolution of a specific part of images [90]. It can also generate high-quality visual scenes of complete game scenes and characters through combination with the game system [105].

5) Cross applications. The cross-integration of GAN with certain special healthcare industries facilitates the generation of hard-to-obtain sample data to complement real data. For example, to generate more medical samples with available data when datasets on medical sciences are not sufficient [98], [99].

In conclusion, for the long-term development of artificial intelligence, using GAN to enhance the abilities of machines to understand the world and let machines have ''awareness'' is a question worth studying.

REFERENCES

[1] I. J. Goodfellow *et al.*, ''Generative adversarial nets,'' in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[2] S. Liu *et al.*, ''Face aging with contextual generative adversarial nets,'' in *Proc. ACM*, 2017, pp. 82–90.

[3] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, ''A novel image steganography method via deep convolutional generative adversarial networks,'' *IEEE Access*, vol. 6, pp. 38303–38314, 2018.

[4] P. Isola *et al.*, ''Image-to-image translation with conditional adversarial networks,'' in *Proc. CVPR*, Jul. 2017, pp. 5967–5976.

[5] C. Wang, C. Xu, C. Wang, and D. Tao, ''Perceptual adversarial networks for image-to-image transformation,'' *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4066–4079, Aug. 2018.

[6] A. Odena, C. Olah, and J. Shlens, ''Conditional image synthesis with auxiliary classifier GANs,'' in *Proc. 34th Int. Conf. Mach. Learn. (PMLR)*, vol. 70, 2017, pp. 2642–2651. [Online]. Available: https://arxiv.org/abs/1610.09585

[7] Y. Zhan, D. Hu, Y. Wang, and X. Yu, ''Semisupervised hyperspectral image classification based on generative adversarial networks,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 212–216, Feb. 2018.

[8] D. Berthelot, T. Schumm, and L. Metz. (2017). ''BEGAN: Boundary equilibrium generative adversarial networks.'' [Online]. Available: http://arxiv.org/abs/1703.10717

[9] K. Kim and H. Myung, ''Autoencoder-combined generative adversarial networks for synthetic image data generation and detection of jellyfish swarm,'' *IEEE Access*, vol. 6, pp. 54207–54214, 2018.

[10] N. Li, Z. Zheng, S. Zhang, Z. Yu, H. Zheng, and B. Zheng, ''The synthesis of unpaired underwater images using a multistyle generative adversarial network,'' *IEEE Access*, vol. 6, pp. 54241–54257, 2018.

[11] R. Salakhutdinov and G. Hinton, ''Deep Boltzmann machines,'' *J. Mach. Learn. Res.*, vol. 5, no. 2, pp. 1967–2006, 2009.

[12] C. E. Rasmussen, ''The infinite Gaussian mixture model,'' in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 554–560.

[13] L. Jiang, H. Zhang, and Z. Cai, ''A novel Bayes model: Hidden naive Bayes,'' *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 10, pp. 1361–1371, Oct. 2009.

[14] L. R. Rabiner, ''A tutorial on hidden Markov models and selected applications in speech recognition,'' *Proc. IEEE*, vol. 77, no. 2, pp. 267–296, Feb. 1989.

[15] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, ''ImageNet: A large-scale hierarchical image database,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[16] Y. Pu *et al.*, ''Variational autoencoder for deep learning of images, labels and captions,'' in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2016, pp. 2352–2360. [Online]. Available: https://arxiv.org/pdf/1609.08976

[17] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, ''Generative adversarial networks: An overview,'' *IEEE Signal Process.*, vol. 35, no. 1, pp. 53–65, Jan. 2017.

[18] Y. LeCun *et al.*, ''Backpropagation applied to handwritten zip code recognition,'' *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ''Gradient-based learning applied to document recognition,'' *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[20] A. Radford, L. Metz, and S. Chintala, ''Unsupervised representation learning with deep convolutional generative adversarial networks,'' *Comput. Sci.*, vol. abs/1511.06434, Nov. 2015. [Online]. Available: https://arxiv.org/pdf/1511.06434

[21] A. Graves, S. Fernández, and J. Schmidhuber, ''Multi-dimensional recurrent neural networks,'' in *Proc. Int. Conf. Artif. Neural Netw.*, 2007, pp. 549–558.

[22] O. Mogren, ''C-RNN-GAN: Continuous recurrent neural networks with adversarial training,'' in *Proc. Conf. Workshop Neural Inf. Process. Syst. (NIPS)*, Nov. 2016. [Online]. Available: https://arxiv.org/abs/1611.09904

[23] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, ''Generating images with recurrent adversarial networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Feb. 2016. [Online]. Available: https://arxiv.org/abs/1602.05110

[25] M. Arjovsky, S. Chintala, and L. Bottou. (2017). ''Wasserstein GAN.'' [Online]. Available: https://arxiv.org/abs/1701.07875

[26] A. Cherian and A. Sullivan, ''Sem-GAN: Semantically-consistent image-to-image translation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2018. [Online]. Available: https://arxiv.org/abs/1807.04409

[27] R. Huang, S. Zhang, T. Li, and R. He, ''Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2458–2467.

[28] M. Mirza and S. Osindero, ''Conditional generative adversarial nets,'' *Comput. Sci.*, pp. 2672–2680, Nov. 2014.

[29] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, ''InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,'' in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180. [Online]. Available: https://arxiv.org/pdf/1606.03657

[30] J. Zhao, M. Mathieu, and Y. Lecun, ''Energy-based generative adversarial network,'' in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Sep. 2016. [Online]. Available: https://arxiv.org/pdf/1609.03126

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, ''Improved training of Wasserstein GANs,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777. [Online]. Available: https://arxiv.org/pdf/1606.0349

[32] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, ''Least squares generative adversarial networks,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2018, pp. 2813–2821.

[33] N. Kodali, J. Abernethy, J. Hays, and Z. Kira. (2018). ''On convergence and stability of GANs.'' [Online]. Available: https://arxiv.org/abs/1705.07215

[34] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, ''Spectral normalization for generative adversarial networks,'' in *Proc. ICML Workshop Implicit Models*, Feb. 2017. [Online]. Available: https://arxiv.org/abs/1802.05957

[35] H. Zhang, I. J. Goodfellow, D. Metaxas, and A. Odena. (2018). ''Self-attention generative adversarial networks.'' [Online]. Available: https://arxiv.org/abs/1805.08318

[36] W. Nie and A. Patel. (2018). ''JR-GAN: Jacobian regularization for generative adversarial networks.'' [Online]. Available: https://arxiv.org/abs/1806.09235

[37] R. Saqur and S. Vivona, "CapsGAN: Using dynamic routing for generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018. [Online]. Available: https://arxiv.org/abs/1806.03968

[38] J. Adler and S. Lunz, "Banach Wasserstein GAN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018. [Online]. Available: https://arxiv.org/abs/1806.06621

[39] G. Zhong, W. Gao, Y. Liu, and Y. Yang, "Generative adversarial networks with decoder-encoder output noise," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2018. [Online]. Available: https://arxiv.org/abs/1807.03923

[40] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Conf. Int. Conf. Learn. Represent. (ICLR)*, Dec. 2014. [Online]. Available: https://arxiv.org/pdf/1312.6114

[41] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 4, 2014, pp. 3581–3589.

[42] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637. [Online]. Available: https://arxiv.org/abs/1706.08500

[43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. (2016). "Improved techniques for training GANs." [Online]. Available: https://arxiv.org/abs/1606.03498

[44] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2018, pp. 5908–6916.

[45] R. Fergus, R. Fergus, R. Fergus, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1486–1494.

[46] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2242–2251.

[47] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1857–1865. [Online]. Available: https://arxiv.org/abs/1703.05192

[48] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nov. 2016. [Online]. Available: https://arxiv.org/pdf/1611.02200

[49] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Mar. 2018. [Online]. Available: https://arxiv.org/abs/1803.07422

[50] X. Liang, Z. Hu, H. Zhang, C. Gan, and E. P. Xing, "Recurrent topic-transition GAN for visual paragraph generation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3362–3371.

[51] S. Tulyakov, M. Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017. [Online]. Available: https://arxiv.org/abs/1707.04993

[52] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[53] D. He, W. Chen, L. Wang, and T. Y. Liu, "A game-theoretic machine learning approach for revenue maximization in sponsored search," in *Proc. Int. Joint Conf. Artif. Intell.*, 2014, pp. 206–212.

[54] I. Goodfellow. (2016). "Nips 2016 tutorial: Generative adversarial networks." [Online]. Available: http://arxiv.org/abs/1701.00160

[55] M. Arjovsky and L. Bottou. (2017). "Towards principled methods for training generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1701.04862

[56] Y. Li, N. Xiao, and W. Ouyang, "Improved boundary equilibrium generative adversarial networks," *IEEE Access*, vol. 6, pp. 11342–11348, 2018.

[57] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2018–2025.

[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Comput. Sci.*, vol. 41, no. 7, pp. 1074–1080, 2014.

[60] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[61] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *Comput. Sci.*, May 2015. [Online]. Available: https://arxiv.org/pdf/1505.00853

[62] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 44–51.

[63] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 588–598, Sep. 2017.

[64] H. Zheng. (2017). *Amazing Wasserstein GAN*. [Online]. Available: https://zhuanlan.zhihu.com/p/25071913

[65] H. Lee. (2017). *TensorFlow-Generative-Model-Collections*. [Online]. Available: https://github.com/hwalsuklee/tensorflow-generative-model-collections

[66] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. (2017). "Are GANs created equal? A large-scale study." [Online]. Available: https://arxiv.org/abs/1711.10337

[67] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Aug. 2017. [Online]. Available: https://arxiv.org/pdf/1708.07747

[68] R. Szeliski, *Computer Vision: Algorithms and Applications*, vol. 21. New York, NY, USA: Springer-Verlag, 2011, pp. 2601–2605.

[69] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *Comput. Sci.*, Nov. 2015. [Online]. Available: https://arxiv.org/abs/1511.06390

[70] Z. Zheng, C. Wang, Z. Yu, H. Zheng, and B. Zheng, "Instance map based image synthesis with a denoising generative adversarial network," *IEEE Access*, vol. 6, pp. 33654–33665, 2018.

[71] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[72] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. SIGGRAPH*, 2001, pp. 341–346.

[73] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. 2015, pp. 234–241.

[74] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo. (2017). "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation." [Online]. Available: https://arxiv.org/abs/1711.09020

[75] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1060–1069.

[76] F. Gong and Z. Xia. (2018). "Generate the corresponding image from text description using modified GAN-CLS algorithm." [Online]. Available: https://arxiv.org/abs/1806.11302

[77] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1778–1785.

[78] H. Zhang *et al.*, "StackGAN++: Realistic image synthesis with stacked generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[79] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6882–6890.

[80] D. Pathak, P. Krähenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

[81] Y. Li, S. Liu, J. Yang, and M. H. Yang, "Generative face completion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5892–5900.

[82] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Sep. 1999, pp. 1033–1038.

[83] O. Whyte, J. Sivic, and A. Zisserman, "Get out of my picture! Internet-based inpainting," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, London, U.K., Sep. 2009. [Online]. Available: http://www.di.ens.fr/sierra/pdfs/bmvc09.pdf

[84] Y. Yang *et al.*, "Video captioning by adversarial LSTM," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5600–5611, Nov. 2018.

[85] Y. Li and L. Shen, "cC-GAN: A robust transfer-learning framework for HEp-2 specimen image segmentation," *IEEE Access*, vol. 6, pp. 14048–14058, 2018.

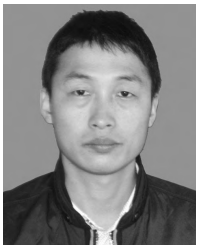
[86] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2242–2251.
[87] M. Mathieu, C. Couprie, and Y. Lecun, "Deep multi-scale video prediction beyond mean square error," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nov. 2015. [Online]. Available: https://arxiv.org/pdf/1511.05440
[88] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 613–621, [Online]. Available: https://arxiv.org/pdf/1609.02612
[89] P. Ghamisi and N. Yokoya, "IMG2DSM: Height simulation from single imagery using conditional generative adversarial net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 794–798, May 2018.
[90] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4681–4690.
[91] Q. Shi, X. Liu, and X. Li, "Road detection from remote sensing images by generative adversarial networks," *IEEE Access*, vol. 6, pp. 25486–25494, 2017.
[92] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1951–1959.
[93] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. (2016). "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling." [Online]. Available: https://arxiv.org/abs/1610.07584
[94] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3d view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 702–711.
[95] M. Gong, X. Niu, P. Zhang, and Z. Li, "Generative adversarial networks for change detection in multispectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2310–2314, Dec. 2017.
[96] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu, "DeLiGAN: Generative adversarial networks for diverse and limited data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4941–4949.
[97] L. C. Yang, S. Y. Chou, and Y. H. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proc. Int. Soc. Music Inf. Retr. (ISMIR)*, Mar. 2017. [Online]. Available: https://arxiv.org/pdf/1703.10847
[98] C. Esteban, S. L. Hyland, and G. Rätsch. (2017). "Real-valued (medical) time series generation with recurrent conditional GANs." [Online]. Available: https://arxiv.org/pdf/1706.02633
[99] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun. (2017). "Generating multi-label discrete electronic health records using generative adversarial networks." [Online]. Available: https://arxiv.org/pdf/1703.06490v1
[100] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "SegAN: Adversarial network with multi-scale L_1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, nos. 3–4, pp. 383–392, 2018.
[101] A. Ghosh, B. Bhattacharya, and S. B. R. Chowdhury, "SAD-GAN: Synthetic autonomous driving using generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nov. 2016. [Online]. Available: https://arxiv.org/abs/1611.08788
[102] A. Kurakin, I. Goodfellow, and S. Bengio. (2016). "Adversarial examples in the physical world." [Online]. Available: https://arxiv.org/abs/1607.02533
[103] L. Yu, W. Zhang, J. Wang, and Y. Yu. (2016). "SeqGAN: Sequence generative adversarial nets with policy gradient." [Online]. Available: https://arxiv.org/abs/1609.05473v5
[104] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. (2016). "Learning to generate images of outdoor scenes from attributes and semantic layouts." [Online]. Available: https://arxiv.org/abs/1612.00215
[105] P. Li, X. Liang, D. Jia, and E. P. Xing. (2016). "Semantic-aware grad-GAN for virtual-to-real urban scene adaption." [Online]. Available: https://arxiv.org/abs/1801.01726
[106] Y. Cao, L. Jia, Y. Chen, N. Lin, and X. Li, "Review of computer vision based on generative adversarial networks," *J. Image Graph.*, vol. 23, no. 10, pp. 1433–1449, 2018.

YANG-JIE CAO received the M.Sc. degree in computer science from Zhengzhou University, in 2006, and the Ph.D. degree in computer science from Xi'an Jiaotong University, in 2012. He is currently an Associate Professor with Zhengzhou University. His current research interests include computer vision, intelligent computing, artificial intelligence, and high-performance computing.

LI-LI JIA received the B.E. degree in computer science and technology from Zhengzhou University, Zhengzhou, China, in 2016, where she is currently pursuing the master's degree in computer technology. Her research interests mainly include deep learning and computer vision.

YONG-XIA CHEN received the M.Sc. degree in systems engineering from Northwestern Polytechnical University. She is currently a Lecturer with Zhengzhou University. Her current research interests include the IoT, intelligent computing, artificial intelligence, and deep learning.

NAN LIN received the B.S. degree in computer science and technology from Zhengzhou University, in 1996, and the M.S. degree in computer technology from the Huazhong University of Science and Technology. She is currently an Associate Professor with Zhengzhou University. Her research interests include the IoT, intelligent computing, computer vision, and deep learning.



CONG YANG received the B.S. degree in information security from Chongqing University, Chongqing, China, and the Ph.D. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2017. He is currently a Lecturer with Zhengzhou University, Zhengzhou, China. His research interests include medical image processing, computer vision, and deep learning.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2886814, IEEE Access

BO ZHANG received the B.Sc. degree in computer science from Southampton University, the M.Sc. degree in data communication networks and distributed systems from University College London, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology. His current research interests include teIoVT, wireless fog, ad hoc networks, multimedia broadcasting, and deep learning.

ZHI LIU (SM'11–M'14) received the B.E. degree from the University of Science and Technology of China, China, and the Ph.D. degree in informatics from the National Institute of Informatics. He was a Junior Researcher (Assistant Professor) with Waseda University and a JSPS Research Fellow with the National Institute of Informatics. He is currently an Assistant Professor with Shizuoka University.

His research interests include video network transmission, vehicular networks, and mobile edge computing. He is a member of IEICE. He was a recipient of the IEEE StreamComm 2011 Best Student Paper Award, the 2015 IEICE Young Researcher Award, and the ICOIN 2018 Best Paper Award. He is and has been a Guest Editor of journals, including *Wireless Communications and Mobile Computing*, *Sensors*, and the *IEICE Transactions on Information and Systems*. He has been serving as the chair for a number of international conferences and workshops.

XUE-XIANG LI received the B.S. degree in mathematics from Zhengzhou University, and the M.Sc. degree in computing mathematics from the Dalian University of Technology. He is currently a Professor with Zhengzhou University. His research interests include high-performance computing, cloud computing, and artificial intelligent.

HONG-HUA DAI received the M.Sc. degree from the Graduate School, Chinese Academy of Sciences, in 1986, and the Ph.D. degree from the Department of Computer Science, RMIT University, in 1994. After he received his Ph.D. degree, he was a Research Fellow with the Department of Computer Science, Monash University, from 1994 to 1997. He joined Deakin University, in 1999. His recent research interests include machine learning, data mining, and artificial intelligence. He is a member of ACM and the IEEE Computer Society.