Abstract—Image generation based on the generative adversarial network (GAN) has been widely used in the field of computer vision. It helps generate images similar to the given data by learning their distribution. However, in many tasks, training on small datasets of scenes may lead to mode collapse, such that the generated images are often blurred and almost the same. To solve this problem, we propose a generative adversarial network with transfer learning for mode collapse called TL-GAN. Owing to the size of the training dataset, we introduce transfer learning (VGG pre-training network) to extract more useful features from the underlying pixels and add them to the discriminator, which can be used to calculate the distance between samples, and to provide the discriminator with a new training target. The discriminator thus learns the best features that can distinguish between real data and generated data using the proposed model. This also enhances the learning capability of the generator, which learn further about the distribution of real data. Meanwhile, generator can produce new images more realistic. The results of experiments show that the TL-GAN can guarantee the diversity of samples. A qualitative comparison with several prevalent methods confirmed its effectiveness.

Index Terms—Image generation, Generative Adversarial Network, Mode collapse, Transfer learning, VGG pre-training network

I. INTRODUCTION

Most advanced deep neural network algorithms can learn highly complex problems and patterns [1]–[5], and their capabilities are impressive. However, humans can do far more than these algorithms on image recognition and speech recognition tasks, and it had appeared unlikely that such task could be automated. However, GANs [6]–[8] have made this possible.

Generative Adversarial Network (GAN) is an unsupervised method [9] that is among the most successful computer vision algorithms. It has been extensively researched and developed, especially in the context of image generation [10]–[13]. GAN was proposed by Ian Goodfellow in 2014, and consists of two parts: a generator network $G$ and a discriminator network $D$. The generator network generates samples by learning the distribution of real data. Its training objectives include maximizing the probability of error. The goal of the generator is to deceive the discriminator by generating images from random noise that are similar to real data, and the discriminator attempts to learn the differences between the generated samples and real ones.

Many applications of the GAN have been proposed [8], [14], [15]. For example, the iGAN contains two kinds of guidance information [8]: to paste the high-definition texture in the original image on the shape of the given object in it by using light field information in a different space. The texture on the paste is iterated until the resulting image appears realistic. Like the iGAN, the GP-GAN copies and pastes images directly [14], better integrates them into the original images and performs blending. However, it is a supervised training model. In the process of learning to blend, there is a supervised goal and a supervised loss function. CycleGAN allows two domains to transform each others images [15]. Traditional GAN features one-way generation while CycleGAN involves mutual generation. It is called a cycle because its network is ring-like. The input to CycleGAN is a pair of images.

Despite its wide application, it has many problems, such as training instability, gradient disappearance, mode collapse and so on. It is an important failure mode. Specifically, model collapse is called the most critical failure mode in GAN networks. Generators may reproduce exactly the same image, called pattern crash. Generally speaking, real world data distribution is highly complex and multimodal. The probability distribution described by the data has several “peaks” with different sub-groups of samples. The generator folds into a very narrow distribution, and the generated samples are no
**Fig. 1.** Model architecture. \( z \) denotes random variables from a uniform or a Gaussian distribution, and \( P_g \) denotes fake images generated by the generator network. \( P_{data} \) represents real data, and conv and deconv denote the convolutional and the deconvolutional layers, respectively. FC denotes a fully connected layer.

longer changed. This obviously violates GAN’s essence. For example, for face generation, we want to train a GAN that can generate face images. Generators need to learn hundreds of thousands or even millions of faces. However, if there are only tens of thousands of training data sets, the generator may collapse, resulting in poor diversity in the generated samples, which limits the usefulness of learning in GAN.

To solve the problem of mode collapse on small datasets in the GAN, we propose in this paper generative adversarial networks with transfer learning for mode collapse (TL-GAN). We introduce transfer learning (VGG pre-trained model) to extract useful features from the underlying pixels in images and add them to the discriminator. These features can be used to calculate the distance between samples and provide the discriminator with a new training target. This also enhances the learning capability of the generator, which can then learn more about the distribution of real data. Instead of encouraging samples generated by each generator to approach a single maximum likelihood, generator collapse is avoided. The overall performance of these samples are closer to the real image, and a suitable distance between different samples in the space can be ensured.

**II. RELATED WORK**

Although it is widely used, there are problems with the GAN, such as difficulty in training, mode collapse, and a lack of diversity in the generated samples. Several variants of the GAN have been proposed to solve these problems [10]–[13].

**DCGAN:** Since it was proposed in 2015, the DCGAN has been widely used in many applications, and significantly improves the stability of the training of the GAN and the quality of its results [10]. However, this only solves the problem temporarily and does not resolve difficulties in training. Furthermore, it is not easy to train \( D \) and \( G \) to reach equilibrium.

**LSGAN:** Proposed in 2016, the LSGAN uses the least-squares loss function instead of the loss function of the GAN to alleviate instability in its training and the lack of diversity that leads to poor image quality [11]. However, the LSGAN also has drawbacks. Its excessive penalty for outliers may lead to a reduction in the diversity of sample generation. It is likely to yield a simple imitation with minor modifications to the real data.

**WGAN:** The WGAN, developed in 2017, improves GAN in term of the loss function [12]. It removes sigmoid from the last layer of the discriminator, and forcibly truncates the updated weights to a certain range to meet the conditions of Lipschitz continuity. However, clipping the weight parameters blindly to guarantee stable training can lead to low-quality and low-resolution images.

**WGAN-GP:** The WGAN-GP, developed in 2017, is an improvement over the WGAN [13], specifically its continuity constraints. The contribution of WGAN-GP is that a technique to restrict Lipschitz continuity-the gradient penalty is proposed to solve the problem of mode collapse with a vanishing training gradient. While the WGAN-GP is an improvement over WGAN in some cases, it is not significantly superior to reliable GAN methods in terms of results. A number of researchers have attempted to solve problems in the GAN with mixed success.

**III. GAN WITH TRANSFER LEARNING**

**A. Approach**

In general, the GAN model uses a large number of training datasets, of the order of hundreds of thousands of images (CelebA, LSUN) [11], [16]. In such cases, the generated samples are diverse and high quality and conform to the
distribution of real data. However, in the case of a small training dataset, the generator network can learn only a few useful features such that the generated images look very fuzzy or even lead to mode collapse. To solve this problem, we introduce transfer learning that can transfer the influence of our knowledge on the visual perception of image generation tasks using a transmission parameter \cite{17}, \cite{18}. The GAN with a model pre-trained on millions of image datasets using ImageNet outperforms a zero-trained depth model on the same small dataset. We use the trained VGG pre-trained model on real and generated images.

**B. Model Architecture**

Our network structure is shown in Figure 1. It comprises the generator network $G$, discriminator network $D$, and a VGG pre-trained model. The target of $G$ captures the potential distribution of real data samples $I_{\text{data}}$, in addition to generating new data samples $I_{g}$. It starts with a random vector $z$ as input to the generator network followed by a fully connected layer and subsequently employs a high-dimensional tensor with 512 feature maps. Note that we use the batch normalization layer for each deconvolutional layer, and the activation function makes use of the rectifier linear unit (ReLU) except in the last deconvolution layer.

$D$ is a dual classifier and determines whether the input to it is from the real data $I_{\text{data}}$ or generated samples $I_{g}$. The discriminator network also makes use of multiple convolutions and uses the LeakyReLU as an activation function. At the end of the convolution, the tensor is stretched into a vector, which uses a fully connected layer. Our model employs a kernel of size $5 \times 5$ in both the generator and the discriminator. The VGG pre-trained model contains stacked convolution and coupled pooling layers. $P_{\text{data}}$ and $P_{g}$ are used as inputs to the VGG pre-trained model based on transfer learning to extract more high-level features. In the training process, the discriminator network aims to identify real and generated images, while the generator network attempts to make the generated samples more real with the aim of achieving the Nash equilibrium.

**C. Loss function**

1) Adversarial loss: Adversarial loss can maximize the probability of successfully determining whether a given image is from the training data or the generated samples. The goal of $D$ is to separate images generated by $G$ from real images. In this way, $G$ and $D$ constitute a min-max game. To better deceive the discriminator network such that it generates perceptually realistic images, we formally denote by $D$ the adversarial loss of the discriminator network and by $G$ that of the generator network. We define the adversarial losses of $\mathcal{L}_d$ and $\mathcal{L}_g$ as follows, respectively:

\[
\mathcal{L}_d = \mathbb{E}_{I_{\text{data}} \sim p_{\text{data}}(I_{\text{data}})} \left[ D(I_{\text{data}}) \right] + \mathbb{E}_{I_{g} \sim p_{g}(I_{g})} \left[ 1 - D(I_{g}) \right],
\]

\[
\mathcal{L}_g = \mathbb{E}_{I_{g} \sim p_{g}(I_{g})} \left[ 1 - D(I_{g}) \right],
\]

where $\mathcal{L}_g$ is the total loss of $G$, $D(\cdot)$ represents the output of the discriminator network, and $G(\cdot)$ is generated by $G$, which needs to learn real data distribution and produce more realistic images. When $D$ cannot distinguish between real data and generated samples, the generator can better deceive the discriminator network. The final optimization of $D$ is given by Eq. (3):

\[
D^* = \mathcal{L}_d + \lambda \mathcal{L}_p,
\]

where $D^*$ is the total loss of $D$, and $\lambda$ represents a constant.

2) Perceptual loss: We propose perceptual loss based on the VGG pre-training network to extract useful features from the underlying pixels and add some of them to the discriminator. These features are used to calculate the distance between samples and give the discriminator a new training target. It can thus solve the problem of blurred images and even mode collapse on small datasets. Moreover, the computational overhead of perception loss can be reduced by reusing the features

**Fig. 2.** Comparison of the DCGAN, LSGAN, WGAN, and WGAN-GP with our method for different epochs of a single sample on the CelebA dataset.
extracted by the discriminator. To introduce the formula for perceptual loss, we define perceptual loss as:

\[ L_p = |V(I_{\text{data}}) - V(I_g)|, \]

where \( L_p \) represents the perceptual loss, and \( V(.) \) represents the output of the VGG19 network.

IV. EXPERIMENTS

A. Details of Training

We used the CelebA and Cartoon datasets for training. In each training batch, we randomly selected 64 image patches as \( I_{\text{data}} \) patches, where each patch had a size of \( 64 \times 64 \). The random vector satisfied a uniform or a Gaussian distribution \( P_z(z) \) as input to the generator network. To stabilize the network, the range of intensity of the input and output images was set to \([-1,1]\).

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Samples</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CelebA</td>
<td>10,590</td>
<td>64 \times 64</td>
</tr>
<tr>
<td>Cartoon</td>
<td>51,223</td>
<td>64 \times 64</td>
</tr>
</tbody>
</table>

The CelebA face dataset is an open dataset of the Chinese University of Hong Kong [16]. It contains 202,599 face images with 10,177 celebrity identities. For our experiment, we selected only part of the data as a small dataset for training, with 10,590 images. The Cartoon face dataset was downloaded from Konachan.net, a well-known animation gallery, with a total of 51,223 images. Our experiments directly used the Cartoon face dataset shared by HE [19].

Our experiments were implemented in Python on the TensorFlow framework using an NVIDIA Tesla M40 GPU to accelerate training, using an SGD at a learning rate of 0.001 for all layers to analyze model performance. The momentum parameter was set to 0.3, weight decay to 0.0001, and 64 was set as the cardinality of each patch of training. All weights were truncated for a positive distribution and the standard deviation was 0.02. All convolutional layers were followed by leaky rectified linear units (LReLUs) with a slope of 0.2. Training the CelebA and the Cartoon datasets took 4 and 3 hours, respectively.

B. Comparisons and Analysis

To verify the effectiveness of the proposed model, we compared it with state-of-the-art methods, including the DCGAN [10], LSGAN [11], WGAN [12], and WGAN-GP [13]. It is worth pointing out that there are no quantitative indicators to refer to in previous papers because the generated synthetic data have no corresponding real data reference. There was no clear measure to assess the results of image generation tasks. Inspired by [10]–[13], we used the final generated images for comparison. Our experiments contained two parts. In the first, we analyzed the effectiveness of our method on two small datasets. We analyzed changes in a single sample in the training process, and we studied the final visual effects. In the second part, we considered the stability of training in several methods.

1) Evaluation of Visual Results: CelebA

We trained our model for 200 epochs on the CelebA dataset. For a fair comparison, we listed the generated samples for the same steps in each epoch using several methods (see Figure 2). In the final epoch, we observed that the DCGAN [10], LSGAN [11], WGAN [12], and WGAN-GP [13] produced poor results on a single sample. Because the dataset was small and training time limited, the DCGAN experienced mode collapse. It generated images of faces of females and males alternately. The training process of the LSGAN was very unstable. It first generated the images of a face not wearing glasses, then one with glasses, and again one without glasses. The WGAN and WGAN-GP generated blurry images of faces because of slow convergence. On the contrary, TL-GAN generated clear images of faces from random noise in a single sample.

Cartoon

We trained our model for 24 epochs on Cartoon dataset. We list the generated samples in the same steps in...
each epoch from several methods. From Figure 3 it is clear that the samples generated by our method showed a steady increment in quality, and produced a perceptually realistic image. However, the phenomenon of mode collapse appeared in some other methods (DCGAN and LSGAN). We also found that all other methods except the WGAN were unstable, their generated samples did not follow a fixed trend in the generation of shapes of face images, and they constantly changed in the training process. The DCGAN and LSGAN yielded images of faces of different cartoon characters in the training process, and two types of such images were generated in training by the WGAN-GP.

Samples of the CelebA and Cartoon datasets that were used are shown in Figure 4. The DCGAN on CelebA exhibited mode collapse, and it lacked sample diversity, and thus many generated samples were almost identical. LSGAN experienced a moderate mode collapse, and some of the images of faces generated by it look very odd. The WGAN trained very slowly on the CelebA dataset and its resulting images were very blurred. A slight mode collapse was observed in the WGAN on the Cartoon dataset. For example, the images of some faces were unusual and hairy body parts had unnatural details. Its results were better than those of the DCGAN and LSGAN, but the overall effect was unsatisfactory. The WGAN-GP also went through a slight mode collapse on the CelebA and Cartoon face datasets. The images of faces generated by it were vague and had strange details. The TL-GAN solved the mode collapse problem on the small datasets and guaranteed sample diversity. Moreover, the generator network quickly produced the images. It yielded quality images of faces.

2) Comparison of convergence: Loss convergence on the Cartoon dataset is shown in Figure 5. Loss of the DCGAN in the generator network increased, probably owing to the mode collapse caused by a too small dataset. The discriminator loss of the DCGAN also oscillated, and it could not differentiate real from the generated samples accurately. The loss of the generator and discriminator networks oscillated during the training of the LSGAN, which experienced a moderate mode collapse. Its convergence was also slow. The loss of the WGAN in the generator and discriminator networks tended to be normal, but was slower than that in our method and resulted in unclear samples generated after training. Gradient
penalty was introduced in the WGAN-GP, and increased the losses of the generator and discriminator networks at first. As training progressed, the losses decreased. But after training, the results were still not good, and the generated samples were thus insufficiently clear. The results on even the Cartoon dataset were unstable, as a single sample constantly changed and there was no fixed shape to the images generated. The TL-GAN converged more quickly and was more stable than the other methods. It also produced more perceptually realistic images.

V. Conclusion

In this paper, we highlighted the problem of mode collapse on small image datasets and proposed generative adversarial networks with transfer learning for mode collapse (TL-GAN). Experimental results show that our method substantially outperforms state-of-the-art approaches, and samples generated by it were close to real images. It can also ensure sample diversity. Compared with the other methods, the loss convergence of our model was fast and stable. In the future, we will continue to study how to use smaller data sets, such as thousands of images. Whether there will be collapse and good stability in our model.

References