SYNTHESIS OF IMAGES BY TWO-STAGE GENERATIVE ADVERSARIAL NETWORKS

Qiang Huang, Philip J.B. Jackson, Mark D. Plumbley, Wenwu Wang

Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, UK
Email: {q.huang, p.jackson, m.plumbley, w.wang}@surrey.ac.uk

ABSTRACT

In this paper, we propose a divide-and-conquer approach using two generative adversarial networks (GANs) to explore how a machine can draw colorful pictures (bird) using a small amount of training data. In our work, we simulate the procedure of an artist drawing a picture, where one begins with drawing objects' contours and edges and then paints them different colors. We adopt two GAN models to process basic visual features including shape, texture and color. We use the first GAN model to generate object shape, and then paint the black and white image based on the knowledge learned using the second GAN model. We run our experiments on 600 color images. The experimental results show that the use of our approach can generate good quality synthetic images, comparable to real ones.

Index Terms— Generative adversarial networks, conditional, image generation

1. INTRODUCTION

Automatic generation of colorful pictures has attracted increasing attentions in the last couple of years. The use of the GAN [1] and its extensions [2, 3, 4, 5] has improved the ability to generate good-quality pictures for different kinds of objects. A GAN generally contains two competing neural network models. One is called the *generator*, which takes noise as input and generates samples. The other model is called the *discriminator*, which receives samples from both the generator and the training data and distinguishes them. The two networks are trained simultaneously, during which the generator learns to produce more and more realistic samples, and the discriminator learns to get better and better ability to distinguish generated data from real data [6]. To generate good-quality images, GANs are often trained using large amounts of data. In [7], over 3 million training examples of the LSUN bedrooms dataset [8] were used to train a model. However, in many cases, it is hard to collect a large number of data to train GANs for a specific task. In addition, some recent results [2] have shown that the use of GAN can learn recognizable features of animals, such as fur, eyes, and noses, but these features are often not correctly combined to form an animal with realistic anatomical structure.

To tackle the challenging problem mentioned above, we develop a divide-and-conquer approach by separately processing shape and the other two visual features: color and texture. In our approach, we train two GAN models (GAN1 and GAN2). GAN1 is for shape generation, and GAN2 is for color image generation using color, texture and the shape information generated by GAN1. Our approach is based on two factors. The first factor is that shape is the most important element to convey the identity of an object among the three visual features in content-based image processing [9]. When learning new words, humans tend to assign the same name to similarly shaped items rather than to items with similar color, texture, or size [10]. To our knowledge, most previous studies used only GANs to tackle visual features as a whole. However, if we do not have a large number of image instances to train GANs, the number of samples of all visual features combinations may be small and lead to data sparsity. So, as a second factor, separately processing different visual information may be useful to reduce the adverse effect of data sparsity when training a GANs model on a small-sized dataset.

The rest of the paper is organized as follows. We introduce the overview of GAN and related work in Section 2. We present the details of our proposed framework in Section 3. The used data set and related evaluation metric are given in Section 4. We analyze the obtained results in Section 5, and finally conclude in Section 6.

2. BACKGROUND AND RELATED WORK

2.1. Generative Adversarial Networks

A generative adversarial network (GAN) consists of a generator network, G, whose goal is to learn a distribution matching a true data distribution, and a discriminator network D, which tries to distinguish real training data from synthetic data. G and D compete in a two-player minimax game with the following formulation:

$$\min_G \max_D V(G, D) \tag{1}$$

The work is funded by EPSRC grant EP/N014111/1.

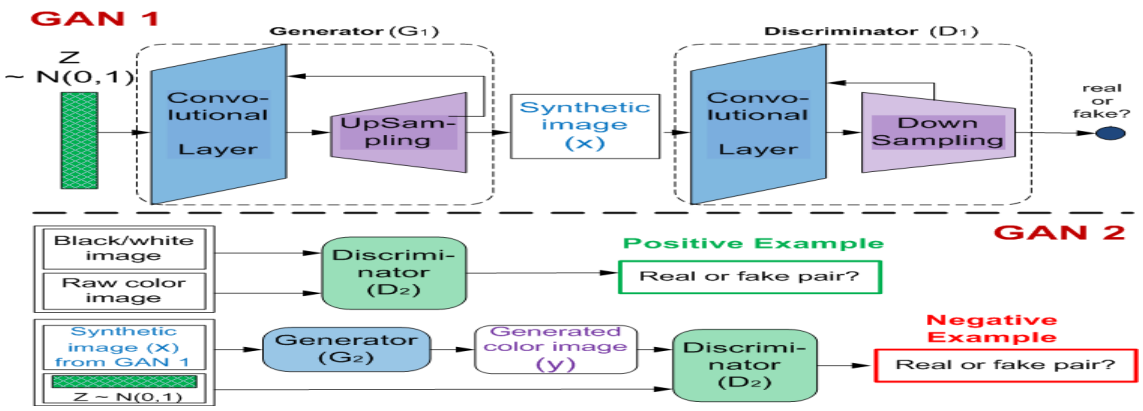

Fig. 1. A divide-and-conquer framework uses two GANs to simulate the procedure of human drawing picture. GAN1 is to generate object contours, and GAN2 is to paint the black-and-white image generated by GAN1.

where

$$V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (2)$$

where $p_{data}(x)$ is the true data distribution, $p_z(z)$ is a distribution to draw samples. The generator network, G, transforms a noise variable z into $G(z)$, which is a sample from distribution p_z, and ideally distribution p_z can converge to distribution p_{data} under mild conditions (e.g. G and D have enough capacity) [11]. The meaning of (1) is that the generator, G, tries to fool out the discriminator, D, while the discriminator wants to maximize the differentiation power between the true and generated samples.

2.2. Related work

GANs have been widely used in image generation [7, 12, 11, 13, 14, 15]. Radford et al. [7] used GANs and further developed a highly effective and stable architecture incorporating batch normalization to achieve striking image synthesis results. Reed et al.[11] built an end-to-end system for automatic synthesis of realistic images, from the character level to pixel level. Pathak et al. [13] applied image-conditional models to tackle inpainting, and Shrivastava et al. [14] also used conditional GANs to predict image. In addition to typical image synthesis mentioned above, some studies [16, 17, 18, 19, 20] have also been developed in image-to-image translation. Isola et al. employed a non-parametric texture model from a single input-output training image pair for image translation [16]. A more recent approach to learning a parametric translation function using GANs on a data set of input-output examples was mentioned in [17]. Similar ideas have also been applied to various tasks such as generating photographs from sketches [18] or from attribute and semantic layouts [19]. Zhu et al. [20] extended previous work [16, 18] to capture correspondences between high-level appearance structures by learning the mapping between two domains rather than between a pair of specific images.

In comparison with these previous studies, our work can be viewed as a combination of image synthesis and image translation mentioned above. We use image synthesis to generate image containing only object contours, and then use image translation to process color, texture information, and the shape information from GAN1. The details of our proposed framework will be discussed in the following section.

3. PROPOSED METHOD

Our approach, as shown in Fig. 1, aims to output a color image y given a noise vector z ($\mathbf{z} \sim N(0,1)$). We train two GAN models. The first GAN model, GAN1, generates an image, x, containing only object contours. The second GAN model, GAN2, paints the black-and-white image, x, different colors in order to produce the finally generated image, y. For GAN1, we use the GANs model mentioned in [7] following Eq. 2 as our task in GAN1 is only to generate object shape. For GAN2, we employ conditional GANs [17], where both the generator and discriminator are conditioned on extra information, x, generated by GAN1. This means x will be used as the input of GAN2.

In GAN1, the generator network is denoted $G_1 : \mathbb{R}^Z \rightarrow \mathbb{R}^M$, the discriminator as $D_1 : \mathbb{R}^M \rightarrow \{0,1\}$, where M is the dimension of the image, and Z is the dimension of the noise input to G_1. In the generator G_1, we sample from the noise prior $z \in \mathbb{R}^Z$. Following this, we use a deconvolutional network to generate a synthetic image x via $G_1(z) \rightarrow x$. For the discriminator D_1, to distinguish the fake and real image, we use convolutional layers followed by batch normalization and the activation function. We finally implement a classification to compute the final score from D_1.

The aim of GAN2 is to learn a mapping from random noise z and the observed image x generated by GAN1 to y, represented as $G_2 : \{z, x\} \rightarrow y$. We view (x, y) pairs as joint

observations and train the discriminator to judge (x, y) as real or fake. Suppose we have a batch $(x_i, y_i)_{i=1}^{n}$ of training images y_i paired with conditional data x_i and let $z_i \sim p_z(z)$ be noise data sampled from the noise distribution p_z. The cost equation for the discriminator D_2 is a logistic cost expression. We thus expect the discriminator to assign a positive label to true example (x_i, y_i), and a negative label to generated example $(G_2(z_i, x_i), x_i)$ [21]:

$$L_D = -\frac{1}{2n}\left(\sum_{i=1}^{n}\log D_2(y_i, x_i) + \sum_{i=1}^{n}\log(1 - D_2(G_2(x_i, z_i), x_i))\right) \tag{3}$$

where L_D is the discriminator loss function averaged over n samples.

The loss function for G_2 is to maximize the probability assigned by the discriminator to samples which come from G [21]:

$$L_G = -\frac{1}{n}\sum_{i=1}^{n}\log D_2(G_2(x_i, z_i), x_i) \tag{4}$$

As a whole, the objective function of the conditional GAN used here can be expressed as [21]:

$$\min_{G_2}\max_{D_2} V_{CGAN}(G_2, D_2) = \mathbb{E}_{x,y\sim p(x,y)}[\log D_2(y, x)] + \mathbb{E}_{x\sim p(x), z\sim p(z)}[\log(1 - D_2(G_2(x, z), x))] \tag{5}$$

The use of z is to learn to match the distribution learned from real data. It is generally represented by a random vector and used as the input of GANs [7]. We follow this step in GAN1. At the input layer of GAN2, we do not directly use vector z but instead apply a dropout function to x by masking some of its values randomly. After using dropout the input dimension is still same as x's, less than the dimension of concatenating z into x. We can thus simplify system structure and reduce computing cost.

4. DATA AND EXPERIMENTAL SETUP

In our work, the empirical evaluation is carried out on 600 bird images, which are extracted from Caltech-UCSD Birds 200 (CUB-200) [22], an image dataset with photos of 200 bird species. Before training our model, we resized all images to 128×128, smaller than their original size. This is to reduce the number of parameters used in our framework.

We extract bird edges using the rough image segmentation of CUB-200 for GAN1, and use both color image and its corresponding black-and-white image for GAN2. All models were trained with mini-batch stochastic gradient descent (SGD) with a mini-batch size of 32 and the Adam solver [23]. We set the learning rate empirically to be 0.0002 with momentum 0.9 for both GAN1 and GAN2. All weights were initialized randomly. The generator noise was sampled from a 100-dimensional unit normal distribution. To obtain stable results, GAN1 and GAN2 were trained with 1000 and 200 epochs, respectively. On this dataset, it took about two hours to train GAN1 and about six hours to train GAN2 on a single NVIDIA Pascal Titan X GPU.

We use two ways to evaluate our proposed approach. The first way is to demonstrate the synthesized images. The second way is to train classifiers using convolutional neural network (CNN) built using Keras [24] to measure whether the synthesized bird images can be categorized into a correct class.

To train classifiers, we consider two cases: (1) Case1: training data and test data are mismatched; (2) Case2: training data and test data are matched. For Case1, the training data is CIFAR10 [25], a 10-class benchmark image dataset, widely used in image classification evaluation [26]. The CIFAR10 image styles, such as resolution and center cropping, are different from CUB-200's. Case1 aims to evaluate the robustness of our synthesized images. For Case2, the bird images in CIFAR10 are replaced with the images from CUB-200, to evaluate the quality of the synthesized images generated using our model in comparison with images from CUB-200.

5. RESULT ANALYSIS

In each of 15 blocks of Fig. 2, we show 30 synthetic images generated in different conditions. The images shown in the first row contain only bird shapes and are generated using GAN1. As comparisons, in the other two rows, we also show the synthetic images generated using GAN1 trained in the other two different conditions. The first different condition (IMD1) is that we use 600 raw color images instead of their object contours. The second different condition (IMD2) is that we use the same 600 raw color images, but black their backgrounds and keep only the bird region of each image. IMD2 uses additional bird shape information in comparison with IMD1. The columns, from left to right in the figure, show an evolution procedure of generated images with increasing epochs. As GAN1 is initialized randomly, the images in the blocks at the leftmost column show random noise. When running more epochs, the generated images in the first row show that clearer birds' shapes are synthesized. However, the images generated in condition of IMD1 shown in the third row are only some colored squares, and the images generated in condition of IMD2 shown in the second row are some fuzzy and irregular colored images. It means that the use of GAN1 on raw color image fails to generate reasonable images. Even if IMD2 takes the bird shape information into account, the generated images in the second row do not show clearly recognizable bird either. As aforementioned in Section 1, this phenomenon is probably caused by the case that a small-sized dataset has a data sparsity problem when simultaneously processing a large number of visual features, which has a poor impact on optimizing a GANs model.

To evaluate classification performance, we test three kinds of bird images. TSGAN-IMG denotes that the images are generated using our approach. GAN-IMG denotes that the images are generated using GANs [7], whose training images contain only the bird and backgrounds in images are blacked.

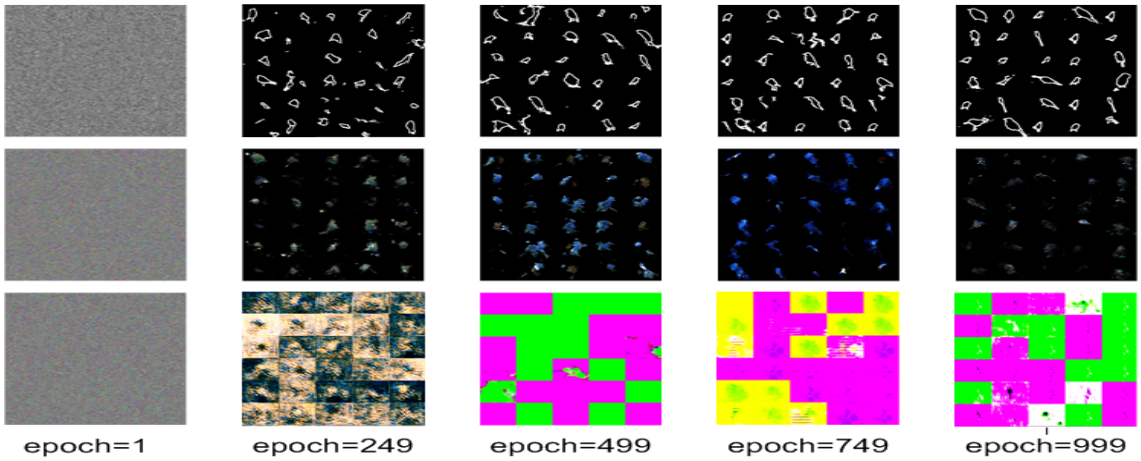

Fig. 2. Comparisons of synthetic bird images generated using GANs (GAN1) on different kinds of training images at five different epochs, $epoch \in \{1, 249, 499, 749, 999\}$. Each block contains 30 synthetic bird images.

	TSGAN-IMG	GAN-IMG	Real-IMG
Case1	32.63%	0.8%	8.40%
Case2	95.90%	37.25%	98.93%

Table 1. Comparisons of image classification accuracy

Real-IMG represents images randomly selected from CUB-200. The bird images selected from CUB-200 for classification test have no overlap with the images for classifier training. To train the classifiers, all training images are resized to 32×32 in order to match the image size of CIFAR10.

In Table 1, we show classification accuracy values obtained using classifiers on the three test datasets, TSGAN-IMG, GAN-IMG and Real-IMG, respectively. In condition of $Case2$, the classification accuracy on TSGAN-IMG reaches 95.90%, which is much better than GAN-IMG and is slightly less than that obtained on the CUB-200 bird images. This case probably means that the images of TSGAN-IMG contain most distinct bird features and have a comparable quality to the real images. The classification results, obtained in condition of $Case1$, are relatively low. This is maybe caused by a factor that to train the classifier we only select 600 image instances of each class, same as the number of instances to train GANs. This training number is relatively small for classification evaluation on CIFAR10 when testing mismatched data. Even if the classification condition is poor, we still find the classification accuracy obtained on TSGAN-IMG in condition of $Case1$ is even better than those obtained on Real-IMG. This is maybe caused by the irrelevant features learned from complex background of real images. They finally interfere the classifier. The low classification accuracy on GAN-IMG is caused by poor image quality.

As a comparison, from top to bottom row of Fig. 3, we also show the examples of TSGAN-IMG, GAN-IMG and Real-IMG, respectly. The quality of the images (TSGAN-IMG) generated using our approach is much better than those

Fig. 3. The three rows show the synthetic bird images of TSGAN-IMG and GAN-IMG, and the real images of Real-IMG.

generated using GANs [7]. In comparison with the real images, the quality of images of TSGAN-IMG is close to the real images although a bit more details appear in the real ones.

6. DISCUSSION

Based on the idea [9, 10] that shape information plays a more important role than color and texture for general object identification. We have presented an adversarial approach in order to strengthen the contribution of shape information. We have demonstrated that the use of our proposed approach can generate better bird images than a typical GANs model using and without using shape information on a small-sized dataset, and our classification accuracy results also show that the quality of generate images are comparable to real ones.

In our future work, we will test our approach on some smaller datasets in different conditions, and further test robust more methods for image segmentation. In addition, we will also consider perceptual testing and subjective evaluations.

7. REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of NIPS*, 2014, pp. 803–806.

[2] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proceedings of NIPS*, 2016.

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of NIPS*, 2016.

[4] X. Huang, Y. X. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in *Proceedings of CVPR*, 2017.

[5] J. Donahue, P. Krahenbunl, and T. Darrell, "Adversarial feature learning," in *Proceedings of ICLR*, 2017.

[6] J. Hayes, L. Melis, G. Danezis, and E. Cristofao, "Logan: Evaluating privacy leakage of generative models using generative adversarial networks," 2016, arXiv preprint arXiv:1705.07663v2.

[7] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, arXiv preprint arXiv:1511.06434.

[8] F. Yu, "LSUN bedroom dataset," Accessed: 2017-8-28, http://lsun.cs.princeton.edu/2017/.

[9] S. Prokopenko, "The basic elements shape, value, color, edge," Accessed: 2017-8-28, http://www.proko.com/basic-elements-shape-value-color-edge/.

[10] S. Ritter, D Barrett, A. Santoro, and M. M. Botvinick, "Cognitive psychology for deep neural network: A shape bias case study," 2017, arXiv preprint arXiv:1706.08606v2.

[11] S. Reed, Z. Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proceeedings of ICML*, 2016.

[12] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," 2016, arXiv preprint arXiv:1609.03126.

[13] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceeedings of CVPR*, 2016.

[14] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," 2016, arXiv preprint arXiv:1612.07828.

[15] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," in *Proceeedings of CVPR*, 2016.

[16] Isola P., J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016, arXiv preprint arXiv:1611.07004.

[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, arXiv preprint arXiv:1411.1784.

[18] P. Sangkloy, J. Lu, C. Fang, F. Yu, and H. J. Scribbler, "Scribbler: Controlling deep image synthesis with sketch and color," in *Proceeedings of CVPR*, 2017.

[19] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016, arXiv preprint arXiv:1612.00215.

[20] J.Y. Zhu, T Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017, arXiv preprint arXiv:1703.10593.

[21] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," Tech. Rep., Symbolic Systems Program, Natural Language Processing Group, 2015.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of ICLR*, 2015.

[24] F. Chollet, "Keras, deep learning library for python," Accessed: 2017-8-28, https://github.com/fchollet/keras.

[25] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," Accessed: 2017-8-28, https://www.cs.toronto.edu/ kriz/cifar.html.

[26] R. Benenson, "What is the class of this image," Accessed: 2017-8-28, http://rodrigob.github.io/are_we_there_yet/build/.