Conditional DCGAN For Anime Avatar Generation

Wang Hang
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai 200240, China
Email: wang–hang@sjtu.edu.cn

***Abstract*—Synthesis of realistic images from text would be interesting and useful, but current algorithms are still far from this goal. However, in recent years generic and powerful recurrent neural network architectures have been developed to learn discriminative text feature representations. Meanwhile, deep convolutional generative adversarial networks have begun to generate highly compelling images of specific categories, such as faces, birds, and rooms. In this work, I develop a novel deep architecture to bridge these advances in text and image modeling. The end result is an off-the-shelf encoder that can produce highly generic sentence representations that are robust and perform well in practice. I demonstrate the capability of my model to generate anime image from text descriptions.**

I. Introduction

In this work I am interested in translating descriptions text directly into image pixels. I aim to generate anime images based on the user's description, which can be used as user' social application avatar. For example, the user's description is "red hair blue eyes" or "yellow hair green eyes", my model will generate the corresponding avatar. In recent years there has been significant progress in learning visual-semantic embeddings, e.g. for zero-shot learning [1], [2] and automatically generating image captions for general web images [3], [4]. Motivated by these works, I aim to learn a mapping directly from words and characters to image pixels. Despite these advances, the problem of relating images and text is still far from solved.

A key challenge in image understanding is to correctly relate natural language concepts to the visual content of images. To solve this challenging problem requires solving two subproblems: first, learn a text feature representation that captures the important visual details; and second, use these features to synthesize a compelling image that a human might mistake for real. Fortunately, deep learning has enabled enormous progress in both subproblems and I build on this for my current task.

However, one difficult remaining issue not solved is that the distribution of images conditioned on a text description is highly multimodal. But learning is made practical by the fact that the word sequence can be decomposed sequentially according to the chain rule; i.e. one trains the model to predict the next token conditioned on the image and all previous tokens.

This conditional multimodality is thus a very natural application for generative adversarial networks (Goodfellow et al.[5], 2014), in which the generator network is optimized to fool the adversarially trained discriminator into predicting that synthetic images are real. By conditioning both generator and discriminator on side information, I can naturally model this phenomenon.

Recently, Generative adversarial networks (GANs) have demonstrated impressive performance for unsupervised learning tasks. The basic idea of GANs is to simultaneously train a discriminator and a generator: the discriminator aims to distinguish between real samples and generated samples; while the generator tries to generate fake samples as real as possible, making the discriminator believe that the fake samples are from real data. The generator and discriminator are typically learned jointly by alternating the training of D and G, based on game theory principles. So far, plenty of works have shown that GANs can play a significant role in various tasks, such as image generation, image super-resolution, and semi-supervised learning.

My main contribution in this work is to develop a simple and effective GAN architecture and training strategy that enables text to image synthesis of anime faces from language descriptions.

The rest of this paper is organized as follows. Section II briefly reviews related work of generative adversarial networks,and section III introduces the background of my work. The proposed method is introduced in Section IV, and experimental results are presented in Section V. Finally, I draw the conclusion about the paper.

II. Related Works

Generative Adversarial Networks (GANs) have been proposed by Goodfellow et al.[5], who explained the theory of GANs learning based on a game theoretic scenario. GANs also can be trained through differentiable networks. Showing the powerful capability for unsupervised tasks, GANs have been applied to many specific tasks, like image generation, text to image synthesis and image to image translation.

Despite the great successes GANs have achieved, improving the quality of generated images is still a challenge. A lot of works have been proposed to improve the quality of images for

GANs. Radford et al.[6] first introduced convolutional layers to GANs architecture, and proposed the network architecture called deep convolutional GANs (DCGANs), which led to improved visual quality.

Another critical issue for GANs is the stability of learning process. Specifically, minimizing the objective function of regular GAN suffers from vanishing gradients, which makes it hard to update the generator. Many works have been proposed to address this problem by analyzing the objective functions of GANs. Arjovsky et al.[7] introduced Wasserstein GANs (WGANs). WGANs introduced a loss function that also acts as a measure of convergence. In their implementation it comes at the expense of slow training, but with the benefit of stability and better mode coverage. More recently, Energy Based GANs [8] (EBGANs) were proposed as a class of GANs that aims to model the discriminator D as an energy function. This variant converges more stably and is both easy to train and robust to hyper-parameter variations.

Key challenges in multimodal learning include learning a shared representation across modalities, and to predict missing data in one modality conditioned on another. Many researchers have recently exploited the capability of deep convolutional decoder networks to generate realistic images. Generative adversarial networks have also benefited from convolutional decoder networks, for the generator network module. Denton et al.[9] used a Laplacian pyramid of adversarial generator and discriminators to synthesize images at multiple resolutions. This work generated compelling high-resolution images and could also condition on class labels for controllable generation. Radford et al.[10] used a standard convolutional decoder, but developed a highly effective and stable architecture incorporating batch normalization to achieve striking image synthesis results.The main distinction of my work from the conditional GANs described above is that my model conditions on text descriptions instead of class labels.

The bulk of previous work on multimodal learning from images and text uses retrieval as the target task, i.e. fetch relevant images given a text query. In contemporary work [11] generated images from text captions, using a variational recurrent autoencoder with attention to paint the image in multiple steps. Impressively, the model can perform reasonable synthesis of completely novel text such as “a stop sign is flying in blue skies”, suggesting that it does not simply memorize. While the results are encouraging, the problem is highly challenging and the generated images are not yet realistic, i.e., mistakeable for real. Recent image and video captioning models [12] go beyond tags to generate natural language descriptions. These models use LSTMs [13] for modeling captions at word level and focus on generating general high-level visual descriptions of a scene. As an alternative to using LSTMs for language modeling, other works have used character-based convolutional networks [14].

Building on ideas from these many previous works, I develop a simple and effective approach for text-based image synthesis using a character-level text encoder and conditional GAN, which leads to compelling visual results.

III. Background

In this section, I briefly describe several previous works that I build my method on.

A. Generative adversarial networks

GANs, originally introduced by Goodfellow consist of two neural networks, G and D. G maps a low-dimensional space to the high dimensional sample space of x. D is a binary neural network classifier. In the training phase, G and D are typically learned in an adversarial fashion using actual input data samples x and random vectors z. An Gaussian prior is usually assumed on z. While G learns to generate outputs $G(z)$ that have a distribution similar to that of x, D learns to discriminate between real samples x and fake samples $G(z)$. D and G are trained in an alternating fashion to minimize the following min-max loss (Goodfellow et al [5], 2014):

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

It was shown that the optimal GAN is obtained when the resulting generator distribution $p_g = p_{data}$, and that under mild conditions (e.g. G and D have enough capacity) p_g converges to p_{data}. In practice, in the start of training samples from D are extremely poor and rejected by D with high confidence. It has been found to work better in practice for the generator to maximize $\log(D(G(z)))$ instead of minimizing $\log(1-D(G(z)))$.

B. Deep Structured Joint Embedding

To obtain a visually-discriminative vector representation of text descriptions, I follow the approach of [15] by using deep convolutional and recurrent text encoders that learn a correspondence function with images. The text classifier induced by the learned correspondence function f_t is trained by optimizing the following structured loss:

$$\frac{1}{N}\sum_{n=1}^{N} \Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n)) \tag{2}$$

where $\{(v_n, t_n, y_n) : n = 1, ..., N\}$ is the training data set, Δ is the 0-1 loss, v_n are the images, t_n are the corresponding text descriptions, and y_n are the class labels. Classifiers f_v and f_t are parametrized as follows:

$$f_v(v) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{t\sim\mathcal{T}(y)}[\phi(v)^T\varphi(t))] \tag{3}$$

$$f_t(t) = \arg\max_{y\in\mathcal{Y}} \mathbb{E}_{v\sim\mathcal{V}(y)}[\phi(v)^T\varphi(t))] \tag{4}$$

where ϕ is the image encoder (e.g. a deep convolutional neural network), φ is the text encoder (e.g. a character-level CNN or LSTM), $\mathcal{T}(y)$ is the set of text descriptions of class y and likewise $\mathcal{V}(y)$ for images. The intuition here is that a text encoding should have a higher compatibility score with images of the correspondong class compared to any other class and vice-versa.

To train the model a surrogate objective related to Equation 2 is minimized. The resulting gradients are backpropagated through φ to learn a discriminative text encoder.

IV. Method

In this section, I present proposed method in Section A. Next, model Architectures is introduced in Section B. Table I shows the notation used later.

TABLE I

x	data
t	text
z	noise
G	generator network
D	discriminator network
T	dimension of the text
I	dimension of the image
$\mathcal{N}$	the dimension of the noise
φ	encoder
$\mathcal{L}$	loss
s	score

A. Proposed GAN

My method is to train a deep convolutional generative adversarial network (DCGAN) under the condition of text features encoded by a mixed character-level convolutional neural network.

The most straightforward approach to train a conditional GAN is to treat (text, image) pairs as joint observations and train the discriminator to judge pairs as real or false. This type of conditioning is naive because the discriminator does not have a clear idea of whether the actual training image matches the text embedding context. The discriminator observes two inputs: real images with matching text, and synthetic images with arbitrary text. Therefore, it must implicitly separate two sources of error: unrealistic images, and real images of the wrong category that mismatch the conditioning information.

Based on the this, I modified the GAN training algorithm to separate these error sources. In addition to the real / fake inputs to the discriminator during training, I add a third type of input, which contains real images with mismatched text, and the discriminator must learn to score it as fake. In addition to the image realism, the discriminator can provide additional signals to the generator by learning to optimize image / text matching.

It is shown that deep neural networks learn representations in which interpolations between embedding pairs tend to be near the data manifold [16]. Inspired by this property, I can generate a large amount of extra text embeddings by simply interpolating between embeddings of training set captions. Importantly, these interpolated text embeddings do not need to correspond to any actual artificial text and therefore do not require additional tagging costs.

This can be seen as an additional term added to the generator objective to minimize:

$$\mathbb{E}_{t_1,t_2\sim p_{data}}[\log(1-D(G(z,\beta t_1+(1-\beta)t_2)))] \quad (5)$$

where β interpolates between text embeddings t_1 and t_2. In practice, fixing $\beta = 0.5$ works well.

Because the interpolated embeddings are synthetic, the discriminator D does not have "real" corresponding image and text pairs to train on. However, D learns to predict whether image and text pairs match or not. Therefore, if D does a good job in this respect, then by satisfying D on interpolated text embeddings G can learn to fill in gaps in the data manifold in between training points.

Algorithm 1 Training algorithm.

```
1: Input: images $x$, texts $t$, $\hat{t}$, training epochs $E$
2: for $n = 1$ to $E$ do
3:     $h \leftarrow \varphi(t)$, $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode text description}
4:     $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
5:     $\hat{x} \leftarrow G(z,h)$ {Generate fake image}
6:     $s_r \leftarrow D(x,h)$, $s_w \leftarrow D(x,\hat{h})$, $s_f \leftarrow D(\hat{x},h)$
7:     $\mathcal{L}_D \leftarrow \mathcal{L}_{s_r} + \mathcal{L}_{s_w} + \mathcal{L}_{s_f}$, $\mathcal{L}_G \leftarrow \mathcal{L}_{s_f}$ {Loss of $D$, $G$}
8:     $D \leftarrow D - \alpha\partial\mathcal{L}_D/\partial D$ {Update discriminator}
9:     $G \leftarrow G - \alpha\partial\mathcal{L}_G/\partial G$ {Update generator}
10: end for
```

Algorithm 1 summarizes the training procedure. After encoding the text, image and noise (lines 3-4), I generate the fake image ($\hat{x}$, line 5). s_r indicates the score of associating a real image and its corresponding sentence, s_w measures the score of associating a real image with an arbitrary sentence, and s_f is the score of associating a fake image with its corresponding text. Note that I use $\partial\mathcal{L}_D/\partial D$ to indicate the gradient of D's objective with respect to its parameters, and likewise for G.I use minibatch SGD for simplicity. Lines 8 and 9 are meant to update network parameters.

B. Model Architecture

I illustrate my network architecture in Figure 1.

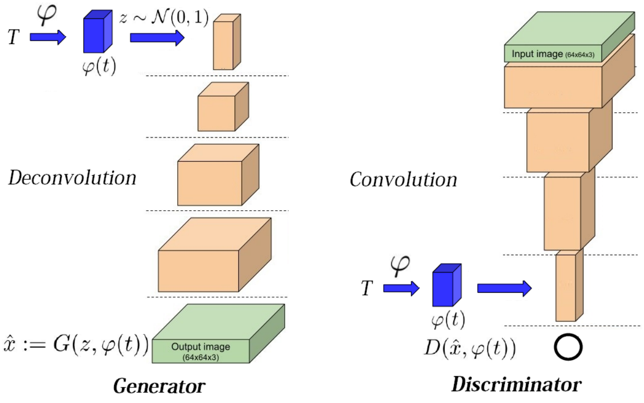


Fig. 1. My model architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator.

In the generator G, first I sample from the noise prior $z \sim \mathcal{N}(0, 1)$ and I encode the text query t using text encoder φ. The description embedding $\varphi(t)$ is first compressed using a fully-connected layer to a small dimension (in practice I used 256) followed by leaky-ReLU and then concatenated to the noise vector z. Following this, there are 4 layers of deconvolution layer (kernel size is 5, stride is 2, filters are [256, 128, 64, 3]), and after each layer of deconvolution. Batch normalization and relu activation is performed on all layers.Finally, a synthetic image $\hat{x}$ is generated via $\hat{x} \leftarrow G(z, \varphi(t))$. Image generation corresponds to feed-forward inference in the generator G conditioned on query text and a noise sample.

In the discriminator D, I perform 4 layers of convolution with spatial batch normalization followed by leaky ReLU. I again reduce the dimensionality of the description embedding $\varphi(t)$ in a fully-connected layer followed by rectification. The image goes through four layers of convolution layer (kernel size is 5, stride is 2, filters are [64, 128, 256, 512]), and both are leaky relu(max(x,0.2*x)), where After the three layers of the convolution layer pass through the leaky relu. Batch normalization is performed on all convolutional layers. When the spatial dimension of the discriminator is 4×4, I replicate the description embedding spatially and perform a depth concatenation. I then perform a 1×1 convolution followed by rectification and a 4×4 convolution to compute the final score from D.

V. Experiment Details

In this section, I introduce the experiment details. First, environment configuration is show in Section A. Next, I present the datasets and the data preprocessing in Section B and Section C. Then, training details is shown in Section D. Finally, results are introduced in Section E.

A. *Environment*

In the experiment, the environment used is shown in Table II.

TABLE II

OS	Ubuntu 16.04	CPU	Intel Core i7-6700HQ
Memory	8GB	GPU	NVIDIA GTX970M
Libraries	Tensorflow-gpu, CUDA, theano, skipthoughts		

B. *Dataset*

In the experiment, the anime_face_data dataset is used as training data. The data is collected by En Yu Fan, who collected the data from the following website: http://konachan.net/post. The size of the image is 64×64.

Meanwhile, the author counts the information of images and makes a tag file. Labels include but are not limited to hair, dressing, hat, eyes, ears etc. For example, a picture's lable is "clannad: 848 sagara misae: 17 blue hair: 11670 long hair: 54412 seifuku: 18412 yellow eyes: 7520". The number after the tag represents the number of people who consider the image has this feature.

C. *Preprocessing*

The image tags have too much information, but I only use the color information of the hair and eyes, so preprocessing of data is needed.

Firstly, delete tags that have no color meanings such as long, short in front of the hair, and use only the two tags with hair and eyes. If there is more than one color in the same part, only the highest number of votes will be used. Therefore, each picture has only one color characteristic of hair and eyes, for example, "blue hair yellow eyes".

After finishing processing, I get a total of 8956 pictures, corresponding to 10 colors for hair and 9 colors for eyes.

Some random seclected images form the processed dataset is shown in Table III.

TABLE III

purplr hair black eyes	blonde hair pink eyes	red hair orange eyes	green hair green eyes

D. *Training*

During the training period, D became very strong and it was easy to distinguish the real and produced pictures (d_loss was small, below 0.5), compared to g_loss (approximately 1 5). After increasing the d:g ratio from 1:2 to 1:5, the g_loss has decreased significantly (falling to 0.7 2), but the quality of the picture has not been significantly improved; finally, I adjust the learning rate of the D to 0.2, slowing discriminator's training speed gives me the clearest picture.

Moreover, I have tested the results of projecting text to 128 dimensions and 256 dimensions respectively. After training, it is found that inputting a 256-dimensional feature, G can produce the right color in the 4th epoch. If it is a 128-dimensional feature, G produce the right color matching the discription until the 13th epoch. So, a 256-dimensional feature is better for my model.

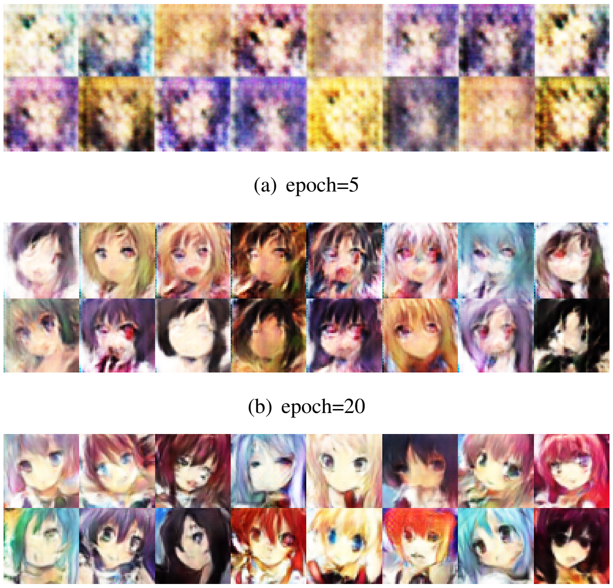
After testing and adjusting parameters, I got my final model. The model converges quickly, but each epoch takes a long time. It is found in the fifth epoch that I can basically see the face in the training process. In the end, the generator can produce images that have a clear outline of the face.I illustrate the training process in Figure 2.

E. *Results*

Given text descriptions, the model will generate corresponding images with size 64×64.

The generated image will go through a stage of getting blurred first, then the outline of the face will appear, gradually become clear, and the final image will be generated. The generation process is shown in Figure 3.

The model learn to generate anime image successfully and some selected images are shown in Table IV. Two major



(a) epoch=5

(b) epoch=20

(c) epoch=40

Fig. 2. training process (When the epoch is 5, 20, 40 respectively, fake images generated by the generator is shown.)

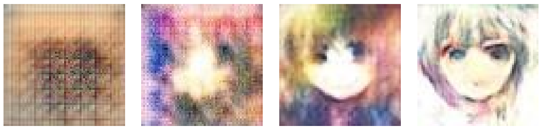
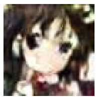
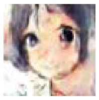
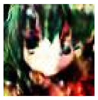
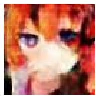
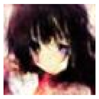
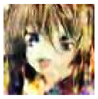


Fig. 3. genarating process

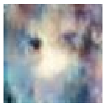
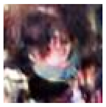
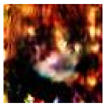
conclusions can be observed. First, the generated images are of medium quality, which can be used as user's Avater. Second, correct images are generated through label vectors.

TABLE IV

black hair black eyes	brown hair purple eyes	gray hair white eyes	green hair black eyes
			
green hair green eyes	orange hair red eyes	pink hair blue eyes	purple hair blue eyes
			
purple hair orange eyes	purple hair purple eyes	white hair orange eyes	yellow hair brown eyes
			

However, there are also some unsatisfactory sample, which are of poor-quality. Some selected images are shown in Table V. The generated images are very vague and even the outline of the human face is invisible, which could be due to the fact that the model is not fully trained.

TABLE V

purple hair red eyes	green hair blue eyes	blue hair brown eyes	green hair green eyes
			
brown hair yellow eyes	blue hair green eyes	brown hair brown eyes	purple hair green eyes
			

Overall, my model has a short convergence time and can see a clear face in a relatively short period of time (Computing resources are limited, so convergence time is a big consideration).

VI. Conclusion

In this work I developed a simple and effective model for generating images based on descriptions. I demonstrated that the model can synthesize the anime image of a given text caption. In future work, I aim to further scale up the model to higher resolution images and add more types of text.

Acknowledgment

I am fortunate and thankful for all the advice and guidance I have received during this work, especially that of Xinbing Wang, Xiaohua Tian, and Luoyi Fu. Additionally Id like to thank carpedm for the open source code of DCGAN (https://github.com/carpedm20/DCGAN-tensorflow).

References

[1] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell, "Zero-shot learning with semantic output code," in *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[2] J. Pennington, R. Socher, and C. D. Manning. Glove, "Global vectors for word representation," in *Empirical Methods on Natural Language Processing (EMNLP)*, 2014.

[3] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[4] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. choi, A. Berg, and T. Berg. Baby talk, "understanding and generating simple image descriptions" in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[6] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2015.

[7] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv: 1701.07875*, 2017.

[8] Junbo Zhao, Michael Mathieu, and Yann LeCun, "Energy-based generative adversarial network," *arXiv :1609.03126*, 2016.

[9] Denton, E. L., Chintala, S., Fergus, R, "Deep gener- ative image models using a laplacian pyramid of adversarial networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[10] Radford, A., Metz, L., and Chintala, S, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016.

[11] Mansimov, E., Parisotto, E., Ba, J. L., and Salakhutdinov,R., “Generating images from captions with attention,” in *International Conference on Learning Representations (ICLR)*, 2016

[12] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, “Neural im- age caption generation with visual attention,” in *International Conference on Machine Learning (ICML)*, 2015.

[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” in *Neural Computation*, 1997.

[14] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification.” in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[15] Reed, S., Akata, Z., Lee, H., and Schiele, B, “Learning deep representations for fine-grained visual descriptions,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] Reed, S., Sohn, K., Zhang, Y., and Lee, H, “Learning to disentangle factors of variation with manifold interaction,” in *International Conference on Machine Learning (ICML)*, 2014.