Automatic Anime Characters Creation

Hongzhi Liu

July 10, 2018

Abstract

Kids like anime characters, which often takes tremendous efforts to master the skill of drawing after which one are capable of designing their own characters. Automatic generation of facial images has been well studied after the Generative Adversarial Network(GAN) came out. There exists some attempts applying the GAN model to the problem of generating facial images of anime characters, but none of the existing work gives a promising result. Today, I read a thesis written by Yanghua Jin who is from School of Computer Science in Fudan University. His team explore the training of GAN models specialized on an anime facial image dataset and they address the issue from both the data and the model aspect, by collecting a more well-suited dataset and empirical application of DRAGAN. With quantitative analysis and case studies, they demonstrate that their efforts lead to a stable and high quality model.

1. Overview of the Model

Existing literature provides several attempts for generation facial images of anime characters. Among them are Mattya and Rezoolab who first explored the generation of anime character faces right after the appearance of DCGAN [5]. Besied, codes are made available online focusing on anime faces generation. However, since results from these works are blurred and distorted on an untrivial frequency, it still remains a challenge to generate industry-standard facial images for anime characters.

In this paper, Yanghua Jin proposes a model that produces anime faces at high quality with promising rate of success [3]. Their contribution can be described as three-fold that are a clean dataset which they collected from Getchu, a suitable GAN model based on DRAGAN, and their approach to train a GAN from images without tags which can be leveraged as a general approach to training supervised or conditional model without tag data.

2. Generative Adversarial Network

Generative Adversarial Network (GAN) [1] proposed by Goodfellow *et al.*, shows impressive results in image generation [5], image transfer [2], super-resolution [4] and many other generation tasks. In this work, Jin and his team explore the training of GAN models specialized on an anime facial image dataset.

2.1. Vanilla GAN

GAN uses a generator network G to generate samples from P_G. This is done by transforming a latent noise variable $z \sim P_{noise}$ noise into a sample $G(z)$. The original GAN uses a min-max game strategy to train the generator

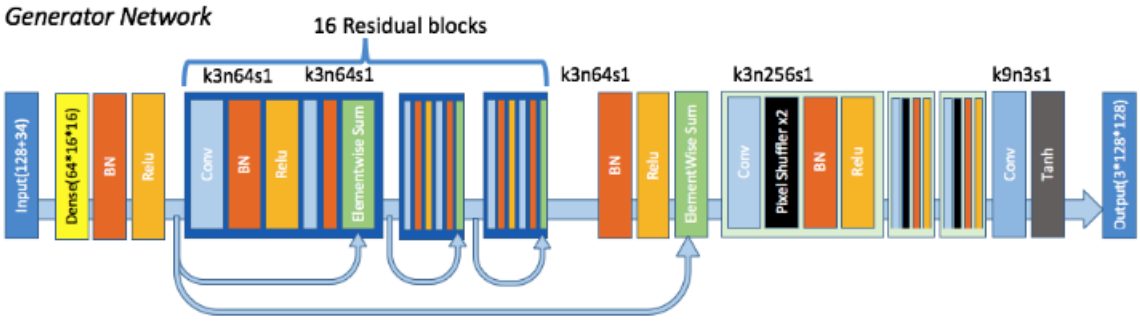

Figure 1. Generator Architecture

Table 1. Number of dataset images for each tag.

blonde hair 4991	brown hair 6659	black hair 4842	blue hair 3289	pink hair 2486	purple hair 2972	green hair 1115
red hair 2417	silver hair 987	white hair 573	orange hair 699	aqua hair 168	gray hair 57	long hair 16562
short hair 1403	twintails 5360	drill hair 1683	ponytail 8861	blush 4926	smile 5583	open mouth 4192
hat 1403	ribbon 5360	glass 1683	blue eyes 8861	red eyes 4926	brown eyes 5583	green eyes 4192

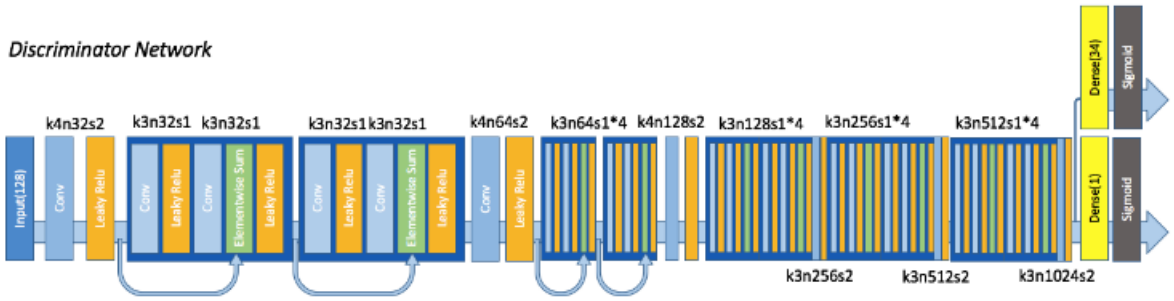

Figure 2. Discriminator Architecture

G, imposing another network D to distinguish samples from G and real samples. Formally, the objective of GAN can be expressed as Eq. 1:

$$\min_G \max_D \mathcal{L}(D,G) = \mathbb{E}_{x\sim P_{data}}[\log D(x)] + \mathbb{E}_{x\sim P_{noise}}[\log(1-D(G(z)))]. \quad (1)$$

Where the discriminator D try to maximize the output confidence score from real samples. Meanwhile, it also minimizes the output confidence score from fake samples generated by G. On contrast, the aim of G is to maximize the D evaluated score for its outputs, which can be viewed as deceiving D.

2.2. GAN with Labels

Incorporating label information is important in their task to provide user a way to control the generator. Their utilization of the label information is inspired by ACGAN [3]: The generator G receive random noise z along with a 34-dimension vector c indicate the corresponding attribute conditions. They add a multi-label classifier on the top of discriminator network, which try to predict the assigned tags for the input images. In detail, the loss is described as Eq. 2:

$$\begin{aligned}\mathcal{L}_{adv}(D) =& -\mathbb{E}_{x\sim P_{data}}[\log D(x)] \\ & -\mathbb{E}_{x\sim P_{noise},c\sim P_{cond}}[\log(1-D(G(z,c)))], \\ \mathcal{L}_{cls}(D) =& \mathbb{E}_{x\sim P_{data}}[\log P_D[label_x|x]] \\ & +\mathbb{E}_{x\sim P_{noise},c\sim P_{cond}}[\log(P_D[c|G(z,c)])].\end{aligned} \quad (2)$$

where P cond indicates the prior distribution of assigned tags.

2.3. Model Architecture

The generator's architecture is shown in Fig. 1, which is a modification from SRResNet [4]. The model contains 16 ResBlocks and uses 3 sub-pixel CNN [6] for feature map upscaling. Fig. 2 shows the discriminator architecture, which contains 10 Resblocks in total. All batch normalization layers are removed in the discriminator, since it would bring correlations within the mini-batch, which is undesired for the computation of the gradient norm. They add an extra fully-connected layer to the last convolution layer as the attribute classifier. All weights are initialized from a Gaussian distribution with mean 0 and standard deviation 0.02.

3. Experiment Results of Model

The technology of making game characters and CGs is evolving continuously, therefore the release year of the game plays an important role for the visual aspect of image quality. Jin and his team train the GAN model using images from games released after 2005 and with scaling all training images to a resolution of 128×128 pixels. This gives 31255 training images in total. As one can know, characters before 2003 are old-fashioned, while characters in the recent games is cuter and have better visual quality. As Tab. 1 states, labels are not evenly distributed in our training dataset, which results that some combinations of attributes cannot give good images.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 1

[2] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2016. 1

[3] Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu, and Z. Fang. Towards the automatic anime characters creation with generative adversarial networks. *arXiv preprint arXiv:1708.05509*, 2017. 1, 2

[4] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1, 2

[5] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science*, 2015. 1

[6] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 2