AI Illustrator: Art Illustration Generation Based on Generative Adversarial Network

Zihan Chen[1,*], Lianghong Chen[1,a], Zhiyuan Zhao[1,b], Yue Wang[1,c]
[1]College of Computer Science, Sichuan University, Chengdu, China
e-mail: [*]chenzihan294@qq.com, [a]yegetaier7977@qq.com, [b]jiao_99@126.com, [c]yuewangdiao@qq.com

***Abstract*—In recent years, people's pursuit of art has been on the rise. People want computers to be able to create artistic paintings based on descriptions. In this paper, we proposed a novel project, Painting Creator, which uses deep learning technology to enable the computer to generate artistic illustrations from a short piece of text. Our scheme includes two models, image generation model and style transfer model. In the real image generation model, inspired by the application of stack generative adversarial networks in text to image generation, we proposed an improved model, IStackGAN, to solve the problem of image generation. We added a classifier based on the original model and added image structure loss and feature extraction loss to improve the performance of the generator. The generator network can get additional hidden information from the classification information to produce better pictures. The loss of image structure can force the generator to restore the real image, and the loss of feature extraction can verify whether the generator network has extracted the features of the real image set. For the style transfer model, we improved the generator based on the original cycle generative adversarial networks and used the residual block to improve the stability and performance of the u-net generator. To improve the performance of the generator, we also added the cycle consistent loss with MS-SSIM. The experimental results show that our model is improved significantly based on the original paper, and the generated pictures are more vivid in detail, and pictures after the style transfer are more artistic to watch.**

Keywords-Image generation; style transfer

I. INTRODUCTION

With the improvement of the spiritual need of the public, people have higher requirements for books, among which the illustration of the relevant words in the books is an urgent solution. The traditional method of completing the related illustrations by illustrators has been unable to meet the need of the growing book market. Fortunately, the task of matching illustrations to books according to their content belongs to text-to-image translation application, and there are many methods based on deep learning that has achieved remarkable results in this task. However, there is still little research on art illustrations based on text. So, we thought about creating a way to solve this problem using deep learning. In this paper, we propose a new scheme, Painting Creator, which enables us to quickly generate artistic illustration images from texts. We just needed to enter the relevant text, Painting Creator will create meaningful and elegant illustrations for users, as showed in Figure 1. There are many research results on the generation of text to real images and style transfer. However, few studies combine such two works. We hope that the input of user input text can generate meaningful art illustrations because we aim to build a more convenient art style generation system for users. At the same time, after our research work, our model has some improvement on the original research.

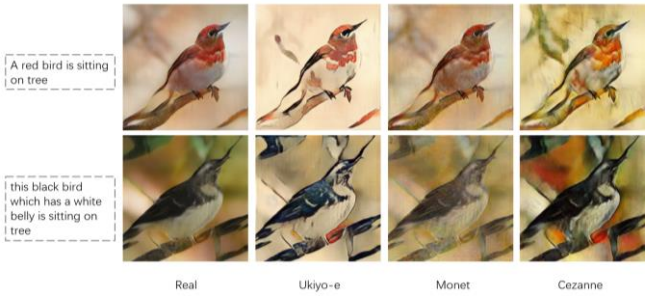

Figure 1. Artistic illustrations created by Painting Creator

To solve the problem of generating illustrations from the given textual information, our model is done in two steps. Considering that the picture dataset of cartoon illustration style is difficult to collect and does not have the universality, our model is divided into two parts. First, we complete the generation from text to real picture, and then we use the method of style transfer to convert the picture into illustration.

For from the text generated vivid content real images, based on StackGAN, we set up IStackGAN (Improved Stack Generative Adversarial Networks) model. Based on the original, we add image category information, feature reconstruction information and structure reconstruction training information, to generate better image quality. Experiments show that we have improved the score of each index to a certain extent compared with the original model.

The artistic style transfer algorithm is proposed in the traditional style transfer [2] can combine two images with low-level features and high-level features by using the pre-trained VGG16 network [14]. However, this method takes too long, and the style of the picture texture color and other features forced into the original picture, often appear unreal. The style transfer effect we need is to transform the real picture according to the style of a class of work, without the need to fuse the color and texture of a particular picture. Therefore, in this paper, we propose an improved

978-1-7281-6661-2/20/$31.00 ©2020 IEEE

ICycleGAN network structure by using the style transfer function of CycleGAN. Experiments have shown that this method can combine multiple styles and content more naturally and elegantly.

The main contributions of our works are summarized below:

- We proposed an improved image generation model, IStackGAN, to generate better and more realistic images through the input text. IStackGAN improved by 11.35% compared to StackGAN's Inception Score performance, and on Oxford dataset, IStackGAN improved by 9.69%.

- We propose an improved style transfer method, ICycleGAN, which can synthesize more attractive art images than traditional neural style transfer.

II. Related Work

A. Text to Image Generation

It is very difficult for a computer to generate a high-resolution, text-matching image from a text description, but this work makes sense in all aspects of engineering applications. In the early stage, variational autoencoder [6] was used to complete the task of text to image generation. With the help of automatic cyclic coding and attention mechanism, some images were drawn iteratively according to the words in the article, and even reasonable images could be generated from sentences not seen in the training set. However, the autoencoder is difficult to generate high-resolution images, and some key areas are often blurred, which cannot meet the needs of the current project. In the field of images, because the pictures produced by variational autoencoder are too fuzzy, people are usually more concerned about variational autoencoder's role as an image feature extractor.

However, since the emergence of generative adversarial networks [3], new ideas have been provided for text-to-image generation, and people have constantly improved the method to obtain higher definition and more detailed pictures. The work of Scott Reed et al. [11] proved for the first time that CGAN could generate images with a resolution of 64×64 acceptable to human senses from texts. Although there is a lack of realistic details and some parts of the objects in the images, this work has pioneered the study of CGAN generated images. Also in Scott Reed's GAWWN model [10], it is proved that the position and size of the target object in the image can help improve the quality of the generated image and the quality of the interpretation of the text and produce a high-resolution image of 128×128. In the work of Anh Nguyen et al. [8], the pre-trained classifier was used as an encoder to extract features from the image, and h was used as the initial input, and then the feature value was modified through continuous iteration to obtain a better image, generating a high-resolution image of 227×227. However, this model needs multiple inefficient iterations to optimize. Han Zhang et al.'s StackGAN [17] model breaks down the complex problem of producing high-quality images into more controlled sub-problems, and for the first time generates a high-resolution image with realistic details from a text description. Our model is based on an improvement of this model and has achieved good results in the experiment set.

B. Style Transfer

At present, most style transfer is in the combination of content and style work, generally to choose a picture as a style picture. Singular Gatys et al. [2], who initiated the study of neural style transfer, proposed a style transfer scheme based on CNN. Although this method uses pre-trained VGG networks for style transfer, which costs a lot in time, it is the first time that CNN has been successfully applied to neural style conversion. In [7], some improvements were made to the original method to change the process of style transfer into the local affine transformation of color space, and the problem of style overflow was solved by means of semantic segmentation. Then there are some new approaches to neural style transfer. In [15], a feedforward image synthesis network is proposed, which is faster than the original method. In [1], Ulyanov et al. proposed a texture network for texture synthesis and style transfer.

III. Model

A. Image Generation Model

In image generation, it is difficult to directly generate high-quality images. The IStackGAN network we proposed is trained in two stages, and the model framework of each stage is shown in Figure 2. First, we introduce some notation and problem definitions. In Figure 2, $\phi(t)$ is the text embedding of text describing t, and c is the conditional variable after conditional enhancement. z is the noise from the Gaussian distribution; L_{ssim} is the loss of image structure, and $L_{feature}$ is the loss of feature reconstruction.

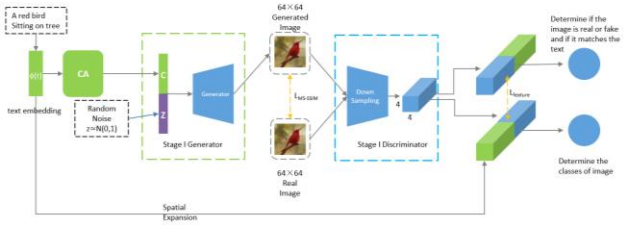

Figure 2(a). IStackGAN phase 1

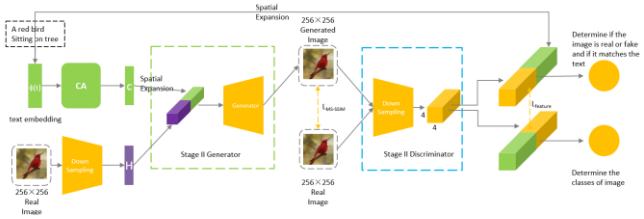

Figure 2(b). IStackGAN phase 2

The biggest difference between IStackGAN and the model in reference [17] is that we add a classifier to the output of the final discriminator to judge the category of the generated image. With current technology, computers can only produce images of the same type (e.g., birds, cats, dogs,

etc.). If there are subcategories under the same category, the neural network will get more information and the quality of the generated images will be better. Finally, we introduce the structural similarity reconstruction loss and feature loss into the loss calculation of the generator to improve the quality of the generated image.

We will explain the IStackGAN model in detail. Since we need to complete the generation from a string of text to an image, rather than a single condition (such as a bird), the image should be closely related to the text, so we use the same method as in [17] to express the text description as a text embedding. In this paper, the text encoder mentioned in article [12] is used to convert the original text into embedding text. Since $\phi(t)$ is a high-dimensional vector (with a dimension higher than 100), in the case of limited data, the latent data manifold will be discontinuous, which is not conducive to the learning of model generation. In order to overcome this problem, the conditional enhancement model proposed in literature [17] is adopted to convert the vector $\phi(t)$ into the low-dimensional conditional variable c. In the conditional enhancement model, the mean value μ_0 and the diagonal covariance matrix σ_0 are obtained from the vector $\phi(t)$ through the full connection layer. The ϵ is randomly sampled from the normal Gaussian distribution, so the conditional variable c is

$$c = \mu_0 + \sigma_0 \odot \varepsilon \tag{1}$$

At the same time, in order to increase data diversity and avoid overfitting, the generator adds the following regular terms during training.

$$D_{KL}(N(\mu_0;\sigma_0) \| N(0;1)) \tag{2}$$

It represents the KL divergence between our randomly sampled gaussian distribution and the standard Gaussian distribution, which is conducive to modeling the problem and generating more diverse images based on a fixed description.

In the first stage generator, the conditional variable c is spliced with the random noise z. After splicing the c and z, it is transformed into a tensor through the full connected layer, and then generate the image I_{f0} with a size of 64×64 through the up-sampling.

In the first stage discriminator, the text embedding $\phi(t)$ is expanded to a tensor of size $M_d \times M_d \times N_d$. Then, the real or generated image will be finally changed into a $M_d \times M_d \times N_{d1}$ tensor after feature extraction by a series of down-sampling. Then, we spliced the two vectors together in the third channel to get the feature vector which will pass into the two convolution layers respectively. Finally, Through a full connection layer, the probability D_{s0} and the probability distribution of the image category label D_{c0}, respectively, are generated to determine whether the image is real and matches the text.

In the second stage generator, the conditional variable c is spatially expanded into a tensor with a size of $M_g \times M_g \times N_g$. The image I_{f0} generated in the first stage is then down-sampling to a tensor with size of $M_g \times M_g \times N_{g1}$. After the above two vectors are spliced together in the third channel, a 256×256 high-resolution image is obtained through a series of up-sampling.

The discriminator in second phase is similar to that in first stage. The only thing is, the scale of the vector is bigger than it was in the previous phase, because the resulting picture is bigger.

While training, $\{(I_{real}, T_{real}), (I_{fake}, T_{real}), (I_{real}, T_{wrong})\}$ are respectively feed into discriminator. Among them, the I_{real} for a real image and text T_{real} matches it; I_{fake} for a fake image generated by generator; T_{wrong} for a text which mismatches an image.

In terms of loss function, discriminator loss l_{Ds} and classification loss l_{Dc} respectively

$$L_{Ds} = E_{(I_0,t)\sim p_{data}}[\log D_0(I_0, \varphi_t)] + E_{z\sim p_z, t\sim p_{data}}[\log(1 - D_0(G_0(z,c), \varphi_t))] \tag{3}$$

$$L_{Ds} = E_{(I_0,t)\sim p_{data}}[\log P(C = c \mid (I_0, \varphi_t))] + E_{z\sim p_z, t\sim p_{data}}[\log P(C = c \mid (G_0(z,c), \varphi_t))] \tag{4}$$

The loss function of the total discriminator is

$$L_D = L_{Ds} + \lambda_c L_{Dc} \tag{5}$$

In the generator loss function, L_{Gs} is the traditional generator loss, and D_{KL} is the divergence mentioned earlier in KL.

$$L_{Gs} = E_{z\sim p_z, t\sim p_{data}}[\log(1 - D_0(G_0(z,c), \varphi_t))] + \lambda_0 D_{KL} \tag{6}$$

In addition, we also bring in the image classification loss, where we use the cross-entropy loss function. In order to make the generator converge better and the generated model not deviate too much from the real sample, the feature reconstruction loss and the structure reconstruction loss are added into the loss function of the model generator. For feature of the reconstruction loss, we use the pre-trained VGG16 network as feature extraction, so as to guarantee the real images and generated images in the feature extraction is similar, where $f_D(I_{real}, T_{real})$ and $f_D(I_{fake}, l_{real})$ respectively denote feature extraction function. We use SSIM to calculate structural reconstruction and it is widely used to make up for L2 distance, which tends to reduce the diversity of models. The formula is shown in equation (7) and equation (8).

$$L_{feature} = \left\| f_D(I_r, l_r) - f_D(I_f, l_r) \right\|_2^2 \tag{7}$$

$$L_{SSIM} = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \tag{8}$$

The total loss function of the generator becomes

$$L_G = L_{Gs} + L_{Gc} + \lambda_1 L_{feature} + \lambda_2 L_{SSIM} \tag{9}$$

In equation (9), $L_{feature}$ and L_{SSIM} respectively represent the category information of the added image and the loss of the reconstruction of the feature and structure.

B. Style Transfer Model

The traditional neural style transfer algorithm [2] often results in almost all the main contents of the original pictures are filled with style pictures, and the process of style transfer of each picture is similar to a training, which takes a lot of time. Therefore, in this paper, we propose the ICycleGAN network structure to complete the task of style transfer. In [5],

G is set to use u-net [13], which is similar to the encoder/decoder architecture. However, we find it difficult to train a simple network using u-net as a generator, and the model often collapse. Based on this, in the ICycleGAN model, we improved the generator architecture on the basis of [18], as shown in Figure 3, which combined the advantages of u-net and residual-net [4]. Although the simple skip connection in u-net can extract image features well, it is also easy to crash. We think that simply skipping joins will lead to confusion in the convolutional layer, and the newly added residual block has the function of skipping joins and further encoding. Therefore, we used the residual block to optimize the skip connection in u-net.

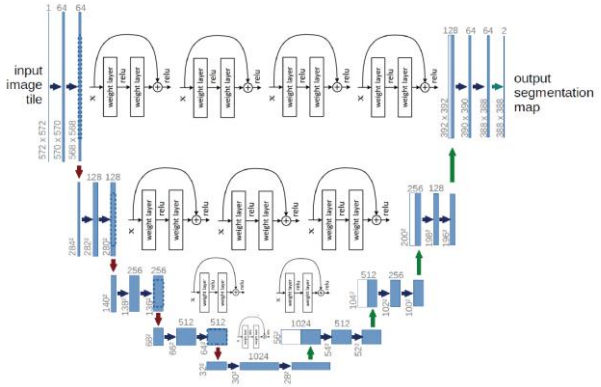

Figure 3. Improved u-net structure

In addition to improve the generator structure, in terms of the loss function, we added the cycle consistent loss with MS-SSIM on the basis of the original. The traditional L2 distance is often used for pixel-level comparisons between images, which can lead to a decrease in the diversity of generators. At the same time, since they are all pictures of the same category, its structure is similar each other, and L2 distance cannot measure the structural similarity of pictures. We need to make improvements in the cycle consistent loss. SSIM loss matches the brightness, contrast and structure information of the generated image and the input image, and it has proved very helpful in improving image quality. The improved multi-scale SSIM loss considers the multi-scale SSIM loss as follows:

$$MS-SSIM(x,y)=[l_M(x,y)]^{\alpha M}\prod_{j=1}^{M}[c_j(x,y)]^{\beta j}[s_j(x,y)]^{\gamma j} \quad (10)$$

The three functions are image brightness comparison function, image contrast comparison function and image structure comparison function.

Before instruction, we introduce some notation and problem definitions. X and Y represent the real image, and X' and Y' represent the generated image. L_{L1} represents the L1 distance loss of the real image and the generated image, while L_{ss} represents the loss of MS-SSIM structure reconstruction The improved model adds MS-SSIM loss to the cycle consistent loss to force the recovery of the similarity between the image and the original image. Therefore, for the two cycle consistency losses, we consider the structural similarity and L1 loss. $X'=F(G(X))$ and $Y'=G(F(Y))$ are respectively cycle consistent reconstruction of the input image.

$$L_{SS}=(1-MS-SSIM(X',X))+(1-MS-SSIM(Y',Y)) \quad (11)$$

$$L_{L1}=\|X'-X\|_1+\|Y-Y'\|_1 \quad (12)$$

Therefore, the cycle consistent loss with MS-SSIM is:

$$L_{cyc+ss}=L_{cyc}+\lambda_{ss}L_{ss}+\lambda_{L1}L_{L1} \quad (13)$$

Among them, L_{cyc} is the cycle consistent loss, λ_{ss} and λ_{L1} are the hyper-parameter.

IV. EXPERIMENT

In order to verify the effectiveness of the model, we conducted experiments on data sets of oxford-102 [9] and CUB-200-2011 [16] respectively. We conducted the experiment based on the following three steps. First, CUB-200-2011 dataset is used to perform the text-to-real image generation experiment on the image generation model, and then the experiment is compared with the effect in StackGAN. Then, we conducted a style transfer experiment. For the style transfer model, we chose Cezanne style, Monet style and Ukiyo-e style. Finally, we evaluate and compare our model with the original model based on baseline, such as Inception, to prove the good performance of our model.

A. Experiment Preparation

Before the experiment started, we did some processing on the data set. First, we normalize all the images in the dataset to between [-1, 1] and size them to 256×256. In addition, our experimental environment is hardware (Intel i5-9400f, 16GB memory, NVIDIA 2070 Super) and software platform (TensorFlow 2.1.0).

B. Experimental Detail

Since the overall model is a combination of the above two models, we will describe the details of model training in the following sections.

IStackGAN: the IStackGAN model is mainly used to generate real images. In practical experiments, we set the hyperparameter λ_c in formula (5) as 1, λ_0 in formula (6) as 2, and the hyper-parameter, λ_1, and λ_2 in formula (9) as 1, 1, respectively. Adam optimizer was used in the training, the learning rate was 0.001, and the number of iterations was set to 1000. As Figure 4 shows, it is obvious that our model is superior to the original StackGAN model in terms of color fineness and detail performance.

Figure 4. Performance comparison of StackGAN and IStackGAN

ICycleGAN: ICycleGAN model receives 256×256 size image synthesized from IStackGAN model, and generates different style illustrations with the pre-trained model. Among them, the experimental results are better when the hyper-parameters in equation (13) are set as 0.7 and 0.3. The Adam optimizer is also adopted, and the optimized iteration number is set to 1000. At the same time, the generator adopts the least squares loss function in the training, which makes a good improvement in the training instability, poor image quality and lack of diversity in the generative adversarial networks. Figure 5 shows some examples that our model generates.

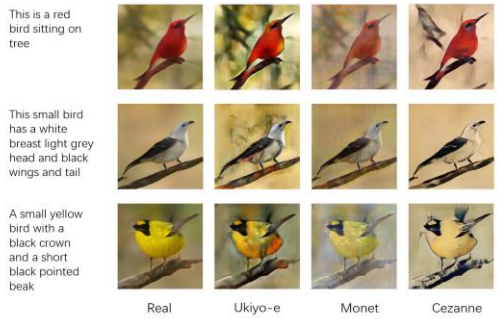

Figure 5. Display of artistic style

C. *The Results of Assessment*

In order to verify the effectiveness of the image generation model, the trained oxford-102 data set and CUB 200-2011 dataset are evaluated in the Inception Score evaluation index to quantitatively analyze the performance of our model.

TABLE I. COMPARISON BETWEEN ISTACKGAN AND SOME MODELS

Dataset	GAN-INT-CLS	GAWWN	StackGAN	IStackGAN
CUB	*2.88 ± .04*	*3.62 ± .07*	*3.70 ± .04*	***4.12 ± .03***
Oxford	*2.66 ± .03*	/	*3.20 ± .01*	***3.51 ± .03***

As can be seen from the Table I, on CUB dataset, IStackGAN improved by 11.35% compared to StackGAN's Inception Score performance, and on Oxford dataset, IStackGAN improved by 9.69%. It is also not difficult to know that the introduction of classified information and structure-level reconstruction information in generative confrontation network can indeed make the generated images more vivid and detailed.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we build a combinatorial art illustration generation model to realize the text-to-art illustration generation task. By improving some of the details in the model, such as the idea of stacking generation against the network, adding classification information and reconstructing information, we produce a better picture. At the same time, for the style transfer model, we introduced MS-SSIM cycle consistent loss and improved the structure of the generator, resulting in higher quality art illustrations. In the following work, we will first build an annotated Chinese image data set. Then continue to improve the model, in order to build an end-to-end text-to-image generation model, and optimize the model part of style migration, hoping that the results can be better improved.

REFERENCES

[1] M. Elad and P. Milanfar. Style transfer via texture synthesis. IEEE Transactions on Image Processing, 26(5): 2338-2351, May 2017

[2] L. A. Gatys, A S. Ecker, and M. Bethge. A neural algorithm of artistic style, 2015

[3] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B Xu, D. Warde-Farley, S. Ozair, A CCourville, and Y Bengio. Generative adversarial nets. In Advances in Neural Infor-mation Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 2672-2680, 2014

[4] K. He, X Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las vegas, NV, USA, June 27-30, 2016, pages 770-778. IEEE Computer Society, 2016

[5] P Isola, J. Zhu, T Zhou, and A. A. Efros Image-to-image translation with conditional adversarial networks. CoRR, abs/1611.07004, 2016.

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In 2nd International Conference on Learning Representations, ICLR 2014, Banf, AB, Canada, April 1.16, 2014, Conference Track Proceedings, 2014

[7] F. Luan, S. Paris, E Shechtman, and K. Bala. Deep photo style transfer. CORR,abs/1703.07511,2017ative networks: Conditional iterative generation of images in latent space, 2010

[8] Nguyen, J Clune,Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug and play generative networks: Conditional iterative generation of images in latent space, 2016

[9] M. Nilsback and A Zisserman. Automated fower classification over a large number of classes. In Sirth Indian Conference on Computer Vision, Graphics 8 Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008, pages 722-729 IEEE Computer Society, 2008

[10] S. Reed, Z. Akata, S Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw, 2016

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

[12] S.E. Reed, Z. Akata, H Lee, and B Schiele. Learning deep representations of fine-grained visual descriptions In 2016 IEEE Conference on Computer Vision and pattern Recognition, CVPR 2016, Las vegas, NV, USA, June 27-30, 2016, pages 49-58. IEEE Computer Society, 2016

[13] O.Ronneberger, P. Fischer, and T Brox U-net: Convolutional networks for biomedical image segmentation. CoRR, abs/1505.04597, 2015

[14] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition, 2014

[15] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feedforward synthesis of textures and stylized images, 2016

[16] C. Wah, S. Branson, P Welinder, P Perona, and S Belongie. The caltech-ucsd birds-200-2011 dataset.2011

[17] H Zhang, T. Xu, H Li, S. Zhang, X Wang, X Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks

[18] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 2242-2251, 2017