Proceedings of the 25th International Conference on Automation & Computing, Lancaster University, Lancaster UK, 5-7 September 2019

Dual Stage Augmented Colorful Texture Synthesis from Hand Sketch

Jinxuan Liu, Tiange Zhang, Ying Gao, Shu Zhang, Jinxuan Sun, Junyu Dong
College of Information Science and Engineering
Ocean University of China
Qingdao, China
dongjunyu@ouc.edu.cn

Hui Yu
School of Creative Technologies
University of Portsmouth
Portsmouth, UK
hui.yu@port.ac.uk

***Abstract*—In this paper, we investigate the texture synthesis method generated from the hand-made sketches. In recent years, GANs have been vigorously studied in the field of image synthesis and generation, yet the texture synthesis from the hand sketch has not been extensively studied. In order to enable the synthesized image not only to possess the texture features, but also to show vibrant colors, we propose a cascaded network model that can synthesize a texture image. The proposed framework firstly generates a grayscale image with basic texture properties from hand sketch based on the conditional GANs. This grayscale texture is then colorized in the second stage. The network in the second stage is pre-trained using our constructed dataset to learn how to translate the grayscale image to a colorful image. We design a series of experiments to validate the effectiveness of our method. Encouraging results are achieved. The results demonstrate that the dual stage model outperforms the state-of-art generative models in the related areas.**

Keywords-texture synthesis; cGANs; image colorization

I. INTRODUCTION

Texture synthesis has always been an important topic in the computer vision and image processing field. Every object has its own specific texture. These textures are universal and different. In many cases, some textures are irregular and random. Synthetic texture refers to the artificially synthesized realistic texture images. They have three key characteristics: (1) The synthesized texture should be visually similar to the existing texture and can successfully fool human eyes; (2) The generated texture should be random, so-called creativity of model. (3) It should be in line with human aesthetic needs. Existing synthesizing methods can be achieved by texture sample mapping method based on Markov random field or deep learning.

Most of the earlier researches on texture synthesis were achieved by an instance-based approach that synthesized new textures similar to existing textures. Kwatra et al. [1] introduced a Graph-Cut method to minimize inconsistencies among the synthetic images. Encouraging results were obtained [1]. However, the computational time was high. As deep learning becomes popular, more researches try to adopt the great advantages from deep network into texture synthesis tasks. Specifically, the texture synthesis based on Convolutional Neural Networks (CNN) has been widely explored. Gatys et al. [2] stated that besides the elegant texture production, the deep networks could also stylize an image with a single texture example.

Recently, the Generative Adversarial Network (GAN) presented by Goodfellow et al. [3] allows the research of image manipulations to enter a deeper and wider domain. GAN utilizes a loss to determine the true and false of the output image in term of real of fake. It uses this loss to train a generated model. The goal of generator G in this model is to generate a realistic picture as much as possible to deceive the discriminator D. The goal of D is to accurately distinguish the fake picture generated by G from the real picture. Accordingly, they constitute a dynamic "gaming process". Based on this concept, the introductions of various GAN networks inspire more new ideas for texture synthesis. A common method is the one based on the conditional GANs (cGANs). It adds additional condition information to both the generator and the discriminator.

The question is to make the computer to generate a texture that we need? To address this problem, Phillip Isola et al. [4] presented an approach to achieve image-to-image translation in pixel-level. Yongli Lu et al. [5] proposed a contextual GAN that used hand sketches to synthesize realistic images. In our network, the shape and contour of textures are limited by hand sketches. The color of textures is generated in the second step.

In this study, the texture synthesis is considered as an image-to-image translation process, which means the style transfer from sketch to texture. The color of texture images usually has a strong randomness. Some textures are colorful, such as flowers and clothes, and some texture are monotonous, such as trees and soil. By experiments, we found that the color had a great influence on the authenticity of synthetic texture. For example, it is unlikely that the trunks of a tree are blue, so it is likely that the pink soil is faked. Although these strange objects may actually exist, they are extremely rare in practice. It is believed that this is because the color of texture is not regular. If there are more images of trees and leaves in the dataset, green and brown will account for a great proportion in the dataset. This problem causes the network to lose a lot of color information during the training process. Inspired by the thought of image-to-image translation, we propose a new method of dual stage augmented colorful texture synthesis from sketch. We divide the texture synthesis process into two steps: The first step generates a grayscale texture image according to

This work were supported by the National Natural Science Foundation of China (Grant No.41576011) and the Shandong Provincial Natural Science Foundation, China (Grant No. ZR2018ZB0852).

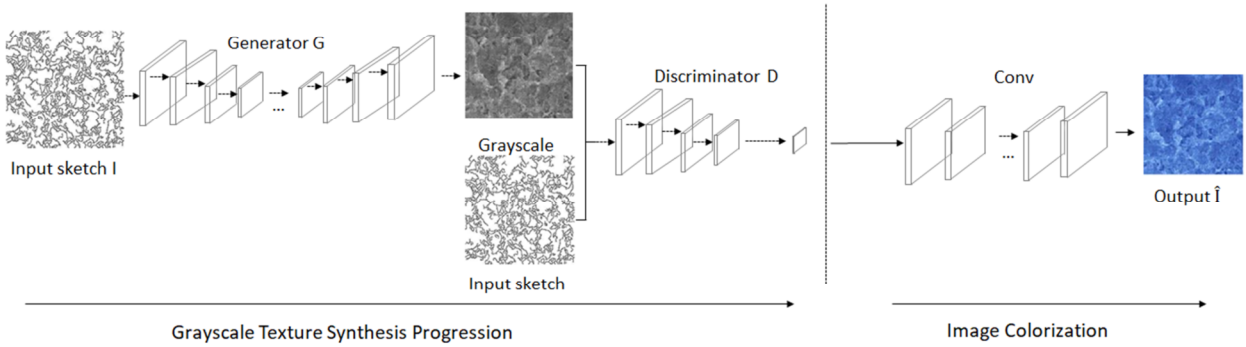

Figure 1. Architecture of our approach. Our model designed to synthesize texture by dual stage. Grayscale texture synthesis progression tries to generate realistic grayscale, which showing texture pattern, roughness, etc. Image colorization tries to color the image credibly.

the given hand sketch. Rather than considering the color information, this step focuses on texture structure and detail of the synthesized image, and enforces the authenticity of the texture. In the second step, the proposed model colorizes the image generated during the first step. This step is not to produce an accurate color in the image which is the same as the ground truth, but a realistic color as realistic as possible. It intends to obtain not only creativity but also aesthetics. The two networks are trained independently, and then merged together. Experiments demonstrate the effectiveness and superiority of our method.

II. Related Work

Texture generation has been a hot research topic since mid-1990s. In the field of computer graphics, any representations of a graphic surface can be treated as texture, including geometric information, material information, color information, etc. In general, the approaches of texture synthesis can be mainly divided into two categories as follows:

Nonparametric methods. There are two approaches to synthesize textures based on nonparametric strategy.

(1) Pixel-based models synthesize a new texture by resampling pixel of the original texture. Based on non-parametric sampling, A. Efros and T. K. Leung [4] suggested a pixel-based approach for texture synthesis. It only synthesized one texture pixel at a time, and gradually covered the entire texture image point by point. It is the earliest type of the texture synthesis algorithms. Additionally, Li Wei et al. [5] also used a nonparametric method based on a tree-structured vector quantization strategy to synthesize texture.

(2) Patch-based models compare the similarities between patches. Compared to the pixel-based method, the patch-based methods synthesize the full image at once instead of one pixel a time. For example, Kwatra et al. [1] proposed to generate image and video using graph cuts. It calculated the size and shape of the posted patch according to the graph-cut method.

However, those nonparametric methods are not able to establish an actual model to describe the natural textures. They only provide a mechanical process without changing its perceptual characteristics.

Parametric methods. These methods demonstrate better performance in synthesizing a wide range of textures. They are statistical measurements based on the filter responses rather than the image pixels [6]. For example, Gan et al. [7] introduced a perception driven texture generation approach.

In recent years, deep convolutional neural networks have been great successful in texture synthesis because of their powerful learning capabilities. Gatys et al. [2] introduced a new parametric texture model, which was based on Convolutional Neural Network. However, this method was instable, and its brightness and contrast could change drastically. Based on Convolutional Neural Networks, a multi-scale synthesis pipeline is proposed to address these issues. Based on the work of [2], Ulyanov et al. [8] presented a feed-forward convolution network which can generate multiple samples of any size. It transferred the given images into some images of artistic style.

After Generative Adversarial Networks were presented, many studies apply GANs to image synthesis [9]. For example, Mirza and Osindero extended GANs as cGANs for conditional image synthesis [10]. The image-conditional GANs have tackled the problem in product image synthesis, image generation from sparse annotations [11], digit and face image generation and many others. Wang and Gupta used a Style and Structure Generative Adversarial Network to factorize the image generation process [12]. Isola et al. [13] improved cGANs and demonstrated their wide applicability in image-to-image translation.

Nevertheless, we still need a huge database from which to retrieve images for sketch-to-image generation [14]. To address this problem, Lu et al. [15] introduced contextual GAN for image generation from hand sketch. Inspired by these works, we proposed a dual stage augmented colorful texture synthesis from a sketch model.

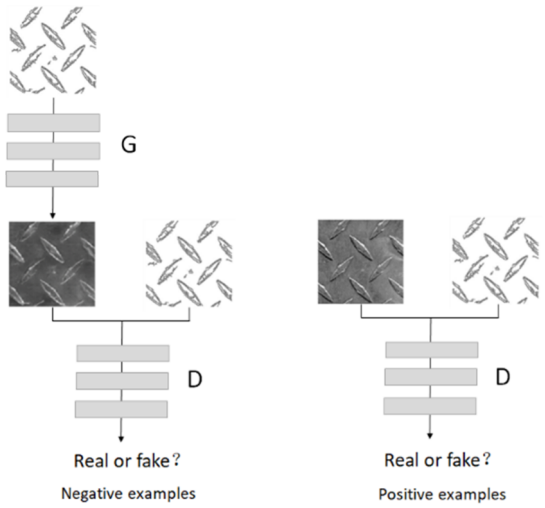

Figure 2. Train process of a conditional GAN. It learns a mapping from sketch to grayscale texture. The generator, G, tries to synthesize fake images that fool the discriminator, D. The D learns to distinguish between fake and real {sketch, grayscale} tuples. Based on cGANs, the input image is available to both G and D.

III. METHODOLOGY

In this paper, we design a sketch-to-texture translation model as a solution to the texture synthesis problem. The goal of our model is, given a texture sketch I, to synthesize a new image $\hat{I}$, where $\hat{I}$ preserves the texture features described in I, such as shape, regularity and roughness. It meanwhile has plausible color. Fig. 1 shows the pipeline of the proposed model. It consists of two stages: a grayscale texture synthesis progression and an image colorization step.

A. Grayscale Texture Synthesis Stage

In this stage, our goal is to generate a corresponding grayscale texture given a hand sketch. At this stage, we only focus on the preservation of the texture features ignoring other color factors. We use the method proposed by Isola et al. [13] to train the network. The training process is demonstrated in Fig. 2. Traditional GANs learn the mapping from a random noise vector z to an output image y, G: z→y [2]. CGAN is also a model of learning mapping that learns the mapping from an input images x and a random noise vectors z to y, G: {x, z} → y. The objective function of a cGAN is as shown in (1).

$$L_{cGAN}(\mathrm{G},\mathrm{D}) = E_{x,y}[logD(x,y)] + E_{x,z}\left[\log\left(1 - D\left(x, G(x,z)\right)\right)\right] \quad (1)$$

For image translation tasks, the input and output actually share a lot of information. Therefore, in order to ensure the similarity between the input and the output, we train the network using an L1 loss:

$$L_{L1}(G) = E_{x,y,z}[\|y - G(x,z)\|_1] \quad (2)$$

The loss function of grayscale texture synthesis stage can now be written as:

$$L_G = \arg\min_G \max_D L_{cGAN}(G,D) + \lambda L_{L1}(G) \quad (3)$$

Generator architecture. The traditional convolutional neural network causes all the layers to hold all information. An encoder-decoder, more specifically a "U-net" structure [16] is used to reduce errors. The network only uses the valid part of each convolution and does not have any fully connected layers [16]. The architecture is illustrated in Fig. 3.

Discriminator architecture. Since L1 loss can accurately capture the low frequencies, in this paper, we use PatchGAN [13], a discriminator architecture to enforce the high-frequency structure. The network cuts an image into different sizes of patches. The goal of the discriminator is to distinguish the fake N*N patch in an image from real. The average of all responses in an image is taken as the final output of discriminator.

B. Colorization Stage

In the second stage, the network is trained based on the method proposed in [17]. The reason we utilize this method is to colorize the grayscale image instead of restoring the true color of the texture image. This method uses the texture, semantics, and other features of the input to predict possible colorization results. The final image is plausible and realistic. This not only reduces the difficulty of colorization, but also satisfies people's aesthetic standards. To address the problem of automatic colorization, [17] designed an appropriate objective function to handle the multi-model uncertainty of the colorization problem and captured multiple colors. Unlike other CNN architectures, the proposed method uses a multinomial cross entropy loss, with rebalanced rare classes, and a VGG-styled network with added depth and dilated convolutions [18]. The network architecture is shown in Fig. 4.

C. Training details

In the first stage, we optimize the proposed networks using the training procedure suggested in [13]. To slow down the rate at which D learns relative to G, we divide the objective by 2 in this step. In our experiments, we apply minibatch SGD and the Adam solver to the training process with a learning rate of 0.0002. The size of input image is 256*256. In the second stage, our networks are pre-trained by our dataset. The entire training process takes approximately 8 - 10 hours.

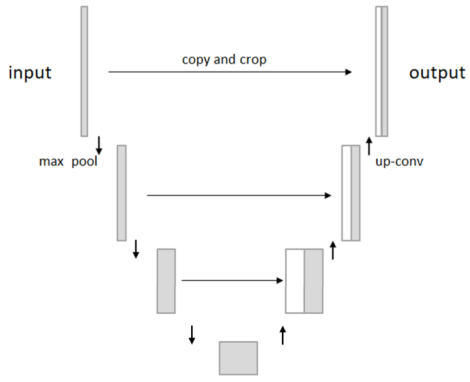

Figure 3. U-net architecture.

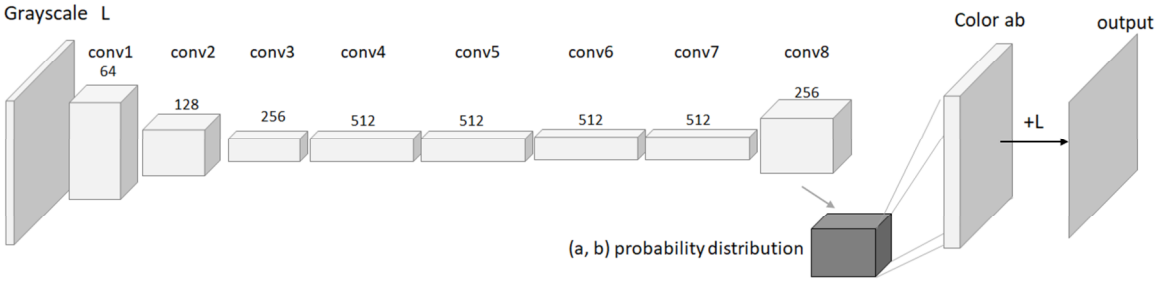

Figure 4. The network of image colorization.

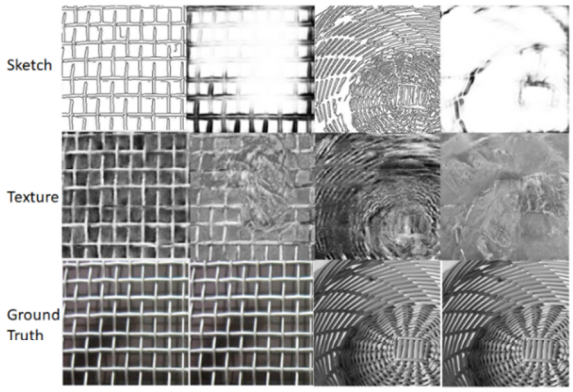

Figure 5. Comparing experiments on "HED" and "Canny". The first and third columns are the result of Canny, and the second and fourth columns are the result of the HED. The second row is the generated images from texture synthesis progressing.

IV. EXPERIMENTS

A. DataSets

The natural texture images we used in our experiments are based on Describable Textures Dataset (DTD). On DTD, there are 47 groups containing 5640 texture images. In order to further eliminate the interference of color factors in the first stage of training, we converted these 5640 pictures into grayscale images with a resolution of 256*256. The key contribution of our network's training is that it has paired data of sketches to textures maps. However, the workload of collecting thousands of hand-drawn images is too much. Therefore, we used HED edge detector and Canny edge detector to extract hand-drawn sketches instead of manual annotations. Through experimental comparisons, the latter results can fit our experimental needs better. We thus finally used the results of Canny edge detector as training data. For each run of experiment, nearly 3900 images were utilized for training and the rest for testing.

To verify the applicability and effectiveness of our network, we collected two sets of real hand drawings. Each set contains around 470 hand drawings, corresponding to 470 texture maps. These images were also used for training and testing.

B. Creation of training data

In order to implement our sketch to texture synthesis, we used two mainstream algorithms to perform edge extraction on the original texture images. The comparisons of the training results of these two data sets are shown in Fig. 5. Different from the training process used in pix2pix, we utilized sketches extracted by Canny edge detector in the training.

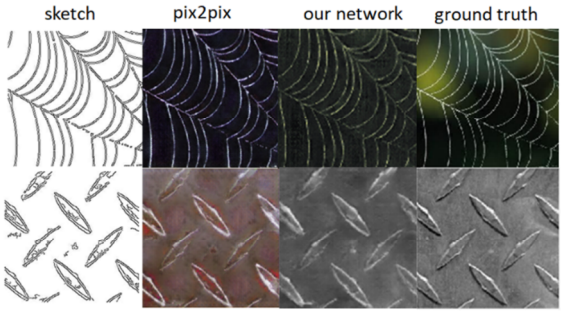

Figure 6. Example results of comparative experiments.

C. Comparison with state-of-the-art methods

Compared to the methods for image-to-image translation [13], the comparative experiments results are shown in Fig. 6. In order to measure the similarity between the generated image and the real image, we use the method of the structural similarity metric (SSIM). Results are shown in TABLE Ⅰ.

We randomly selected one hundred pairs of ground truth and synthesized textures to evaluate. The participants were expected to rate the images with a 10 points scale. The higher the score is, the closer to ground truth it is. We evaluated these images in color, structure, and authenticity. The scores are shown in TABLE Ⅱ. The experimental results proved that our network could generate greatly realistic texture image. It is superior to pix2pix. More importantly, our approach has made great strides in enhancing the color of textures compared to pix2pix. More experimental results on our network are shown in Fig. 7.

TABLE I. SSIM ON TEST SET

Method	***Pix2pix***	***Ours***
SSIM	0.6494	0.8584

TABLE II. EVALUATION RESULTS

	Color	**Structure**	**Authenticity**
pix2pix	8.03	9.40	8.89
Ours	9.21	9.58	9.36

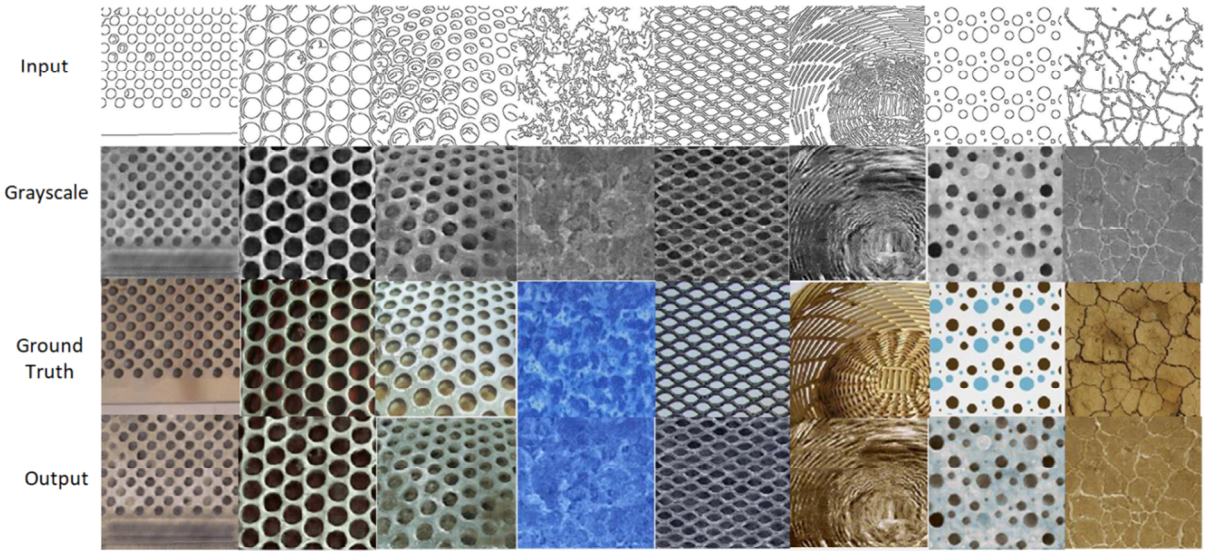

Figure 7. From top to bottom: input sketch, image of texture synthesis progression, ground truth, output of our network.

D. *Results on real hand sketch*

In order to verify the practicality of our method, we conduct the experiments that are shown in Fig. 8 on the real hand sketch. The experiments demonstrate that our method can successfully synthesize textures based on the real hand sketch. From the results, it can be noted that our method not only captures the important details from the input sketch, but also exhibits some degree of freedom in the appearance.

V. CONCLUSION

In this paper, we propose a dual stage deep learning model for hand sketch to colorful texture synthesis problem. Comparing our approach against the previous methods, ours is more robust. The experimental results demonstrate that compared to the one-stage model, the synthesized texture images of dual stage are more realistic and closer to ground truth.

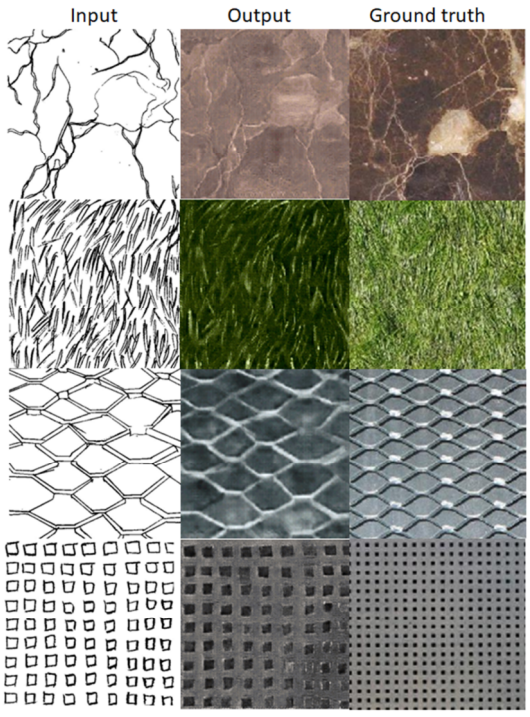

Figure 8. Results on real hand sketch dataset.

REFERENCES

[1] V. Kwatra, A. Schödl, and I. Essa, "Graphcut textures: image and video synthesis using graph cuts," in *ACM Transactions on ...*, 2003.

[2] L. A. Gatys, M. Bethge, and A. S. Ecker, "Texture synthesis using convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, pp. 262–270, 2015.

[3] I. Goodfellow *et al.*, "Generative Adversarial Nets (NIPS version)," *Adv. Neural Inf. Process. Syst. 27*, 2014.

[4] A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, pp. 1033–1038, 1999.

[5] L. Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," *Proc. ACM SIGGRAPH Conf. Comput. Graph.*, pp. 479–488, 2000.

[6] J. Portilla and E. P. Simoncelli, "A Parametric Texture Model Based on Joint . . .," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49--71, 2000.

[7] Y. Gan, H. Chi, Y. Gao, J. Liu, G. Zhong, and J. Dong, "Perception Driven Texture Generation," no. July, pp. 889–894, 2017.

[8] D. Ulyanov, V. Lempitsky, V. Lebedev, and A. Vedaldi, "Texture networks: Feed-forward synthesis of textures and stylized images," *33rd Int. Conf. Mach. Learn. ICML 2016*, vol. 3, pp. 2027–2041, 2016.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation learning with Deep Convolutional GANs," *Int. Conf. Learn. Represent.*, 2016.

[10] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," pp. 1–7, 2014.

[11] S. Reed, Z. Akata, B. Schiele, S. Tenka, S. Mohan, and H. Lee, "Learning what and where to draw," *Adv. Neural Inf. Process. Syst.*, no. Nips, pp. 217–225, 2016.

[12] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *Lecture Notes in Computer Science*, 2016.

[13] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 5967–5976, 2017.

[14] Q. Yu, F. Liu, T. M. Hospedales, T. Xiang, Y.-Z. Song, and C. C. Loy, "Sketch Me That Shoe," 2016.

[15] Y. Lu, S. Wu, C. K. Tang, and Y. W. Tai, "Image Generation from Sketch Constraint Using Contextual GAN," in *Lecture Notes in Computer Science*, 2018.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science*, 2015.

[17] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *Lect. Notes Comput. Sci.*, vol. 9907 LNCS, pp. 649–666, 2016.

[18] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," 2015.