Artistic Image Generation from Sketch by Using Conditional Adversarial Network and Style Feature Transform

Shuaibin Zhang, Haoran Su, Tangbo Liu, Xin Fu
Electronic Information School, Wuhan University, China

Abstract

Creating beautiful artwork is a tough task for non-artists. With the evolution of convolutional neural networks, stylizing a picture comes into reality. However, few works focus on the problem where the input contains little information such as sketch. In our work, we propose a new method of automatically generating artistic picture from casual sketch using conditional adversarial network (cGAN) and style feature transform. We train a deep generative model to learn the distribution of sketch and the corresponding landscape images. Subsequently, we apply pre-trained arbitrary style transfer model to the previous output and obtain the artistic image. According to our experimental results, our method could produce very aesthetically pleasing images with high efficiency.

1. Introduction

Human beings possess a great cognitive capability of comprehend black-and-white sketches. Our mind can create realistic colorful images when see black-and-white edges. However, it demands a great artistic talent to choose appropriate colors and painting styles to create beautiful artistic works. It is not easy for ordinary people to do so. How to automatically turn the sketch into artistic images is a useful application for digital entertainment. In this work, we are interested in solving this problem by employing deep neural networks with designed constraints to transfer the line draft into specific artistic style. Practically, the new model can make up for ordinary peoples artistic talent and even inspire the artists to create new images. Ideally, readers may have the freedom to choose to generate different styles of artwork based on their own tastes.

Artistic image generation from sketches could be regarded as an image synthesis problem. Recently, numerous image synthesis methods based on deep neural networks have emerged [2, 8]. These methods can generate detailed images of the real-world, such as faces, bedrooms, chairs and handwritten numbers. As the photo-realistic images are full of sharp details, the results may suffer from being blurry, noisy and objects being wobbly. Especially the sketch-to-image problem, the control signals are relatively sparse, more ill-posed than colorization problem based on grayscale, it requires a model to synthesize image details beyond what is contained in the input. The network should learn the high frequency textures for the details of the scene elements as well as high-level image styles[7]. Besides, generating aesthetically pleasing artistic image requires more detailed texture and style. So far, the popular generative model could hardly generate high quality realistic image, let alone artistic images.

In this project, we propose a novel way to generate artistic picture from casual sketch. Instead of training an end-to-end generative adversarial network, we first reconstruct realistic image from sketch using a trained conditional adversarial network (cGAN). To overcome the defects of the generated image, we perform arbitrary style transfer afterwards. Therefore, we could obtain beautiful stylized artistic image. Due to our experimental tests, our work could produce high quality artwork from casual sketch with effectively. Code and online demo is available[1].

2. Related Work

2.1. Generative Adversarial Networks

Generative adversarial networks (GANs) were recently regarded as a breakthrough in machine learning [2], it consists of two 'adversarial' modules: a generative module G that captures the data distribution, and a discriminative module D that estimates the probability that a sample came from the training data rather than G. Both G and D could be deep neural networks.

In image synthesis with GANs, the generator attempts to produce a realistic image from an input random vector to fool the simultaneously adversarial trained discriminator, which tries to distinguish whether its input image is from the training set or the generated set. It corresponds to a minimax two-player game. The generator has benefited from convolutional decoder networks, it can go back to the work

[1] https://github.com/mtobeiyf/sketch-to-art

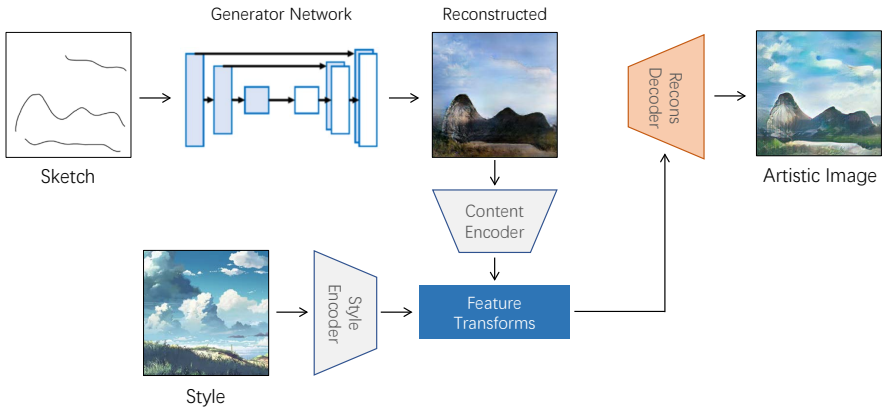

Figure 1. The framework of our proposed method. The sketch is reconstructed with our trained generative model, and is encoded together with the style picture. Using feature transform and we could get the artistic image.

using deep convolutional decoder networks to generate realistic images.

2.2. Conditional GANs

GANs are unconditioned generative models that learn a mapping from random noise vector z to output y: $G : z \rightarrow y$. In contrast, conditional GANs (cGANs) learn a mapping from observed input x and random noise vector z, to y: $G : \{x, z\} \rightarrow y$ [4].

Several works studied different cGAN where the generator is conditioned on inputs such as text, labels and other forms of the images to generate 'fake' images. The discriminator needs to distinguish the authentic and fake pairs of images. In this paper, both of generator and discriminator are conditioned on the input sketch in order to get a better supervised performance.

Image-to-image The 'pix2pix' method proposed in [4] is a 'U-net' architecture [9] to deal with general image-to-image transfer which allows the decoder to be conditioned on encoder layers to get more information. They investigated different kinds of image-to-image transfer tasks that include transforming the image of daylight to night, produce city images from map, and even synthesize shoes and handbags from designer's sketches as shown in Figure 2.

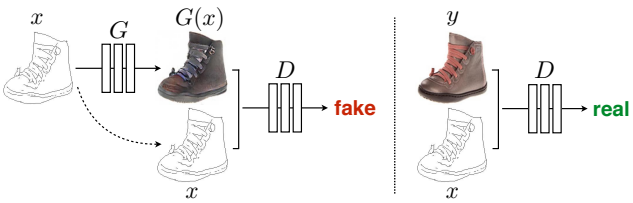

Figure 2. Training a conditional GAN to map edges→photo. The discriminator, D, learns to classify between fake (synthesized by the generator) and real {edge, photo} tuples. The generator, G, learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe the input edge map.

2.3. Style Transfer

Style transfer is an important editing task, which preserves some notion of the content image and carries characteristics of the style image. The key challenge is how to extract effective representations of the style and then match it in the content image. Gatys et al.[1] firstly introduce deep neural networks to solve the problem by Gram matrix or covariance matrix. Since then, significant efforts have been made to synthesize stylized images by minimizing Gram/covariance matrices based loss functions, through either iterative optimization or trained feed-forward networks. In these methods, there exists a problem that how to balance among generalization, quality, and efficiency, which means that optimization-based methods can handle arbitrary styles with pleasing visual quality but at the expense of high computational costs, while feed-forward approaches can be executed efficiently but are limited to a fixed number of styles or compromised visual quality.

To address the issue, Li et al.[5] propose a simple yet effective method by a pre-trained decode-encode network and the whitening and coloring transforms(WCT). It can generate high-quality stylized images of arbitrary style at a very fast speed. More details of the method are described in next section.

3. Method

Our method contains mainly two steps: (1) generating real image using cGAN and (2) stylize the output of arbitrary style.

3.1. Image Synthesis using cGAN

3.1.1 Objective

The objective of a conditional GAN can be expressed as

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))] \quad (1)$$

where G tries to minimize this objective against an adversarial D that tries to maximize it, i.e. $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D)$.

Using L1 distance is proved to be more effective at producing less blurry result rather than L2[4]:

$$\mathcal{L}_{L1} = \mathbb{E}_{x,y,z}[\|y - G(x,y)\|_1] \quad (2)$$

So, the final objective of a cGAN is:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G) \quad (3)$$

Thus, the training process becomes optimizing the give objective function above.

3.1.2 Network Architecture

Generator In the original work in [4], the authors choose "U-Net"[9] as the the basic structure of encoder-decoder network and add skip connections between each layer i and layer $n - i$, where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer $n - i$. Here, we use this structure as our image generator. Figure 3 shows the comparisons between traditional encoder-decoder and the "U-Net".

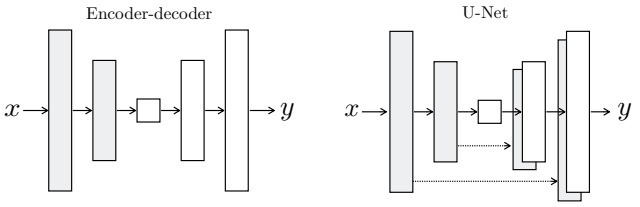

Figure 3. The "U-Net" is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

Discriminator We use a *Patch*GAN – that only penalizes structure at the scale of patches. This discriminator tries to classify if each $N \times N$ patch in an image is real or fake. We run this discriminator convolutationally across the image, averaging all responses to provide the ultimate output of D. Such a discriminator effectively models the image as a Markov random field, assuming independence between pixels separated by more than a patch diameter. Therefore, the PatchGAN can be understood as a form of texture/style loss.

3.2. Style Feature Transform

In our method, style transfer is formulated an image reconstruction process coupled with whitening and coloring transformation. The reconstruction part is responsible for inverting features back to the RGB space and the feature transformation matches the statistics of a content image to a style image. The single-level stylization is shown in Figure 4.

3.2.1 Reconstruction Decoder

This paper employ the VGG-9 [10] as the encoder and train a decoder network simply for inverting VGG features to the original image. To evaluate with features extracted at different layers, they select feature maps at five layers of the VGG-19, $Relu\_X\_1 (X = 1, 2, 3, 4, 5)$, and train five decoders accordingly. And thee reconstruction loss and feature loss are defined:

$$L = ||I_o - Ii||_2^2 + \lambda ||\Psi(I_o) - \Psi(I_i)||_2^2 \quad (4)$$

where I_i, I_o are the input and output image. And Ψ is the VGG encoder that extracts features. λ controls the balance between the loss. After training, the decoder is fixed and used as a feature inverter.

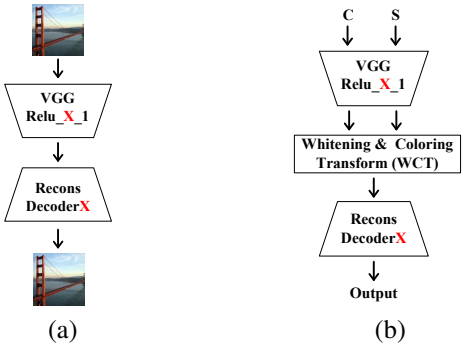

Figure 4. Universal style transfer pipeline. (a) Reconstruction, (b) Single-level stylization.

3.2.2 Whitening and Coloring Transforms

Given a pair of content image I_c and style image I_s, this method first extract their VGG feature maps $f_c \in \Re^{C \times H_s W_s}$ and $f_s \in \Re^{C \times H_s W_s}$ at a certain layer, where W_s, H_s, C are the width height, channels number of feature map. After extracting features, they use a use a whitening and coloring transform to directly transform the f_c to match the covariance matrix of f_s. At last, they construct the image by feeding the transformed feature into the reconstructed decoder.

Whitening transform is a linear transformation that transforms a vector of random variables with a known covariance matrix into a set of new variables whose covariance is the identity matrix, meaning that they are uncorrelated and each have variance. Before whitening, they first center f_c by subtracting its mean vector m_c, then the transform f_c can be defined as

$$\hat{f}_c = E_c D_c^{-\frac{1}{2}} E_c^T f_c \tag{5}$$

where D_c is an diagonal matrix with the eigenvalues of the covariance matrix $f_c f_c^T \in \Re^{C\times C}$ and E_c is the corresponding orthogonal matrix of eigenvectors, satisfying $f_c f_c^T = E_c D_c E_c^T$

Coloring transform is the inverse of the whitening transform. They first center f_s by subtracting its mean vector m_s and then carry out the coloring transform. They obtain $\hat{f}_{cs}$ which has the desired correlations between its feature maps ($\hat{f}_{cs}\hat{f}_{cs}^T = f_s f_s^T$)

$$\hat{f}_{cs} = E_s D_s^{\frac{1}{2}} E_s^T \hat{f}_c \tag{6}$$

where D_s is the diagonal matrix with the eigenvalues of the covariance matrix $f_s f_s^T \in \Re^{C\times C}$ and E_s is the corresponding orthogonal matrix of eigenvectors. Finally, they re-center the $\hat{f}_{cs}$ with the mean vector m_s of the style.

After the WCT, they may blend $\hat{f}_{cs}$ with the content feature f_c before feeding it to the decoder:

$$\hat{f}_{cs} = \alpha \hat{f}_{cs} + (1-\alpha) f_c \tag{7}$$

where α serves as the style weight for users to control the transfer effect.

3.2.3 Multi-level Coarse-to-fine Stylization

As we all know, different layers of convolutional neural network describe different features of images. The higher layer features capture more complicated local structures, while lower layer features carry more low-level information. Thus in this paper, they propose a coarse-to-fine stylized method. They start by applying the WCT on $Relu\_5\_1$ features to obtain a coarse stylized result and regard it as the new content image to further adjust features in lower layers. And the multi-level stylization pipeline is shown in Figure 5.

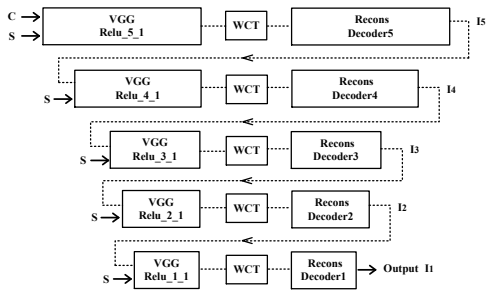

Figure 5. Multi-level stylization

4. Experimental Results

4.1. Sketch to Image Synthesis

We adopt the conditional adversarial network proposed in [4] to transform the sketch to real image.

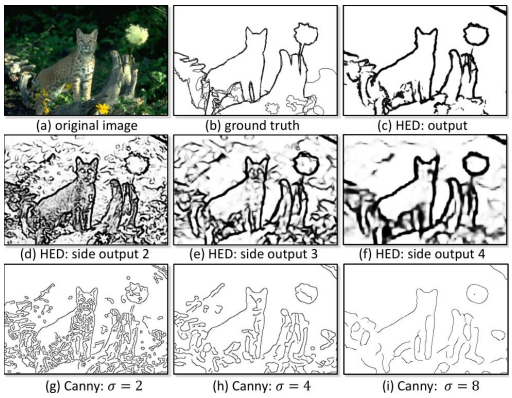

Figure 6. Example output of HED edge detection model

Dataset To better meet the requirements that the cGAN could generate much more generic real image, we choose to use landscape image as our dataset. We carefully select a high quality landscape subset of CUHKPQ for image quality assessment, where 817 natural landscape images are included.

Afterwards, we have to generate "sketches" from real images which will later be used for training. Here, instead of using traditional image edge detector like Canny operator, we utilize a deep learning-based method called holistically-nested edge detection (HED), which performs image-to-image prediction by means of a deep learning model that leverages fully convolutional neural networks and deeply-supervised nets. HED gives us much better result than simple edge detection operator.

Training

The training curve is shown in Figure 7. We set the batch size to 1 and the learning rate is 0.0002. The process took about 20 hours with 330,000 steps on the entire training set. The L1 loss $\mathcal{L}_{L1}$ keeps dropping indicates the generated image is becoming more and more alike to the original picture.

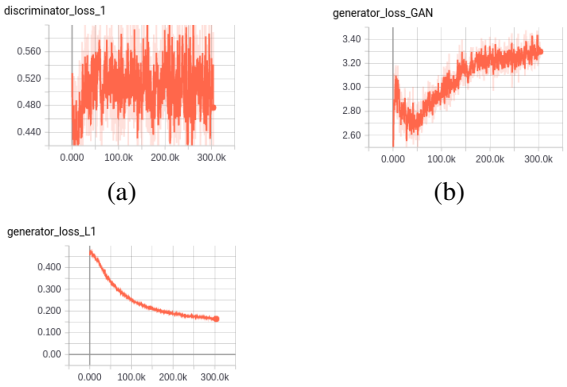

Figure 7. Training losses. (a) discriminator loss, (b) generator loss GAN, (c) generator loss L1

With the model trained on the dataset for 400 epochs, the quality of the generated image is similar to a real landscape image, as shown in the Figure 8. We could see that the input images contain little information but the edges concerning the landscape. The generator model learned the mapping from edge to real image. Therefore, we could exploit this model to generate fake landscape image from sketches.

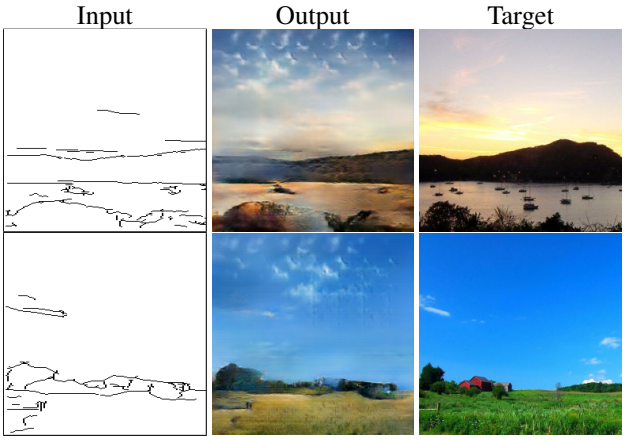

Figure 8. Sample output of our trained cGAN. The first column is the input edges, the sectond is the generated image and the last column is the target image.

4.2. Arbitrary Style Transfer

We directly take the pre-trained model from [5], which have been trained on large scale images[6] so the models are able to extract effective features concerning the content and the style from arbitrary images. Figure 9 shows examples of style transfer using the model.

Furthermore, we compare the reconstructed results with different α which controls the style weight for users to control the transfer effect. And the results is as shown in Figure 10.

Figure 9. Examples of arbitrary style transfer.

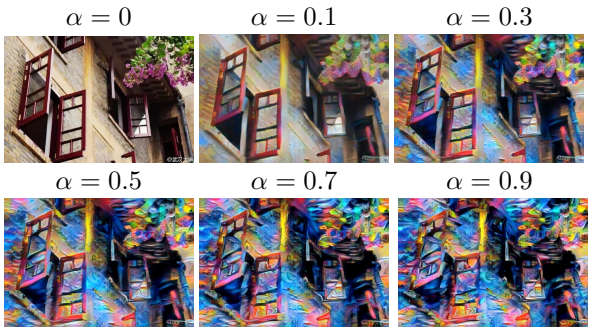

Figure 10. Style feature transform with different α.

Also, we notice that using WCT as the style feature could produce image that has stronger style while using adaptive instance normalization features from [3], the result has more detailed style texture. Therefore we provide choices for users to choose which feature to employ so as to meet their requirements.

4.3. Combined Results

By using the generated image from cGAN model and the style transfer model, we could generate brand new artwork from user's sketches. Figure 11 shows some examples of the generated and stylized images. It demonstrates that applying different styles to the reconstructed image could generally overcome the defects and noise produced by the generative model, which results in better output quality.

To better demonstrate our work, we also build a online web application which enables user draw their own sketch and choose styles so that they could create their unique artistic work.

4.4. Efficiency

We use a CPU-based computer to host the service, which has 4 processors (Intel(R) Xeon(R) CPU @ 2.50GHz) and 3.5Gb memory.

To further reduce the time consumption, we set both the input and output size to 256×256. Typically, the process of

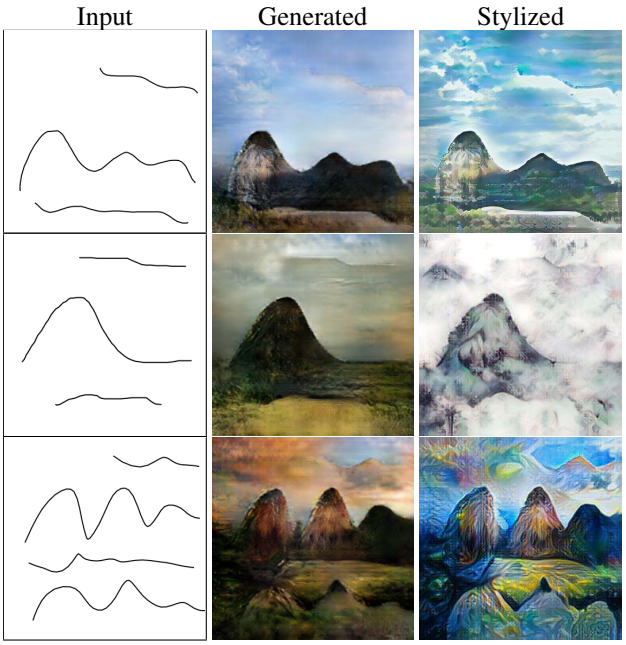

Figure 11. Sample output of our combined models. The first column is the input sketches, the second is the generated real image and the last column is the stylized artistic images.

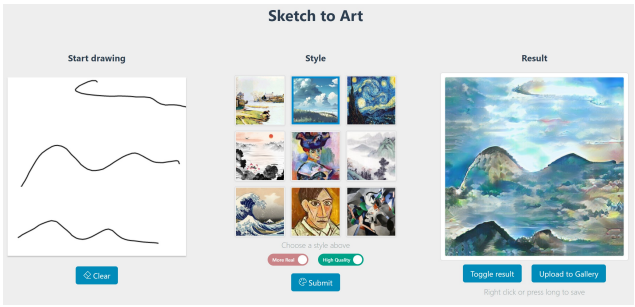

Figure 12. Screenshot of the online application.

generating realistic image from sketch takes 3 seconds and the stylization takes 8 seconds.

5. Conclusion

In this project, we propose a novel way to generate artistic picture from casual sketch. Instead of training an end-to-end generative adversarial network, we first reconstruct realistic image from sketch using a trained conditional adversarial network (cGAN). To overcome the defects and noise of the previously generated image, we perform arbitrary style transfer afterwards. Therefore, we could obtain beautiful stylized artistic image. According to our experimental tests, our work could produce high quality artwork from casual sketch effectively.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. A Neural Algorithm of Artistic Style. *arXiv:1508.06576 [cs, q-bio]*, Aug. 2015.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014.

[3] X. Huang and S. Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. *arXiv:1703.06868 [cs]*, Mar. 2017.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004 [cs]*, Nov. 2016.

[5] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal Style Transfer via Feature Transforms. *arXiv:1705.08086 [cs]*, May 2017.

[6] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*, May 2014.

[7] Y. Liu, Z. Qin, Z. Luo, and H. Wang. Auto-painter: Cartoon Image Generation from Sketch by Using Conditional Generative Adversarial Networks. *arXiv:1705.01908 [cs]*, May 2017.

[8] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, Nov. 2015.

[9] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*, May 2015.

[10] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Sept. 2014.