CPTR: Cartoon portrait style transfer based on StyleGan V2

Xiao Yang 31520221154251[*], information school class
Chi Huang 31520221154184[*], information school class
Yuyao Zhou 31520221154242[*],information school class
Changli Wu 31520221154187[*],information school class
Suhuang Wu 31520221154188[*], information school class

Abstract

Style transfer is to use some means to convert an image from the original style to another style while ensuring that the content of the image does not change. In this paper, we mainly focus on sample-based portrait style transfer, a core problem that aims to transfer the style of sample artistic portraits to the target face. There is a problem with previous works: a lot of energy and computing resources have to be spent to retrain a GAN model if you want to change a new style, which is a very cumbersome process. Inspired by Tonnify's work, we fine-tune another stylized StyleGAN on a new style dataset with a pre-trained real StyleGAN and obtain the style-transferred image by interpolating StyleGAN, which can greatly reduce the training cost of the model. In addition, we also introduce the CLIP model to guide the generation of face images with different expressions by parsing text information. The experimental results show that the model is able to accomplish the given task and generate cartoon face pictures with the corresponding styles. However, we also notice that the generated images still have some defects, such as mottled skin tones and corrupted face structure.

Introduction

Artistic portraits are popular in our daily lives, especially in industries related to comics, animation, posters, and advertising. Neural style transfer aims at transferring the artistic style from a reference image to a content image. Starting from (Gatys, Ecker, and Bethge 2015), numerous works based on iterative optimization and feed-forward networks improve style transfer from either visual quality or computational efficiency. Despite tremendous efforts, these methods do not generalize well for multiple types of style transfer. Universal style transfer (UST) is proposed to improve this generalization ability. The representative UST methods include AdaIN (Huang and Belongie 2017), WCT(Li et al. 2017), and Avatar-Net(Sheng et al. 2018). These methods are continuously extended by . While achieving favorable results as well as generalizations, these methods are limited to disentangling and reconstructing image content during the stylization process.

In recent years, some new progress has been made in the field of style transfer. GNR proposes a simple and effective definition of style and content and introduces an adversarial loss that guarantees mapping diversity.Artlns proposes a novel unsupervised algorithm, which obtain different artistic style components from the latent space consisting of different style features and generate fresh styles by linear combination according to various style components.

We are interested in using StyleGAN for style migration tasks. Stylgan is a kind of confrontation generation network, which can generate high-quality face images. Its main components include generator, discriminator, and style vector. The generator is a depth convolution neural network, which maps random vectors (also known as noise) to the generated image. The generator is a convolutional neural network, which contains many convolutional layers and pooling layers. The discriminator is also a depth convolution neural network, which is used to distinguish the real image from the generated image. The discriminator receives an image as input and attempts to predict whether the image is real. StyleGAN uses a style control module in the generator, which allows users to control various appearance features of the generated image, such as age, gender, skin color, etc. The style control module achieves this by multiplying the input vector and the middle layer output of the generator. In this way, users can control the appearance characteristics of the generated image without changing the structure of the generator or training data. How to improve the effect of the image generated by this structure is a problem worth discussing.

For style transfer, there are some problems in using the GAN model for style conversion: if you want to change to a new style, you need to spend a lot of energy and computing resources to retrain a GAN model, which is very tedious. In order to solve these problems, we were inspired by Tony's work. We used the pre trained real StyleGAN to fine tune another stylized StyleGAN on the new stylized dataset, and interpolated the StyleGAN to get the pictures after style migration. This can greatly reduce the training cost of the model. Using the pre trained StyleGAN can ensure that the model has learned some basic image generation skills before training, so that it is not necessary to start training from scratch when fine-tuning the new stylized model. In addition, by using the interpolation method, you can make a smooth transition between different styles, rather than directly converting from one style to another. This can avoid a great loss of image quality and make the effect of style transfer more

[*] These authors contributed equally.

natural.

We use Flickr-Faces-HQ (FFHQ) (Karras, Laine, and Aila 2019) as the source domain dataset to train stylegan2 model (Karras et al. 2020b) of 256 resolution. For the target domain dataset, we conduct experiments on a variety of datasets, including Naver Webtoon (Back 2021), Metfaces (Karras et al. 2020a), and Disney (Pinkney and Adler 2020). We perform the style transfer task on these datasets, and we also do some interesting experiments, such as style mixing and style clip. The former mixes two different face images from the same style, while the latter uses text information to generate a face image with a certain attribute we want, such as anger. The experimental results show that the model is able to accomplish the given task of generating cartoon face pictures in the appropriate style. However, we also notice that there are still some areas for improvement in the generated images, such as the generated faces may have mottled skin tones or the face structure is corrupted, i.e., the generated cartoon faces do not match the origin face structure, etc.

Related Work

Image to image translation (I2I) involves learning a mapping between two different image domains. In general, we want the translated image to maintain certain image semantics from the original domain while obtaining visual similarities to the new domain. Early works on I2I involves learning a deterministic mapping between paired data. This was later extended to a multimodal mapping in BicycleGAN(Zhu et al. 2017). However, due to the limited availability of paired data, this approach is cannot scale up to bigger unpaired datasets. The pioneering work of CycleGAN solves this problem by employing the use of cycle consistency to learn image to image translation for unpaired data. StarGANv2 (Choi et al. 2020)employs a single generator to produce diverse images for multiple domains. Mode Seeking GAN builds on top of DRIT and encourages output diversity by penalizing output images that are similar to each other when their input style codes are different.

Image style transfer is also a long standing research topic. Before deep neural networks are applied to the style transfer, several algorithms based on stroke rendering (Hertzmann 1998), image analogy, and image filtering are proposed to make artistic style transfer. These methods usually have to trade-off between style transfer quality, generalization, and efficiency. Gatys et al. introduce a Gram loss upon deep features to represent image styles, which opens up the neural style transfer era. Inspired by Gatys et al., numerous neural style transfer methods have been proposed. We categorize these methods into one style per model, multi-style per model, and universal style transfer methods with respect to their generalization abilities.

Neural flows refer to a subclass of deep generative models, which learns the exact likelihood of high dimensional observations (e.g., natural images, texts, and audios) through a chain of reversible transformations. As a pioneering work of neural flows, NICE is proposed to transform low dimensional densities to high dimensional observations with a stack of affine coupling layers. Following NICE, a series of neural flows, including RealNVP, GLOW, and Flow++(Ho et al. 2019), are proposed to improve NICE with more powerful and flexible reversible transformations. The recently proposed neural flows(Ma et al. 2019) are capable of synthesizing high-resolution natural/face images, realistic speech data (Prenger, Valle, and Catanzaro 2019), and performing makeup transfer. In comparison, BeautyGlow(Chen et al. 2019) shares the similar spirits but is not applicable for unbiased style transfer.

In general, current multi-modal frameworks lack a proper definition of style and content; it is unclear what exactly they each constitute. Also, visual inspection of their outputs reveals mode collapse. For a given image, the multiple outputs look very similar, often with just color and slight stylistic changes. One recent work in CouncilGAN(Nizan and Tal 2020) enables diverse outputs by collaborating between multiple GANs. However in the difficult setting of selfie2anime, CouncilGAN cannot capture the complex artistic style of animes, collapsing to few modes and not expressing the stylistic diversity we expect. Very recently, AniGAN (Li et al. 2022) proposes new normalizations to allow selfie2anime by transferring color and textual styles while maintaining global structure. AniGAN generates multimodal outputs based on reference images. Like previous methods, AniGAN does not have explicit style diversity and lack output diversities.

Method

Fine-tune SytleGANv2

In other tasks before, FreezeD (Mo, Cho, and Shin 2020) is used to freeze the lower layers of the network when training generators, and fine tune the upper layers. In this case, the freezing discriminator can help the generator better capture the complexity of data distribution, because the discriminator no longer suppresses the generator. After the discriminator is frozen, the generator can try to generate more complex images, and the discriminator will no longer try to suppress these images. However, freezing the discriminator may lead to the degradation of the generated image quality, because the discriminator can no longer help the generator learn the details of data distribution.

FreezeG is inspired by freezeD, which is just the opposite of freezeD. FreezeG freezes the shallow layers of the generator generator in the training process, fine tunes the deeper layers of the generator, and finds that it can also achieve very good results. The reason may be that the freezing generator can help the discriminator better learn the details of data distribution, because the discriminator is no longer interfered by the generator. After freezing the generator, the discriminator can try to distinguish the real image from the generated image more accurately.

We noticed that FreezeG plays a similar role in generating images to source images, and we refer to (Back 2021) and find that the factors that determine the generated image involve injecting the style vector into the generator in addition to the early layers of the generator. Therefore, in the process of fine-tuning styleGAN2, we froze the generator's initial fast and initial style vectors.

The validity and reliability of their structure has been

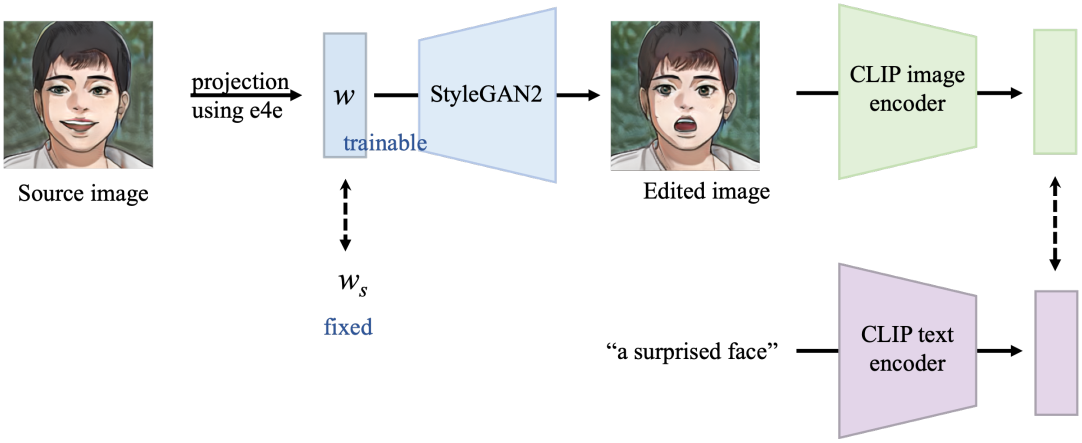

Figure 1: The architecture of StyleCLIP (using the text prompt "a surprised face", in this example).

proven in FreezeSG, so we also use their loss function, which is divided into two parts, Adversarial Loss and Structure loss.

$$\mathcal{L}_{adv} = E_{x \sim p_{data}}[\log D(x)] + E_{z \sim p(z)}[\log(1 - D(G(z)))] \tag{1}$$

$$L_{structure} = \sum_{k=1}^{n} E\left[G_{l=k}^{s}\left(w_s\right) - G_{l=k}^{t}\left(w_t\right)\right] \tag{2}$$

In the structural loss, similar to stylegan2, the input/output skipping structure is adopted in the three structures of MSG-GAN, input/output skipping, and residual network, and the architecture is simplified by upsampling and suming the contribution to the corresponding RGB outputs of different resolutions. Because the structure of the image is determined at low resolution, we apply the structure loss to the values of the low-resolution layer so that the resulting image is similar to the image in the source domain. The structure loss causes the source generator's RGB output to be fine-tuned during training to have a similar value to the target generator's RGB output.

Style Mixing

Style mixing is a technique used to train neural networks that can be used to generate new images that incorporate elements of two different styles of images.The basic steps to implement style mixing are as follows: First, select two style images: the first image is the primary style you want to use, and the second image is the secondary style you want to mix. Using these two style of images, train a neural network model so that it can classify input images into different styles. This is typically done using a convolutional neural network (CNN), which contains a series of convolutional and pooling layers that extract features from the image. Using an image as input and a corresponding style label as output to rain the model using style images. Using the trained model, run the Neural Style Transfer algorithm on two style images. This generates two new images that represent the content and style of the two style images. Then, merge two new images together to create a new image. This can be done by adjusting the proportions of the two images to adjust the proportions of the individual styles in the blended image. If the classification results are not as expected, you can adjust the proportions of the individual styles in the image and run the neural style conversion algorithm again until you get satisfactory results.

StyleCLIP Text-Driven Manipulation

We begin with a simple latent optimization scheme, where a given latent code of an image in Style -Gan's W+ space is optimized by minimizing a loss computed in CLIP space. Despite it's useful, it still needs several minutes to perform a single manipulation, and the method can be difficult to control. So a more stable approach is used, where a mapping network is trained to infer a manipulation step in latent space. In a single forward pass, training takes a few hours. But it must only be done once per text prompt. The direction of the manipulation step may vary depending on the starting position in W+, which corresponds to the input image, which we named local mapper.

The overall architecture is shown in Figure 1. Previous work(Karras, Laine, and Aila 2019) has shown that different StyleGAN layers are responsible for different levels of detail in the generated images. Therefore, it is common to divide these layers into three groups (coarse, medium, and fine) and use different parts of the (expanded) latent vector as input for each group. We design the mapper accordingly, with three fully connected networks, one for each group/section. The architecture of these networks is the same as that of the StyleGAN mapping network, but with fewer layers (4 instead of 8 in our implementation). Denoting the latent code of the input image as $w = (w_c, w_m, w_f)$, the mapper is defined by:

$$M_t(w) = (M_t^c(w_c), M_t^m(w_m), M_t^f(w_f)) \tag{3}$$

Our mapper is trained to manipulate desired properties of the image indicated by the textual cue t, while preserving other visual properties of the input image. The CLIP loss $L_{CLIP}(w)$ guides the mapper to minimize the cosine distance in the CLIP latent space:

$$L_{CLIP}(w) = D_{CLIP}(G(w + M_t(w)), t) \tag{4}$$

where G again denotes the pre-trained StyleGAN generator. To preserve the visual properties of the original input image, we minimize the L_2 norm of the operation steps in the latent space. The similarity to the input image is controlled by the L_2 distance and the identity loss in the latent space:

$$L_{ID}(w) = 1 - < R(G(w_s)), R(G(w)) > \tag{5}$$

where R is a pretrained ArcFace network for face recognition, and $<\ ,\ >$ computes the cosine similarity between it's arguments. Finally, for edits that require identity preservation, we use the identity loss defined in Equation4. Our total loss function is a weighted combination of these losses:

$$L(w) = L_{CLIP}(w) + \lambda_{L2}||M_t(w)||_2 + \lambda_{ID} L_{ID}(w) \tag{6}$$

Experiments

In this paper, we generate cartoon faces by fine-tuning the StyleGAN2 model. We use Flickr-Faces-HQ (FFHQ) (Karras, Laine, and Aila 2019) as the source domain dataset to train stylegan2 model (Karras et al. 2020b) of 256 resolution, which has 70,000 high-quality images of human faces. For the target domain dataset, we conduct experiments on a variety of datasets, including Naver Webtoon (Back 2021), Metfaces (Karras et al. 2020a), and Disney (Pinkney and Adler 2020). In addition, we also choose several kinds of webtoons from Naver Webtoon datasets, which has 15 kinds of webtoons totally, to train in this experiment.

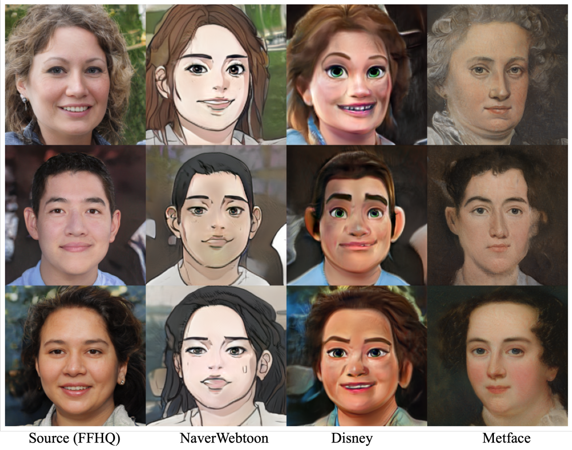

Figure 2: The results of the style transfer in Naver Webtoon, Disney, and Metfaces.

Style Tranfer

We first show the style conversion results in Naver Webtone, Disney, and Metface in Figure 1. As shown in Figure 1, the face image from the source domain data set is successfully converted into a cartoon face of the corresponding style. It can be seen that the image we generated can well reflect some features of the original image, including the arrangement and shape of the five facial features, the general hairstyle and face shape. At the same time, we can also see that some features are stylized among different styles. In Disney style, the size of eyes will be enlarged, and the proportion of eyes and face will be adjusted, so as to achieve more affinity for cartoon characters; In metface, the face shape of the portrait will be elongated as much as possible to achieve a very solemn visual effect. It is not difficult to find that our network has also learned these characteristics very well. However, we can observe that the generated cartoon face image still has some shortcomings. For example, the skin color of the generated Disney style face is so mottled that it looks a little scary.

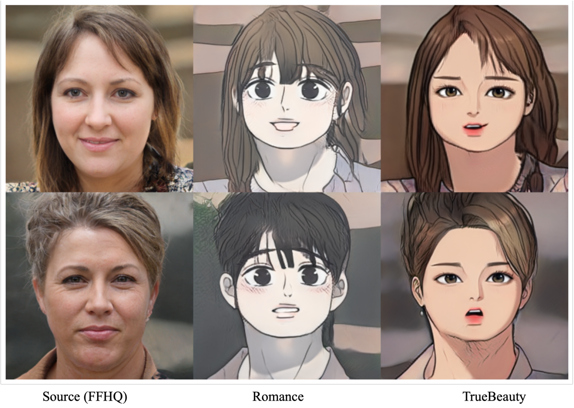

Figure 3: The results of the style transfer in Romance101 and TrueBeauty webtoon.

Figure 3 shows the result of style transformation in Romance101 and TrueBeauty webtoon. Although the corresponding style images have also been successfully generated, it can be seen that their defects are more obvious than the previously generated images, in which the facial structure and facial features have been destroyed, and the output images obtained from different input images are very uniform. Only a few facial features have slight differences, and only some low resolution features of facial features, such as hairstyle recognition, are better.

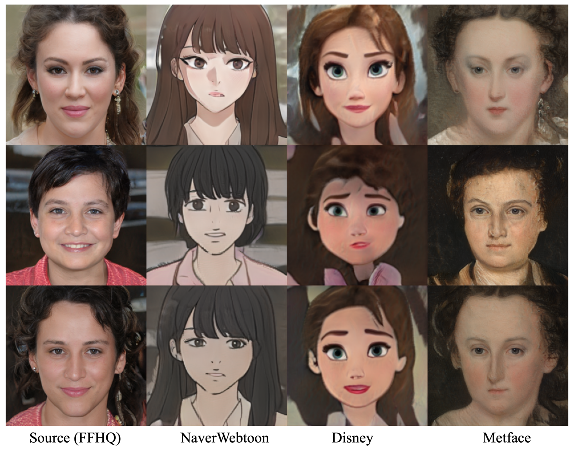

Figure 4: The results of the style mixing in Naver Webtoon, Disney, and Metfaces.

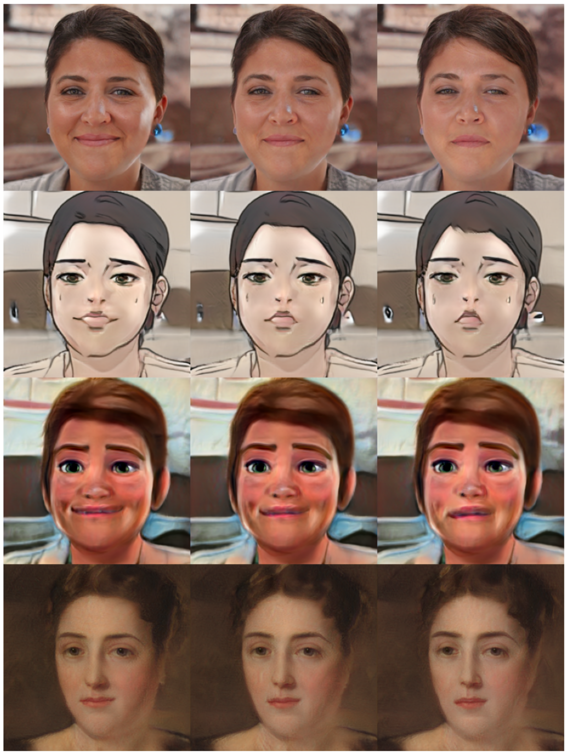

Figure 5: The results of the style clip using test ”a really angry face” in Naver Webtoon, Disney, and Metfaces.

Style Mixing

It is an interesting experiment to mix two images of the same style and see what will happen. As shown in Figure 4, we successfully mix the two images so that they have the features of both images. For example, in the Disney style, we can observe that the hair and face of the generated face are closer to the first image, while the features, such as the eyes and nose, look more like the second image.

Style Clip

We try to use text information to make the images generated in the direction we want them to go. Specifically, we generate an angry face and a sad face by text ”a really angry face” and ”a really sad face”, as shown in Figure 5 and Figure 6. In Figure 5, we can see that the expression of the face does change, although the final angry expression is not particularly obvious. In fact, we believe that a frown would be a better way to express anger, but the resulting expression lacks this feature. We suspect that this may have something to do with the dataset. On the other hand, in Figure 6, we can find that expression of the faces are more obvious, concentrating on the lowered corners of the mouth as well as the lowered eyebrows. And we think this phenomenon make sense with reason that these features would be more common than frowning, which allows the model to learn more about this.

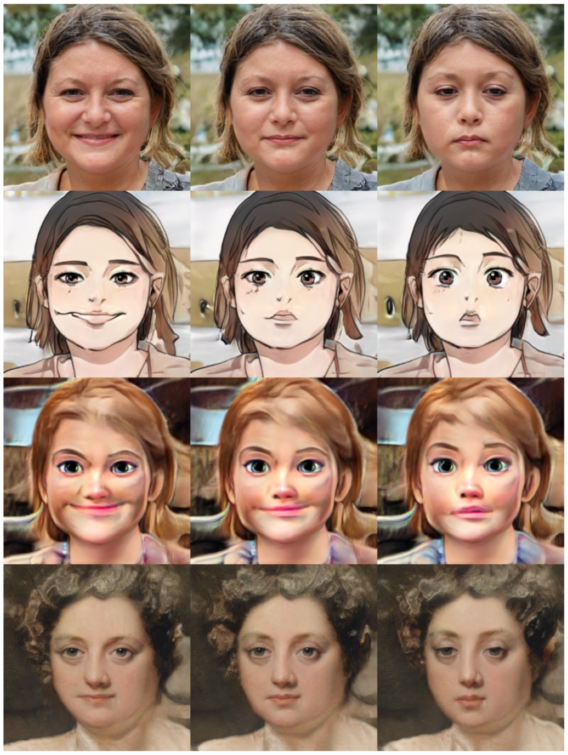

Figure 6: The results of the style clip using test ”a really sad face” in Naver Webtoon, Disney, and Metfaces.

Conclusion

In this paper, we mainly focus on the transfer of the portrait style based on samples. In the previous work, we need to spend a lot of energy and computing resources to retrain a GAN model every time we change a style. We refer to the work of Tony to improve this problem. The real StyleGAN we used to train the source domain dataset has fine tuned another StyleGAN on the new target domain styled dataset, which can greatly reduce the training cost of the model. In addition, we also tried some style mixing experiments to exchange pictures of different facial details of the same style. Finally, we introduce the CLIP model to generate face images with different expressions by parsing text information, and experiment in different styles of images. The experimental results show that the model can complete the given task and generate cartoon face images with corresponding styles, but the generated cartoon images still have some defects, such as the disharmony of facial features, and the slightly stiff expression, which will be left for future work.

References

Back, J. 2021. Fine-Tuning StyleGAN2 For Cartoon Face Generation. *CoRR*, abs/2106.12445.

Chen, H.; Hui, K.; Wang, S.; Tsao, L.; Shuai, H.; and Cheng, W. 2019. BeautyGlow: On-Demand Makeup Transfer Framework With Reversible Generative Network. In *IEEE Conference on Computer Vision and Pattern Recogni-*

tion, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, 10042–10050. Computer Vision Foundation / IEEE.

Choi, Y.; Uh, Y.; Yoo, J.; and Ha, J. 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 8185–8194. Computer Vision Foundation / IEEE.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. Texture Synthesis Using Convolutional Neural Networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 262–270.

Hertzmann, A. 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In Cunningham, S.; Bransford, W.; and Cohen, M. F., eds., *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998, Orlando, FL, USA, July 19-24, 1998*, 453–460. ACM.

Ho, J.; Chen, X.; Srinivas, A.; Duan, Y.; and Abbeel, P. 2019. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 2722–2730. PMLR.

Huang, X.; and Belongie, S. J. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 1510–1519. IEEE Computer Society.

Karras, T.; Aittala, M.; Hellsten, J.; Laine, S.; Lehtinen, J.; and Aila, T. 2020a. Training Generative Adversarial Networks with Limited Data. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Karras, T.; Laine, S.; and Aila, T. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 4401–4410. Computer Vision Foundation / IEEE.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020b. Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 8107–8116. Computer Vision Foundation / IEEE.

Li, B.; Zhu, Y.; Wang, Y.; Lin, C.; Ghanem, B.; and Shen, L. 2022. AniGAN: Style-Guided Generative Adversarial Networks for Unsupervised Anime Face Generation. *IEEE Trans. Multim.*, 24: 4077–4091.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M. 2017. Universal Style Transfer via Feature Transforms. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 386–396.

Ma, X.; Kong, X.; Zhang, S.; and Hovy, E. H. 2019. MaCow: Masked Convolutional Generative Flow. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 5891–5900.

Mo, S.; Cho, M.; and Shin, J. 2020. Freeze Discriminator: A Simple Baseline for Fine-tuning GANs. *CoRR*, abs/2002.10964.

Nizan, O.; and Tal, A. 2020. Breaking the Cycle - Colleagues Are All You Need. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 7857–7866. Computer Vision Foundation / IEEE.

Pinkney, J. N. M.; and Adler, D. 2020. Resolution Dependent GAN Interpolation for Controllable Image Synthesis Between Domains. *CoRR*, abs/2010.05334.

Prenger, R.; Valle, R.; and Catanzaro, B. 2019. Waveglow: A Flow-based Generative Network for Speech Synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, 3617–3621. IEEE.

Sheng, L.; Lin, Z.; Shao, J.; and Wang, X. 2018. Avatar-Net: Multi-Scale Zero-Shot Style Transfer by Feature Decoration. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 8242–8250. Computer Vision Foundation / IEEE Computer Society.

Zhu, J.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017. Toward Multimodal Image-to-Image Translation. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 465–476.