What if you meet a Disney princess in the real world? — Image style transfer from imagination to reality

1st Tzu Hsuan Su
DSPMT
National Tsing-Hua University
Hsin-Chu, Taiwan

2nd Yu Chun Lin
Department of Computer Science
National Tsing-Hua University
Hsin-Chu, Taiwan

3rd Wei Wei Du
DSPMT
National Tsing-Hua University
Hsin-Chu, Taiwan

4th Dai Rong Wu
Interdisciplinary Program of Engineering
National Tsing-Hua University
Hsin-Chu, Taiwan

5th Yueh Hsin Lu
Department of Computer Science
National Tsing-Hua University
Hsin-Chu, Taiwan

***Abstract*—What if you meet a Disney princess in the real world? In this project, we compare six different generative adversarial network(GAN) [1] models to implement image style transfer, especially transform from cartoon to reality. In addition to our results, this paper also contains all details of the training process, and the figures for each model when training. Finally, we conclude that CycleGAN and Pix2pix are the model which has the outstanding performance among all models.**

***Index Terms*—Style transfer, GAN, Image-to-image translation**

I. Introduction

Can the wonderland in Disney occur in reality? Imagine the Disney princesses/princes become your neighbors or even wife/husband. What would they look like? Using style transfer will give you a picture.

Style transfer is an optimization technique used to take two images—a content image and a style reference image (such as an artwork by a famous painter)—and blend them together, so the output image looks like the content image, but "painted" in the style of the style reference image.

As a result, we plan not only to use a style transfer algorithm to achieve the goal, but also to compare different model performance.

We choose neural style transfer to be the base method compared to GAN. Since neural style transfer is a fast and uncomplicated way to try, implement our goal—make Disney characters real-looking. We take the portrait photography as the style and Disney character as content.

Apply six different generative adversarial networks(GAN) to implement the transformation. The architecture of GAN includes two parts: The generator learns to generate plausible images and become negative training examples for the discriminator. And the discriminator learns to distinguish the generator's fake image from the real images. The discriminator penalizes the generator for producing implausible results. The key of GAN is the idea of adversarial loss. It's powerful for image generation tasks and aims to optimize the performance.

II. Literature Review

Regarding image style transfer, there are five main methodologies [3], including stroke-based rendering, image analog, image filtering, texture synthesis and neural style transfer. The common limitation of former four methods, except for neural style transfer, are that they do not have enough flexibility, style diversity, and effective image structure extraction; in other words, they only focus on the style and ignore the importance of content. Therefore, we take a deeper look at neural style transfer.

A. Neural Style Transfer

The main idea of neural style transfer is combining visual texture modeling and image reconstruction. Based on different permutations and combinations, there are several derivations of neural style transfer. (1) Parametric neural method with summary statistics (2) Non-parametric neural method with MRFs [4] (3) Per-style-per-model neural method (4) Multiple-style-per-model neural method (5) Arbitrary-style-per-model neural method [5]

In my opinion, method selection will highly depend on the evaluation criteria. For instance, compared method (1) and (2), non-parametric neural methods with MRFs may get better performance for photorealistic style, which is our objective in this project, due to the patch-based MRF loss. However, the biggest drawbacks are the low speed and high computational cost.

Therefore, another category, method (3), (4), and (5), use a pre-trained feed-forward network to optimize the model. The differences depend on the number of artistic styles a single generator can produce. I believe method (5), arbitrary-style-per-model neural methods, will have better performance in our

project. Because characters in Disney have slight differences, such as Elsa and Cinderella, the model can use one single trainable model to transfer arbitrary artistic styles.

B. Real-world applications

a) Prisma: Prisma Labs is a mobile technology company specializing in deep learning-related products. They created an app called Prisma [6] not only changing the color or style but also repainting the image by neural style transfer algorithm. However, this application is not open source.

b) Ostagram: Ostagram is an application created by a Russian team. They used the deep neural network algorithm proposed by Gatys [7] to generate images. Figure 1 are the input content image, input texture image, and the final output.

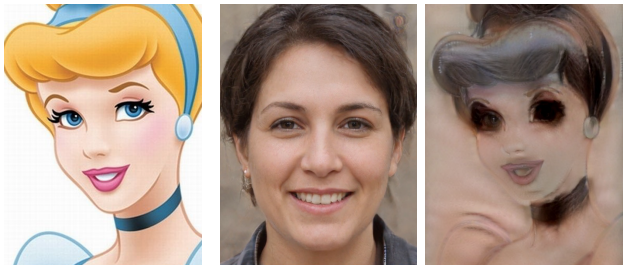

Fig. 1: Result by using Ostagram

III. Methods

A. Preprocessing

In the training phase, we use FFHQ [8] dataset as the real image and Disney dataset as the cartoon image. In the testing phase, we use Disney character images different from the training set.

1) CycleGAN , CartoonGAN:

a) Data augmentation: Because the number of images that Disney dataset [9] provided by Nvidia is only 317, we use the data augmentation techniques to significantly increase the diversity of data available for training models, getting 2000 Disney images.

b) Resize and Crop: Because the input sizes of the images are different, we process a different extent of resizing and cropping in each model.

2) StyleGAN, AnimeGAN, CUT, Pix2Pix:

a) Dataset: In the training phase, we use FFHQ dataset as the real image and the output image generated by Toonify as the cartoon image. In the testing phase, we use the original Disney character images.

b) Resize and Crop: Because the input sizes of the images are different, we process a different extent of resizing and cropping in each model.

c) Generate images: To get more cartoon images with the similar style, we use "Toonify Yourself" [10], a pretrained model that can turn realistic images into cartoon style. It took us two days to generate 373 images.

B. CycleGAN

CycleGAN [11] is an unpaired image-to-image translation method. It was published by Berkeley AI Research(BAIR) laboratory in August 2020. The model can capture the special characteristics of one image collection and figure out how these characteristics could be translated into the other image collection.

The model includes two mappings and two adversarial discriminators. The discriminator aims to distinguish between images and translated images. The functions are cycle-consistent: forward cycle consistency: x $\rightarrow$ G(x) $\rightarrow$ F (G(x)) $\approx$ x and backward cycle consistency: y $\rightarrow$ F (y) $\rightarrow$ G(F (y)) $\approx$ y

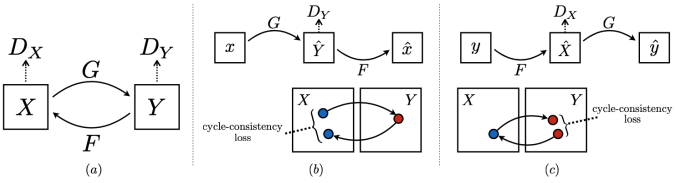

Fig. 2: CycleGAN architecture.

C. CartoonGAN

CartoonGAN [12] is a model designed for transforming photos of real-world scenes into cartoon style images, on the basis of the author's assumption, the cartoon images have two important characteristics. The first point is high level simplification and abstraction, and the other point is having clear edges, smooth color shading and relatively simple textures. CartoonGAN uses CNN as their generator and discriminator, and the structure is as below.

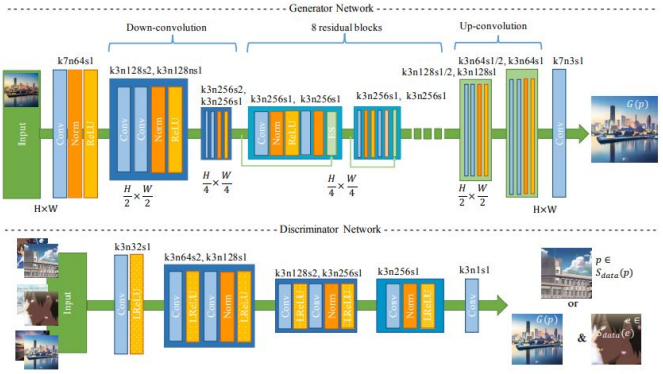

Fig. 3: CartoonGAN architecture.

The special part of this model is in the loss function, and it is composed of content loss and the adversarial loss. Content loss means the difference between input image and the generated image, and it uses the VGG16 as a feature extractor to compute the loss. This design does not change the outline of the images too much, but just change the texture of the images. As for the adversarial loss, we use the BCE function to compute the loss of target style images, target style smoothed images separately. Besides, the additional smoothed image computed by the loss

function gives output a smooth color shading. We picked this model in the beginning because we thought this model can have an inverse transformation, but the paper and the result both show it is not as we thought before.

D. Toonify (StyleGAN)

Stylegan is a GAN, first proposed by NVIDIA, with a different generator architecture. While a traditional generator feeds latent code, NVIDIA presented a new architecture called style-based generator. The new architecture is looked like Figure 2 [13]. Stylegan takes the style that generators get from two different images, and uses the style mixing skills (they say in the paper) to decide different weights from two images, to generate a new image, such as non-existent person faces.

Toonify [14] is a network that based on Stylegan, which is used to transfer real person faces into cartoon-like faces. In Toonify, we use two Stylegan models. One is NVIDIA pretrained model, and the other one is the transfer learning model leaning on some cartoon datasets. Toonify combines these two models. In this method, we modify the weight between two Stylegan models, we did not do any other training task, because of the limitation of the computing resources. The original usage is to change a real person face into cartoon-like face.

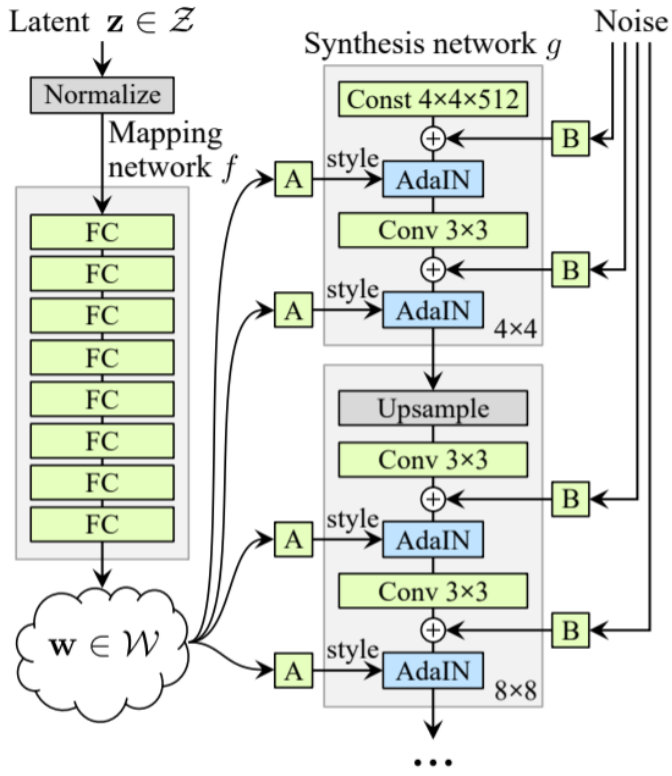

Fig. 4: Style-based generator architecture.

E. AnimeGAN

AnimeGAN [15] combines neural style transfer and generative adversarial networks for transforming photos of real-world scenes into anime style images. There are some problems in the existing model for anime style transfer: 1) The performance of the content preserved of the original image and the anime style is not well in the output image; 2) the model size is too big and inefficient. The loss functions in this model are grayscale style loss, color reconstruction loss and grayscale adversarial loss. The former two are used in the generator for making the generated image have a more obvious anime style and preserve the color of the photos. The later one in the discriminator makes the output image have vivid color and clear edges. The architecture of the model is below:

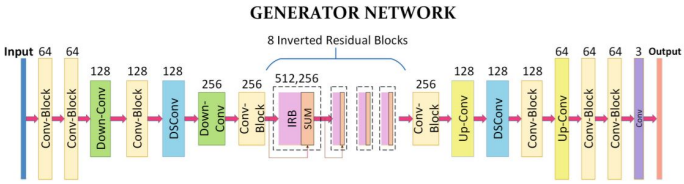

Fig. 5: AnimeGAN generator

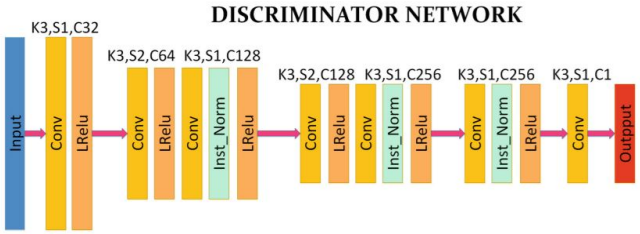

Fig. 6: AnimeGAN discriminator

F. Contrastive Unpaired Translation

Contrastive Unpaired Translation(CUT) [16] is a new unpaired image-to-image translation model, the biggest improvement is that the model cost less training time while preserve the outstanding performance. CUT-model utilizes the concept of contrastive learning, through the feature extractor to get the positive and negative sample from the image. Using the image below as an example, the horse head will be the positive sample and the horse back and grass are negative samples. The goal is to make the positive sample become more likely to output. In this way the model can learn by higher level features, and not rely on cycle consistency loss to compare the relation of results and inputs.

G. Pix2Pix

Pix2pix [17] is a GAN model based on cGAN, while the biggest different is that we use u-net structure generator, and PatchGan discriminator. In this model, we need paired training data, the data has one real person image (from FFHQ dataset), and the other one is cartoon-like face (we generate by using Toonify).

H. Comparison

Following are two tables comparing the similarity and difference of the six models. Except for Pix2pix, other models use the unpaired input, which are easier to collect the data. As for the learning time, CycleGAN spends the most time. As for model size, AnimeGAN has the biggest model size.

TABLE I: Model comparation

Modelname	***CycleGAN***	***CartoonGAN***	***Toonify (StyleGAN)***
Pretrained	VGG16	VGG16	StyleGAN
Input	unpaired	unpaired	unpaired
Dataset	2000*256*256 1000*256*256	2000*256*256 2000*256*256	70000*1024*1024 (StyleGAN)
Learning time	35 hr	6.5 hr	6 d 14 hr
Model size	113 MB	140 MB	not mentioned
Learning epoch	200	103	not mentioned

Dataset:Disney/FFHQ: CycleGAN, CartoonGAN; toonified/FFHQ: StyleGAN

TABLE II: Model comparation

Modelname	***AnimeGAN***	***CUT***	***Pix2Pix***
Pretrained model used	VGG16	none	none
Input	unpaired	unpaired	paired
Dataset	2000*256*256 2000*256*256	373*256*256 373*256*256	373*256*256 373*256*256
Total learning time	6.5 hr	10 hr	2.5 hr
Model size	446MB	45MB	207MB
Learning epoch	10	500	200

Dataset: Disney/FFHQ: AnimeGAN; toonified/FFHQ: CUT, Pix2Pix

IV. Results

A. CycleGAN

The model is good at dealing with the character that it has learned in the training set. The more similar the input angles and poses are, the better output performances are. For example, Snow White has appeared in the training set and Tiana has not, so Snow White has better performance.

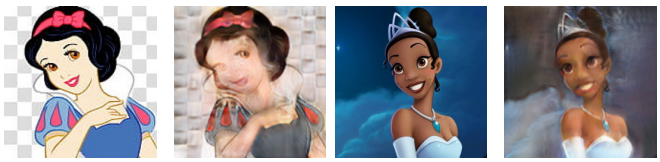

Fig. 7: Output of Snow White and Tiana by CycleGAN

Besides, as the number of epochs increases, the model learns more features and the output will become more realistic. Example is as follows:

Also, we find some interesting things. For example, the model has learned the color of human skin, so it can transform Shrek green skin into the color of human. Moreover, among all facial features, eyes have the authentic and suitable performance. The eyes of Disney characters are too big and not in the normal proportion in average, and our model can transform into more like humen.

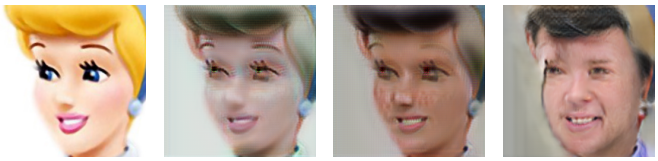

Fig. 8: Original image, output after 35 epochs, 100 epochs and 200 epochs by CycleGAN

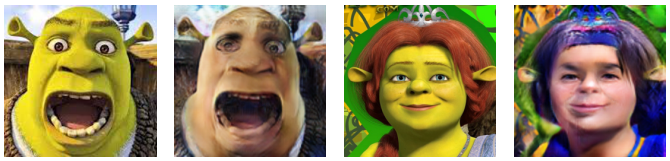

Fig. 9: Output of Shrek by CycleGAN

B. CartoonGAN

In the original paper, the author has used the techniques of cropping and smoothing in preprocessing process. However, we consider that it will affect the model to detect the detail of the target style images, so we did not crop the image and only resize it. In addition, we found that the adversarial loss is quite low compared to the content loss, so we adjust the weight of content loss and adversarial loss. After training for 103 epochs, the improvement is quite subtle that the only difference between input and output is the image became a little darker than before.

C. Toonify (StyleGAN)

It performs very well when we try to transfer the real person faces into cartoon style. However, it does not perform very well in the opposite side.

D. AnimeGAN

After many epochs of training, the main change in our output is that the model can distinguish the background and the person profile in order to smooth the edges and change the color histogram of specific objects. The structure of the object in the image does not have an obvious change. The output image still looks like a cartoon because the only change is the color distribution.

E. Contrastive Unpaired Translation

After training for 500 epochs, the biggest change when using a 3D image as input is that the texture will change a little. As for using the 2D anime image as input, color changes a little, and the model will draw eyes on the picture. To sum up, results have no change on the outline of the input picture but a little transformation on the texture.

F. Pix2Pix

The performance of the model is good when the input image is similar to Disney animation, which means the style is similar to Toonify image. The structure of the object and some

parts of human face like eyes or nose are transformed from cartoon style to real successfully. The output images of Belle and Hiccup are shown below. The colors are also transformed obviously.

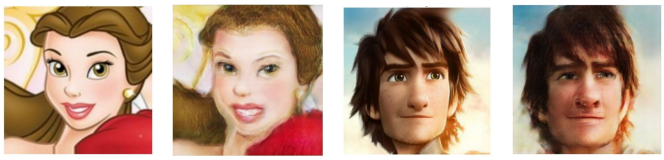

Fig. 10: Output of Belle and Hiccup by Pix2Pix

V. Discussion

A. CycleGAN

We spent about 35 hours on training by using NVIDIA Geforce RTX 2070 super. It is undoubtedly time-consuming. We tried to modify some parameters, such as learning rate, but the effect is not significant. The original model was designed for two equivalent features, like apple to orange, horse to zebra and summer to winter. However, the difference between cartoon and reality is too big, as a result, it is hard for the model to learn the feature. In the future, we suppose that if we add more training set and train for longer periods, the performance will increase.

B. CartoonGAN

We found that this is a specific model for turning realistic images into cartoon images, and this model is not bidirectional, so the result is not that meets our expectations. Every image put in this model will only have a smoother result, and because our input has already used the cartoon image, so the texture did not change a lot, only the overall color turned darker. We also tried to adjust the weight of the model, but it turned out that there was more blocky color on the images, so this model is totally not suitable for our goal.

C. Toonify (StyleGAN)

The model is hard for us to train in our condition, and also if we need to use this model, we need to modify it a lot. Therefore, we only adjust the parameter between two Stylegan models. The result doesn't look good. Most of the picture larger the eyes, or just mess up the face. It might due to the model original usage. It is used to transfer a real person picture into cartoon-like face. Although, we try to modify the weight, but it doesn't seem to do a lot of work.

D. AnimeGAN

Since AnimeGAN is highly related to CartoonGAN, they share many concepts and ideas on how to transform real images into cartoons/animation. The result of the inverse usage, which means reverse the input data for training, is not as good as expected. We try to preprocess the image more specific for Disney style. However, the main obstacle in this model originated from its idea for edge preserving learning, which is focused on how to make the output look like anime.

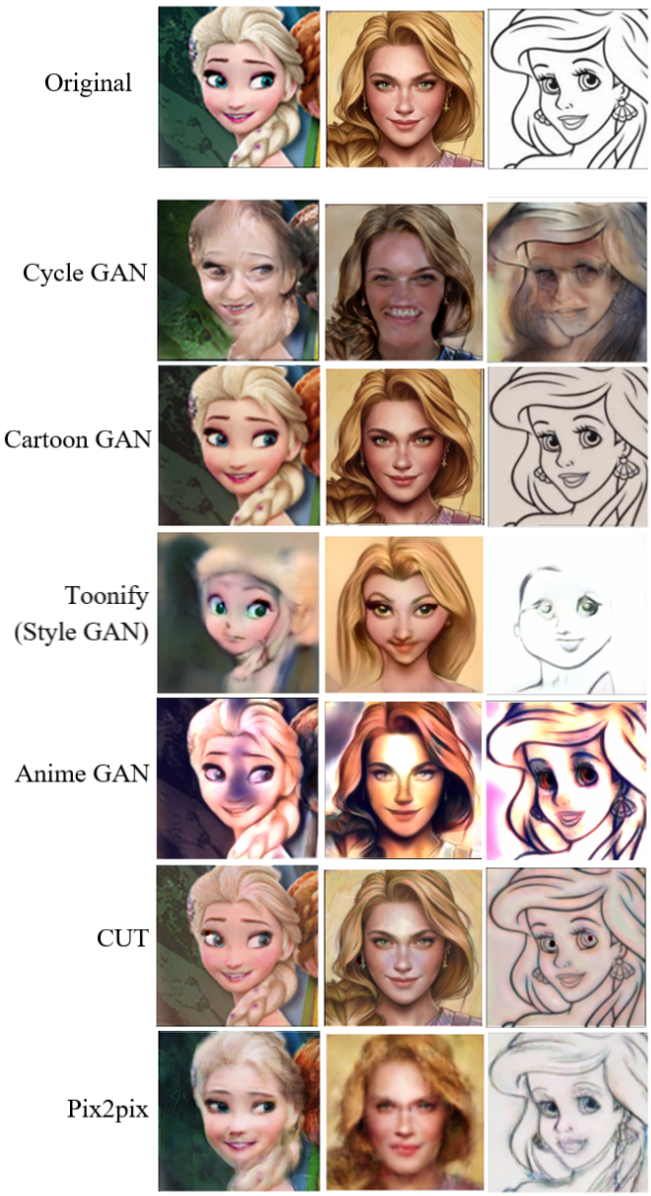

Fig. 11: Models comparison
We pick three different art style images as the criterion to compare different model performances. (Elsa: 3D animation style, Rapunzel: realistic style, Little mermaid: 2D black-and-white sketch style.)

E. Contrastive Unpaired Translation

Although this model announces it has a better training time and will have a similar effect, we think that there must be some tradeoff between the training time and result. 500 epochs can be finished in 12 hours is fast, while the result is obvious not good enough. In the original paper, we found that their example took a lot of epochs for training. For instance, the Cat-to-Dog contains 5,000 training and 500 val images from AFHQ Dataset. So even though the result trained by the CUT model is not satisfying, maybe the result will be as well as CycleGAN does if we train enough epoch times in our cases. At least this model support a bidirectional transformation compared to CartoonGAN.

F. Pix2Pix

Most of the output images are a little ambiguous. However, in some images (style close to real person or Disney animation), we do see the model try to apply some real-person-like feature on to the input picture (face shape and texture). In this training, we only use 373 pairs of training data and run 200 epochs. Maybe the result will become better if we increase the number of training data.

VI. Conclusion

After weeks of hard work, we have concluded several points to sum up our final project. In the field of preparing datasets, in order to get more cartoon images with the similar style, we spent two days to generate 373 images by pretrained model that can turn realistic images into cartoon style.

Besides, we found that the model performance will be highly related to the training dataset. To be more specific, if we use toonified images to train our model while using the 2D Disney character to test the result, the outcome will be pitiful.

On the other hand, the design of the model influenced the result a lot. Some models can be used to transform to many kinds of styles, but some are specific to transform to a certain style. For instance, AnimeGAN and CartoonGAN use the strategy of smoothing the images to emphasize the smoothed style.

In conclusion, CycleGAN and Pix2Pix are the best models in cartoon-to-real translation. Image-to-image translation takes lots of training time and computing power, and tuning parameters are also time-consuming. Depending on the model original design motivation and architecture, every model is suitable for different usage and has completely different pros and cons. If you are not satisfied with the outlook of the current princesses, ensemble (combining different models) may be a possible solution.

VII. Author Contribution Statements

- **T.H.S. (20%): CartoonGAN, CUT, Presentation, Generate toonified images**
- **Y.C.L. (20%): AnimeGAN, Pix2Pix, died UGATIT**
- **W.W.D.(20%): CycleGAN, Proposal Presentation**
- **D.R.W. (20%): StyleGAN, Pix2Pix**
- **Y.H.L.(20%): Neural Style Transfer**
- **Paper survey, final report: co-editing**

References

[1] GOODFELLOW, Ian, et al. Generative adversarial networks. Communications of the ACM, 2020, 63.11: 139-144.

[2] Jing, Yongcheng, et al. Neural style transfer: A review. IEEE transactions on visualization and computer graphics. 2019. p. 3365-3385.

[3] Jing, Yongcheng, et al. Neural style transfer: A review. IEEE transactions on visualization and computer graphics. 2019. p. 3365-3385.

[4] Li, Chuan, and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[5] Huang, Xun, and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. Proceedings of the IEEE International Conference on Computer Vision. 2017.

[6] https://prisma-ai.com

[7] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.

[8] FFHQdataset, https://github.com/NVlabs/ffhq-dataset.

[9] Disneydataset, https://github.com/justinpinkney/toonify.

[10] Tonify Yourself, https://toonify.photos/original.

[11] ZHU, Jun-Yan, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. 2017. p. 2223-2232.

[12] CHEN, Yang; LAI, Yu-Kun; LIU, Yong-Jin. Cartoongan: Generative adversarial networks for photo cartoonization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 9465-9474.

[13] KARRAS, Tero; LAINE, Samuli; AILA, Timo. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2019. p. 4401-4410.

[14] PINKNEY, Justin NM; ADLER, Doron. Resolution Dependant GAN Interpolation for Controllable Image Synthesis Between Domains. arXiv preprint arXiv:2010.05334, 2020.

[15] CHEN, Jie; LIU, Gang; CHEN, Xin. AnimeGAN: A Novel Lightweight GAN for Photo Animation. In: International Symposium on Intelligence Computation and Applications. Springer, Singapore, 2019. p. 242-256.

[16] PARK, Taesung, et al. Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision. Springer, Cham, 2020. p. 319-345.

[17] ISOLA, Phillip, et al. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 1125-1134.