Master Thesis

Animation Line Art Colorization based on Machine Learning

Graduate School of Advanced Science and Engineering, Hiroshima University

LI JIAYIN

September 2023

Master Thesis

Animation Line Art Colorization based on Machine Learning

Adviser Prof. Yasuaki Ito

Informatics and Data Science Program, Graduate School of Advanced Science and Engineering
Hiroshima University

M216065 LI JIAYIN

Thesis submission: August 2023

Abstract

Automatic colorization of animation line art is a very practical task. Whether you are an amateur or a professional, it takes a lot of time and effort to color the lines. At the same time, for amateurs, drawing and coloring a richly detailed line art can be extremely challenging.

Therefore, automatic colorization of animation line art using machine learning methods has become a hot topic in the field of computer vision.

We have come up with a machine learning-based method that can quickly colorize line arts and add more details to less detailed line arts.

This method utilizes three Pix2Pix networks and one SCFT network. The three Pix2Pix networks are the line addition network, the coloring network and the line addition dataset production network. The line addition network is used to generate artificial line arts with more details, the coloring network is used to generate automatic coloring results, and the line addition dataset production network is used for making the training dataset of line addition network. The SCFT network is used to provide coloring results based on the reference image, making the coloring results more personalized.

Experimental results show that this coloring method has good results and has a certain practicality.

Contents

List of Figures

Chapter 1

Introduction

1.1 Research Background

Anime is a popular art form, with strong ornamental and cultural communication value. The expression form of animation design refers to the use of relatively realistic graphics and the reproduction of prototypes by exaggeration and refinement, which is a creative method with distinctive characteristics of prototypes [1]. To create animation, designers need to have solid art skills and be able to extract characteristic elements from natural prototypes and represent them in an artistic way. In the process of animation creation, the coloring step often takes a great deal of time and effort [2].

Therefore, automatic coloring of animation line arts using computer technology is a very important topic.

The traditional method is to color by component. That is, floodfill [3] is applied to the image. The most common application is the "paint bucket" function in the released drawing applications. The principle is to start from one point and fill all the nearby pixels with the specified color until all the pixels in the closed area are filled with that color. However, the problem of this method is that the color of the closed area must be specified manually, and only the specified color can be selected to fill the area. If the specified color area is not closed, the problem of color overflow will occur, so it is not intelligent and efficient. In addition, there are some methods [4] [5] to use the establishment of a large color database, and then search and match the sketch with similar image slices in the database to complete the color filling. This method requires a large amount of image data and does not have better scalability which cannot color an image that does not exist in the database.

In recent years, As time goes, many image generation methods based on deep neural networks have been proposed. Such as deep boltzmann machine (DBM) [6], convolutional deep belief networks [7], etc. Machine learning-based coloring methods have come into being. Compared with traditional algorithms, they are more efficient and intelligent. Generative adversarial network (GAN) [8] is proposed to make the task of image to image translation easier. Although images generated using GAN are often noisy or blurred compared to real images [9], thanks to it' s subtle design, with a large amount of data,

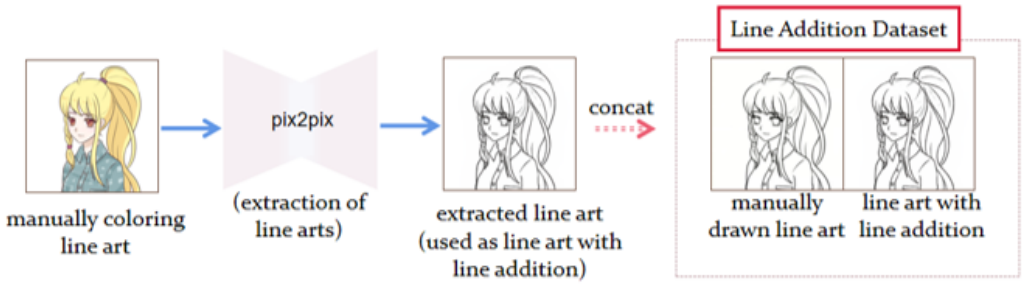
the network can find the optimal parameters of the generator and discriminator through iterative training, so that the loss function has a very small maximal value, thus realizing the image-to-image translation. Based on GAN, conditional generative adversarial nets (cGAN) [10] was proposed. GAN is unsupervised training, cGAN adds conditional information (category labels or other modal information) to the input and becomes a GAN using supervised training. Compared to GAN, cGAN can better control the generate content. Based on cGAN, the Pix2Pix network was proposed [11]. This network is widely used in image-to-image translation as well. Pix2Pix is essentially a cGAN. The image is fed into the generator and discriminator as a condition of this cGAN. It uses the U-Net construct for the generator part and PatchGAN for the discriminator [12]. By patch, we mean that no matter how large the generated image is, it will be cut into multiple fixed-size patches and fed into the discriminator for judgment. The advantages of this design are: the input to the discriminator becomes smaller, the computation is also smaller and the training is faster. The Pix2Pix network can be directly applied to the coloring of animation line art, which is less time consuming and automatic than the traditional methods mentioned above. However, coloring with Pix2Pix directly still suffers from low completion and color confusion. Therefore, a certain degree of processing and grouping of images before coloring with Pix2Pix will give better results. In addition, it is also a good coloring method to provide a reference image that is semantically similar to the animation line art to be colored, and then map it to complete the conversion of the line art to the reference image. Current methods for coloring on this principle include AdaIN [13] and SCFT [14], among others. In this work, we utilize two models, Pix2Pix and SCFT, to achieve more diverse coloring results. Among them, self-attetion [15] is utilized and improved in the SCFT method.

1.2 Research Purpose

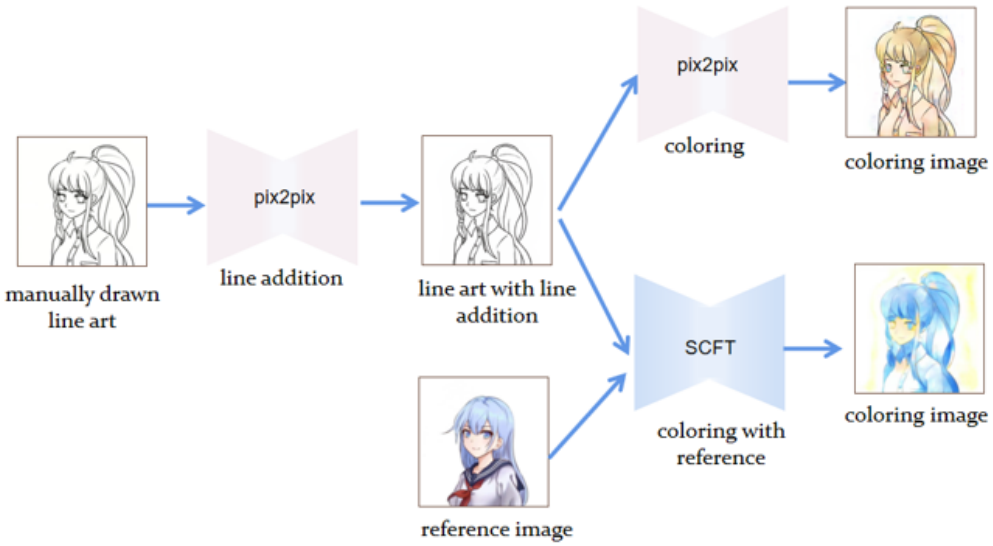
This paper proposes a method to process line art using Pix2ix network, and the processed dataset can be subsequently colored with higher accuracy. Meanwhile, this paper utilizes two models, Pix2Pix and SCFT, to color the processed line art to provide more choice space. When using Pix2Pix, the coloring results can be generated automatically without providing reference images. When using SCFT, the coloring can be done according to the provided reference image, and the coloring result is similar to the reference image. It should be noted that when training the Pix2Pix network, paired data need to be provided. We divide this task into multiple steps to accomplish it. Figure 1.1 illustrates our proposed line art coloring method.

The method consists of three Pix2Pix networks and one SCFT network. The first Pix2Pix network is the line addition network, which allows us to realize the function of adding details to a simple line art by training on a specially designed dataset. We design this network for this reason, for example, we want to colorize a line art of an amateur, but due to the amateur's low level of drawing, his line art has many less finished parts. At this point, we complicate it. When the lines are complicated, the details will increase, and then we will color the detail-rich line arts to get the final colored image. The second pix2pix

network is the coloring network, which is used to color the line art that has added details. While the first two networks will be directly involved in the image creation process, the third network is used to create the dataset for the line addition network and does not participate in the actual colorization process. Each data of the dataset is the pair of the line art without added details and the line art with added details. The SCFT network is used in the case where one wants to formulate the coloring result. This network takes a processed line art and a reference image as inputs, and can get a similar coloring result as the reference image. After all the networks are trained, details can be added to an input line art and then colored, the SCFT network is used for coloring when the user wants to provide a reference image, and the coloring Pix2Pix network is used for coloring when there is no reference image.



(a) Generating line addition dataset



(b) Generating coloring image

Figure 1.1: The outline of the proposed method

1.3 Chapter Structure

This thesis is organized as follows:In the first chapter, the background of the study and the purpose of the study are introduced. The second chapter introduces related research such as Pix2Pix, SCFT and other related concepts. In Chapter 3, we describe the line art coloring method in detail. In Chapter 4, we present the experimental parameters, experimental results and comparative experiments. At the end, in Chapter 5, we give a summary of this paper and present the goals of future research.

Chapter 2

Related Work

2.1 Pix2Pix network

Pix2Pix is a type of Generative Adversarial Network (GAN), which is mainly used in the field of image translation.

Pix2Pix is improved from cGAN. The core of cGAN lies in the fact that it adds additional conditional information to the inputs of the generator and the discriminator, and the pictures generated by the generator not only need to reach a certain degree of realism, but also have to satisfy the specified conditions in order to be recognized by the discriminator. Pix2Pix can be regarded as a cGAN whose conditions are pictures, but it also makes many improvements on the basis of cGAN, making it more suitable for accomplishing image translation tasks which need to preserve the structure of the input image (such as the line art coloring).

2.1.1 The structure of Pix2Pix

The structure of Pix2Pix is shown in Figure 2.1. It consists of a generator and a discriminator. The purpose of the generator is to convert the input image in the X-domain into the fake image $G(x)$ in the Y-domain. $G(x)$ has to fool the discriminator as much as possible. On the other hand, the purpose of the discriminator is to determine, as far as possible, that the image $G(x)$ generated by the generator is fake. After training, $G(x)$ will not be distinguishable by the adversarially trained discriminator. At this time, $G(x)$ will be very close to Y.

2.1.2 The objective function of Pix2Pix

The objective function of Pix2Pix consists of two parts, the objective function of cGAN and the L1 loss function. The objective function of cGAN is as follows:

$$\mathcal{L}_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_x[\log(1 - D(x, G(x))] \tag{2.1}$$

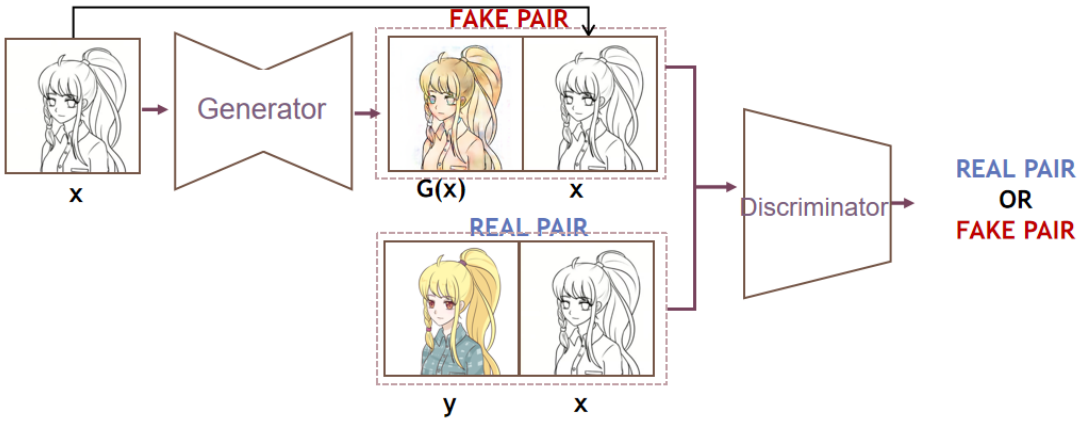


Figure 2.1: The structure of Pix2Pix

The L1 loss function is as follows:

$$\mathcal{L}_{L1}(G) = E_{x,y}\left[\|y - G(x)\|_1\right] \tag{2.2}$$

In Pix2Pix, L1 loss and cGAN loss are used in combination because L1 loss recovers the low-frequency part of the image(color blocks) while cGAN loss recovers the high-frequency part of the image(edges, etc.). Thus, the final objective function of Pix2pix is:

$$\mathcal{L}_{Pix2Pix} = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{2.3}$$

The generator tries to minimize the objective function, while the discriminator tries to maximize it. Where λ is a hyperparameter that can be adjusted according to the situation, when λ equals to 0, it means that the L1 loss function is not used.

2.1.3 The structure of the generator

The generator uses the U-Net [16] structure. Since in many image to image translation problems the input image and the generated image basically keep the same structure, for example, in the line art coloring problem, the input image and the output image share the same line, so in order to keep the original structure of the image, the generator adds a skip connection to share information between the input and the output which is shown in Figure 2.2.

2.1.4 The structure of the discriminator

The discriminator uses PatchGAN [12]. Unlike ordinary discriminator, PatchGAN is fully convolutional, after the image passes through each convolutional layer, it is not input to

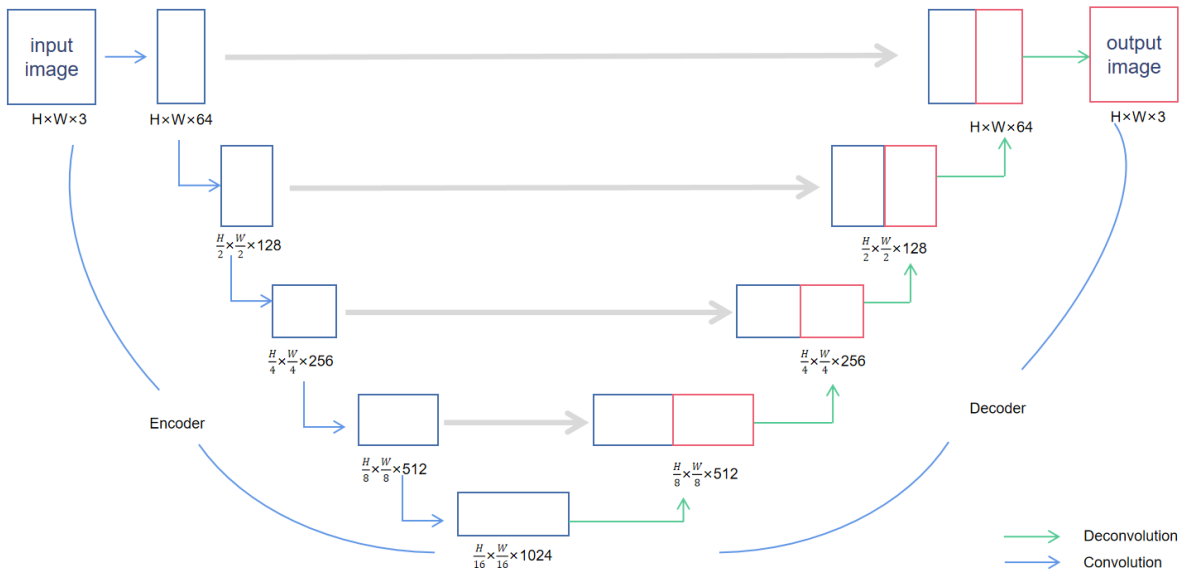


Figure 2.2: The structure of the generator

the fully connected layer or activation function, instead, it uses the convolution to map the input to an $N \times N$ matrix, which is equivalent to the final evaluated value in the original GAN, and it is used to evaluate the generated image of the generator. The advantages of this design are: the input to the discriminator becomes smaller, the computation becomes smaller as well and the training is faster. The structure is shown in Figure 2.3.

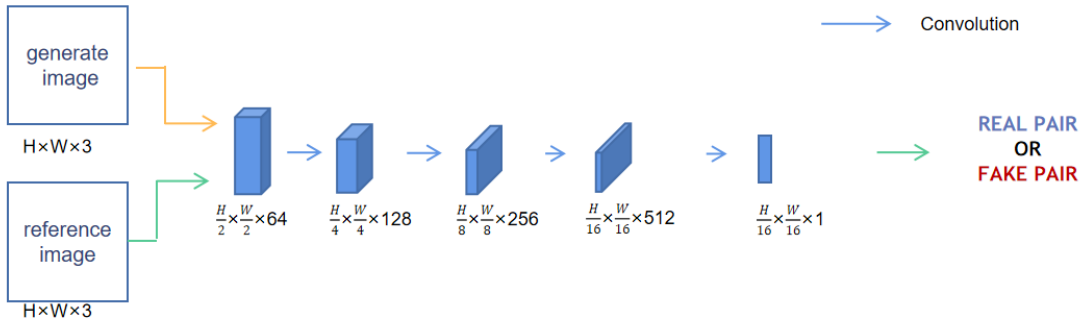


Figure 2.3: The structure of the discriminator

2.2 SCFT network

SCFT is also a kind of Generative Adversarial Network(GAN). It is mainly used in the field of image translation with reference images.

The most important part of SCFT is the spatially corresponding feature self-attention module, which is designed basing on the self-attention mechanism.

2.2.1 The structure of SCFT

As shown in Figure 2.4, the model mainly consists of a generator(containing two encoders and one decoder) and a discriminator. Firstly given an input coloring image I, it will be extracted by the outline extractor to produce line art I_s. On the other hand, thin plate splines transformation(TPS) is used to obtain an augmented-self reference image I_r. The corresponding activation maps f_s and f_r are obtained by taking I_s and I_r as the inputs of encoder E_s and encoder E_r, respectively, which are then used as the inputs of the SCFT module. From this, the coloring image I_c is finally obtained by the decoder.

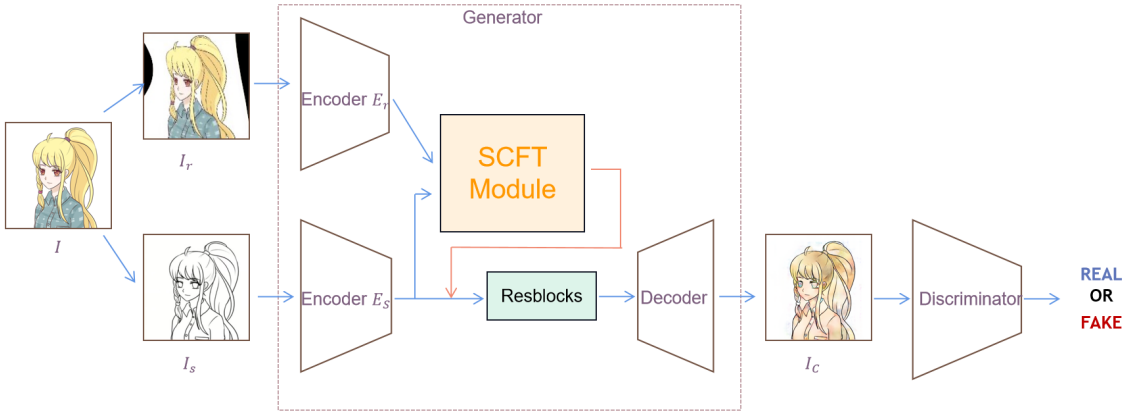


Figure 2.4: The structure of SCFT

2.2.2 The objective function of SCFT

The objective function of SCFT consists of five parts, which are the similarity-based triplet loss, the L1 loss, the cGAN loss, the perceptual loss and the style loss.

The similarity-based triplet loss is based on triplet loss [17]. This loss is used to supervise the pixel-wise query vector to be used in the SCFT module (will be introduced in the next section in detail) and the key vector affinity as follows:

$$\mathcal{L}_{tr} = \max\left(0, \left[-S\left(v_q, v_k^p\right) + S\left(v_q, v_k^n\right) + \gamma\right]\right) \tag{2.4}$$

Where v_k^p is the sample of positive region and v_n^p is the sample of negative region, γ denotes the minimum distance between $S\left(v_q, v_k^p\right)$ and $S\left(v_q, v_k^n\right)$.

L1 Loss has already described in the Pix2Pix section, where I_{gt} is the ground truth image extracted from I_r.

$$\mathcal{L}_{rec} = E\left[\left\|G\left(I_s, I_r\right) - I_{gt}\right\|_1\right] \tag{2.5}$$

The cGAN loss has also been described in section 2.1.2 as well.

$$\mathcal{L}_{cGAN} = E_{I_{gt},I_s}\left[\log D\left(I_{gt}, I_s\right)\right] + E_{I_s,I_r}\left[\log\left(1 - D\left(G\left(I_s, I_r\right), I_s\right)\right)\right] \tag{2.6}$$

The perceptual loss is used to reduce semantic gap, where $\hat{I}$ represents the generated graph and ϕ_l represents the activation graph of layer l.

$$\mathcal{L}_{\text{perc}} = E\left[\sum_l \left\|\phi_l(\hat{I}) - \phi_l\left(I_{gt}\right)\right\|_{1,1}\right] \tag{2.7}$$

In style loss $\mathcal{G}$ is a gram matrix. denoted as follows:

$$\mathcal{L}_{\text{style}} = E\left[\left\|\mathcal{G}\left(\phi_l(\hat{I})\right) - \mathcal{G}\left(\phi_l\left(I_{gt}\right)\right)\right\|_{1,1}\right] \tag{2.8}$$

The final objective function is given below:

$$\mathcal{L}_{\text{total}} = \lambda_{tr}\mathcal{L}_{tr} + \lambda_{\text{rec}}\,\mathcal{L}_{rec} + \lambda_{cGAN}\mathcal{L}_{cGAN} + \lambda_{\text{perc}}\,\mathcal{L}_{\text{perc}} + \lambda_{\text{style}}\,\mathcal{L}_{\text{style}} \tag{2.9}$$

2.2.3 SCFT module

Spatially corresponding feature transfer(SCFT) module is the core module of SCFT network, which is used to fuse the features of line art and the reference coloring image. Different from ordinary feature fusion algorithms, it adopts the self-attention mechanism, which can combine the contextual information to target which part of the reference coloring image should be converted to which corresponding part of the line art. For example, if the line art depicts a bust of a woman with long hair, and the reference image is a full-body image of a woman with short hair, the algorithm can transfer the information obtained from the reference image to the corresponding position in the line art, for instance, the skin, eye, hair, and clothing colors of the woman with short hair will be transferred to the bust line art, and the rest of the redundancy parts will not be attended to.

The structure of this module is shown in Figure 2.5.

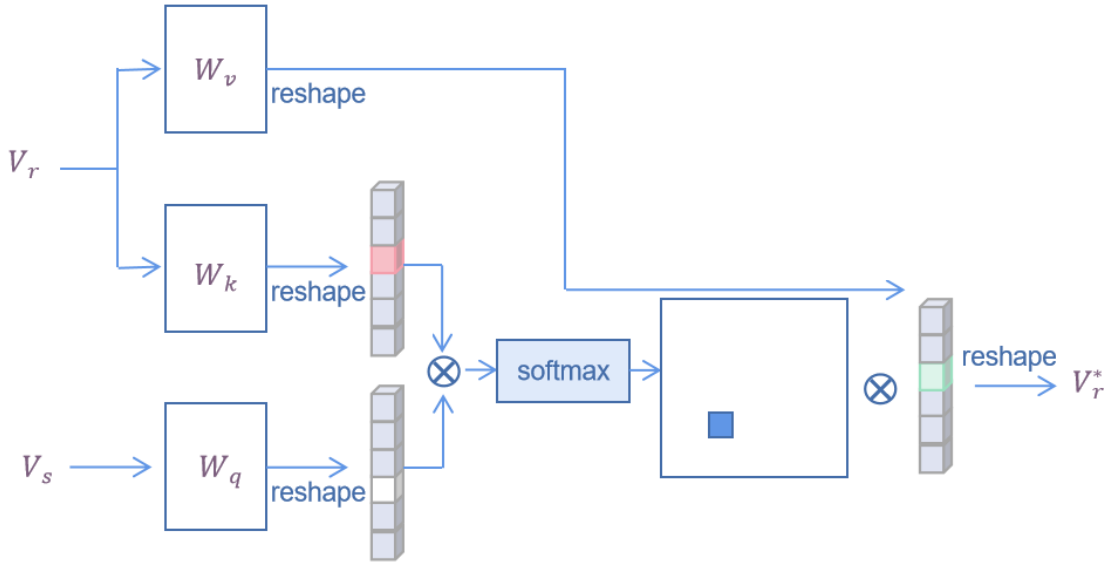


Figure 2.5: The outline of the proposed method

The activation maps generated by each convolutional layer of Encoder E_r and E_s are downsampled to the size of the last convolutional layer, then the spatially downsampling function is used, and finally the corresponding V_r and V_s can be obtained by performing the channel-wise concatenation operator. W_q and W_k are two matrices that can be used to calculate the corresponding query and key vector respectively, and W_v is a linear transformation matrix that can be used to calculate the vector of color features containing the semantically relevant parts of the reference map. After softmax activation of the key vector and query vector, the final feature fusion vector will be obtained by using the vector obtained from V_r to perform operations and reshape. This is the computational principle of SCFT module.

Chapter 3

Proposed Method

We propose a method to accomplish the coloring of animation line art by using three Pix2Pix networks and one SCFT network for better coloring. The three Pix2Pix networks are the line addition network, the coloring network and the line addition dataset production network. We will introduce the use of these four networks in detail in this chapter.

3.1 Addition of lines

Throughout the coloring process, we will first complicate the lines of the input animation line art. This is to solve the problem that the original line art is not well finished. For example, we want to color an amateur's line art, but due to the amateur's low level of skill, his line art has many unfinished parts. At this point, we perform line complication on it. When the lines are complicated, the details will increase, and then we will color the detail-rich line art, which will give us a more desirable result. Even if the original line art has a better finish, the complexification will help the network to produce better results.

The line addition is done based on the Pix2Pix network, which is what we refer to as the line addition network, which is structured as Figure 3.1. If we input an animation line art without line addition to the network, we can get an animation line art with line addition as output.

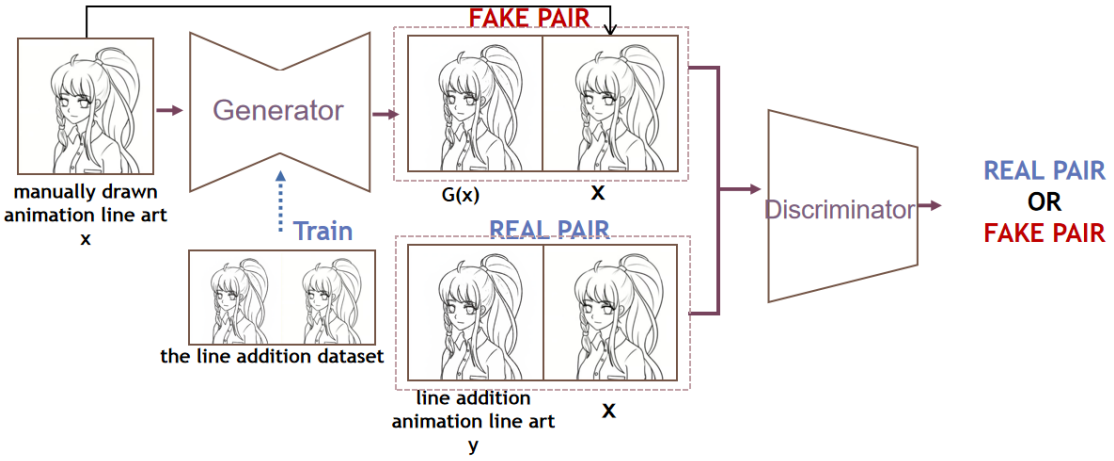


Figure 3.1: The structure of the line addition network

Training this network requires the use of pairs of data. Each image of a dataset consists of two parts, the manually drawn animation line art and the line addition animation line art.

Due to the uniqueness of this dataset, it had to be created by ourselves. First of all, we collected hundreds of manually drawn animation line art and corresponding colorized versions from the Internet, and processed the images with graphic software. Firstly, we modified the manually drawn animation line art to make it more coherent and more suitable for the coloring effect. Secondly we removed the background of the corresponding colorized image using tools like magic wand. Finally we aligned and cropped the two images to make pairs. We call this dataset Manually Colored Pair. The process of making the dataset is shown in Figure 3.2.

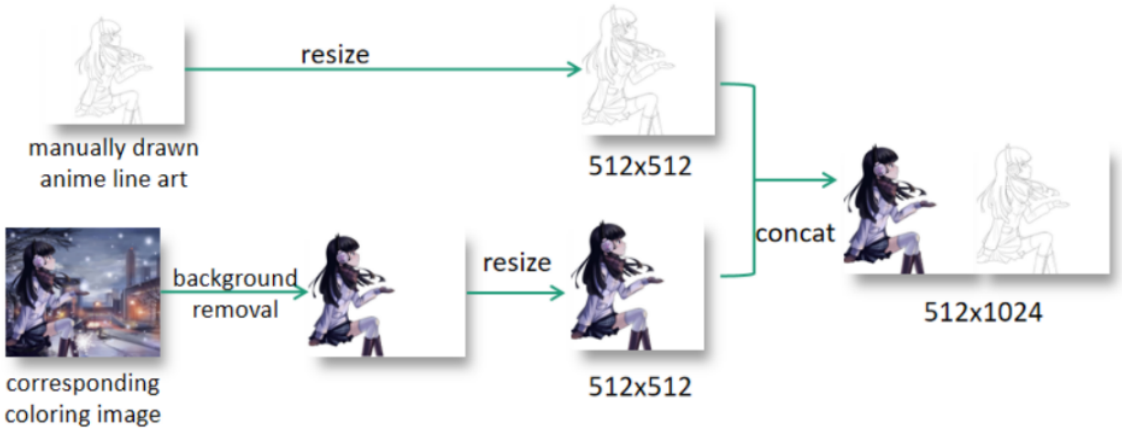


Figure 3.2: The process of making Manually Colored Pair

Next, we use the publicly available dataset Anime Sketch Colorization Pair on Kaggle as the training dataset. Each data in the dataset consists of colorized animation line art paired with corresponding line art to be extracted using some method. From this we train the line addition dataset production network as Figure 3.3. With this network, we can obtain line art as output when we take the colorized line art as input. The extracted line art is called machine extracted line art, and there is a big difference between the machine extracted line art and manually drawn animation line art. If you look closely, you can see that there are much more lines than the manually drawn line art in the machine extracted line art, but the overall smoothness of it is a little bit less than that of the manually drawn line art. Compared to the manually drawn animation line art, the machine extracted line art will give better coloring results when used as input, which is why we want to process the manually drawn animation line art.

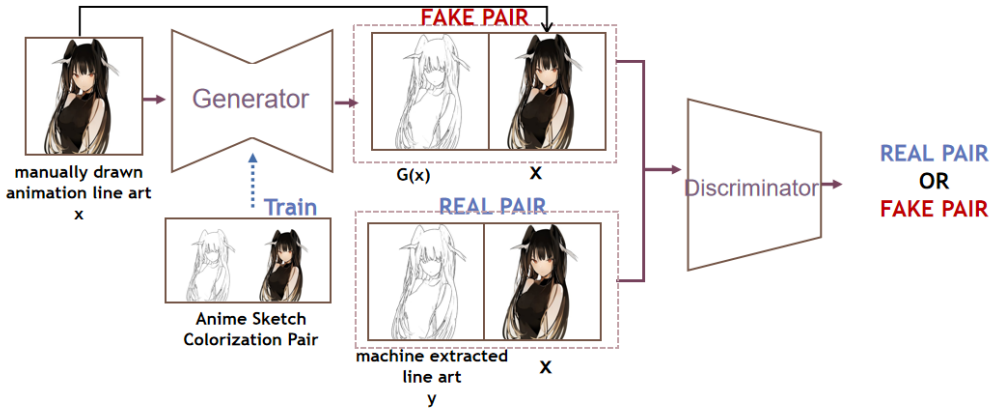


Figure 3.3: The structure of the line addition dataset production network

Finally, we stitch the manually drawn animation line art in the Manually Colored Pair dataset with the corresponding generated machine extracted line art into a single image, and complete the production of the line addition dataset used to train the line addition network. The production process is shown in Figure 3.4.

3.2 The colorization of line arts

For the part of coloring the image, we used the coloring network which is based on the Pix2Pix network, and the SCFT network. The reason for using two networks is to allow more freedom in the coloring result. For example, if you don't have any special requirements for the coloring, but just want to get a colorized line art, you can use the coloring network directly. if you want to change the image into the color of the specified reference image, you can use the SCFT network.

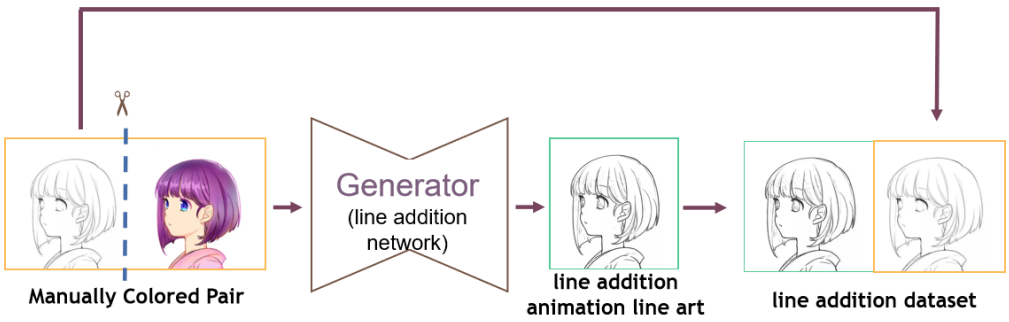


Figure 3.4: Process of producing the line addition dataset

In the coloring network, we use the Anime Sketch Colorization Pair mentioned above as the training dataset. To prevent too many confusing mix of colors from affecting the results, we filtered these datasets and tried to pick blonde-haired characters as training data. The advantage of this is that the colorization results are not messy and colorful like a rainbow, but rather cleaner. The disadvantage is that the coloring results are relatively monotonous. Figure 3.5 shows the structure of this network.

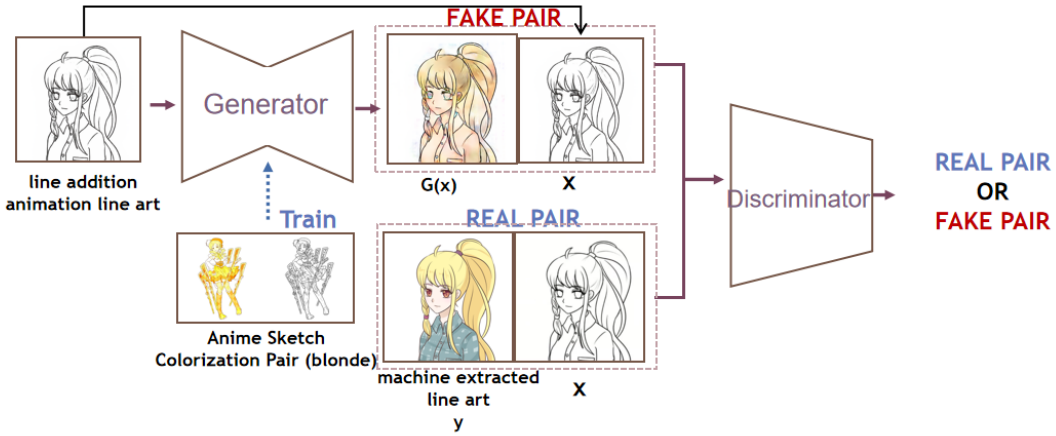


Figure 3.5: The structure of the coloring network

In SCFT network, we cropped the Anime Sketch Colorization Pair dataset and kept only the portion of color images as the training dataset. This is because the training of

this network does not require pair data, just separate color images. Figure 3.6 shows the structure of this network.

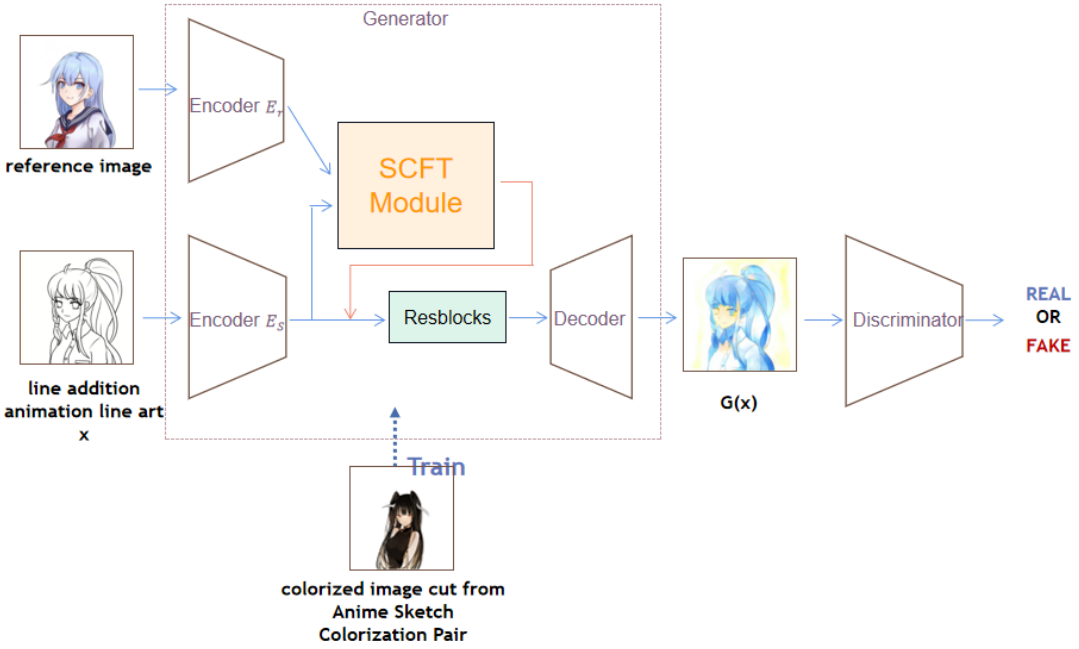


Figure 3.6: The structure of the SCFT network for coloring

Chapter 4

Experiments

In this section, we describe the dataset used in this thesis and the various experiments conducted.

4.1 Experiment setup

As introduced in the chapter 3, due to the different purposes of each network, we mainly used three datasets to complete the training and validation of the model. Manually Colored Pair, the line addition dataset and Anime Sketch Colorization Pair. The first one was created by our personal collection of manually line art and the corresponding manually colored images from the Internet, which were processed by the image processing software and produced, as shown in Figure 4.1. The second dataset was created using our network, as shown in Figure 4.2. The last one is a public dataset on Kaggle called Anime Sketch Colorization Pair, this dataset has colorized images from the web, and line art to complete the extraction using certain edge extraction algorithms, as in Figure 4.3. These images are all 512×512 in size, and the paired data is 1024×512 in size.

Considering that our coloring method is geared towards line art drawn by real people, we used a portion of the data in the Manually Colored Pair as the test data, and, which is mixed with images drawn by amateurs for more effective test results.

We compare our model results with the coloring results of Pix2Pix and SCFT networks when images without added line are used as input, and also with the results of cycleGAN [18] and Petelica [19] models.

4.2 Implementation details

We implemented the model with Pytorch 1.11.0. We used Nvidia A6000 GPU, 48GB RAM. Our batchsize was set to 4, epoch number was 200, learning rate was 0.0001. The optimizer mainly uses Adam.

Figure 4.1: Manually Colored Pair

Figure 4.2: The line addition dataset.The left one of each pair is the art line with line addition.

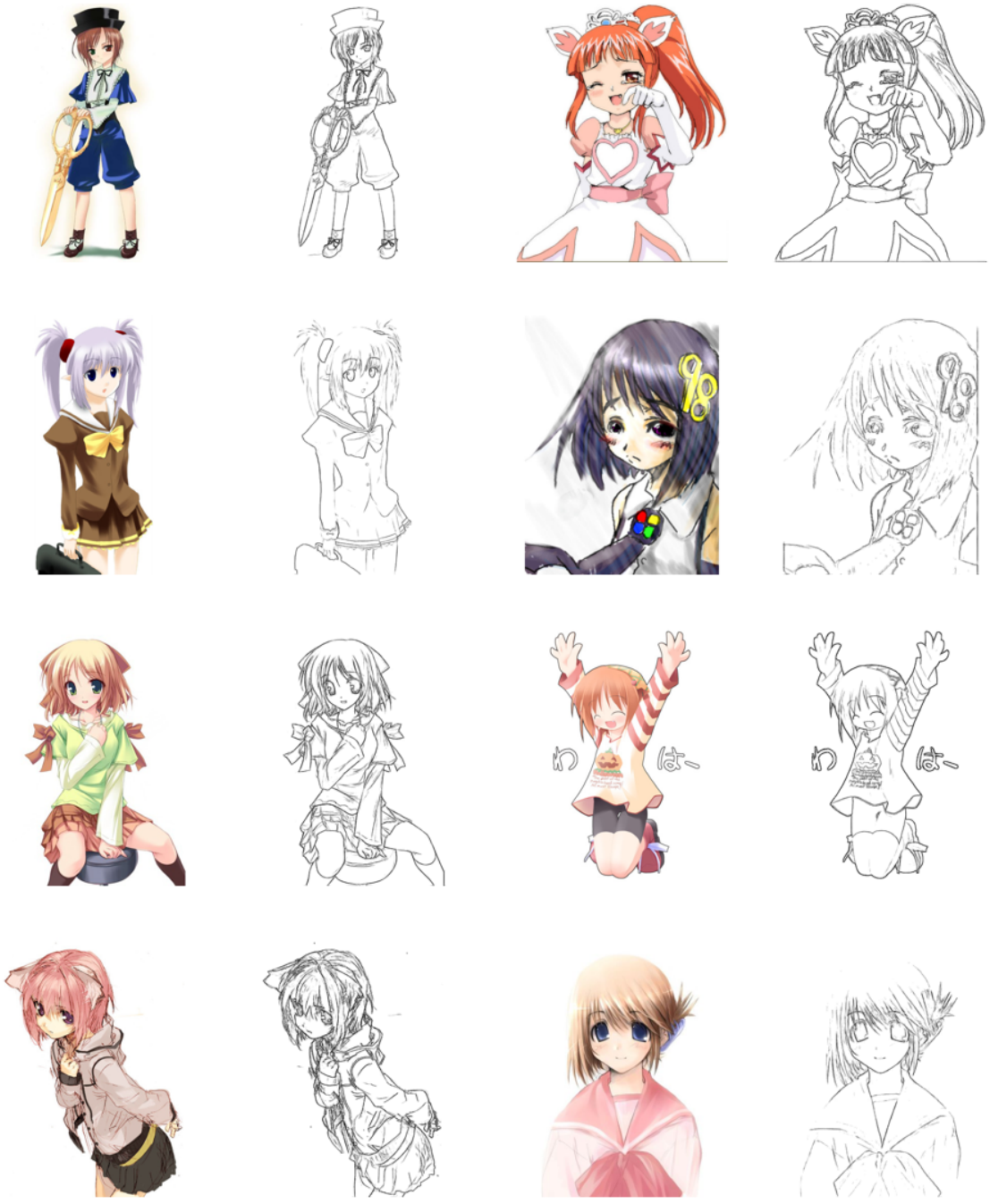


Figure 4.3: Anime Sketch Colorization Pair

4.3 Experimental results

4.3.1 Results of the Pix2Pix coloring network

As Figure 4.4 shows the final coloring results using the coloring network, the training dataset for this network mentioned above is the Anime Sketch Colorization Pair that has been classified, and when classifying it we mainly picked blonde haired characters to prevent the color from getting mixed up.

Figure 4.5 shows the result of coloring when the dataset is a random pair. It is easy to find from the image that the colors of all the characters appear blended, similar to the effect of an overturned palette. Each part of the character is mixed with multiple colors, and there is no relatively complete color block. This is because when the Pix2Pix-based coloring network is being trained, the colors in the dataset are too heterogeneous for the network to learn a uniform coloring style, at which point the result will be a mixture of colors that prevents normal coloring.

Therefore, we ultimately chose to use a filtered dataset as the training dataset. This yields results that are not overly chaotic, such as as in Figure 4.4, where all the characters are colored blonde, with red or blue eye colors. The downside of this is that the coloring results are more uniform. If we want to get more color results, we need to prepare datasets classified with different colors as the base and train the networks separately to get multiple coloring networks.

4.3.2 Results of the SCFT network

Figure 4.6 shows the final coloring result using the SCFT network.

For comparison, we also used the AdaIN [13] network, which has been applied to style transformation tasks, to try to convert line art to the color of the reference image. This AdaIN network uses the AdaIN normalization method, and we slightly modified the structure of the network by using two encoders to extract the features of the line art and the reference image, fusing them, and then going through the decoder to get the colorized image. In theory, the AdaIN network is capable of accomplishing the color transfer.

In the third column of the figure, the SCFT network is used to make the manually drawn line art girl successfully obtain the pink hair and black eyes of the girl in the reference image, and the skin color is also extremely similar. The result of AdaIN network, on the other hand, is not so good, it does not complete the transfer of color, but only blurs the color of the reference image.

From the results we can see that the SCFT network can complete the color transfer task of the reference image. And AdaIN network is still far away from accomplishing this task.

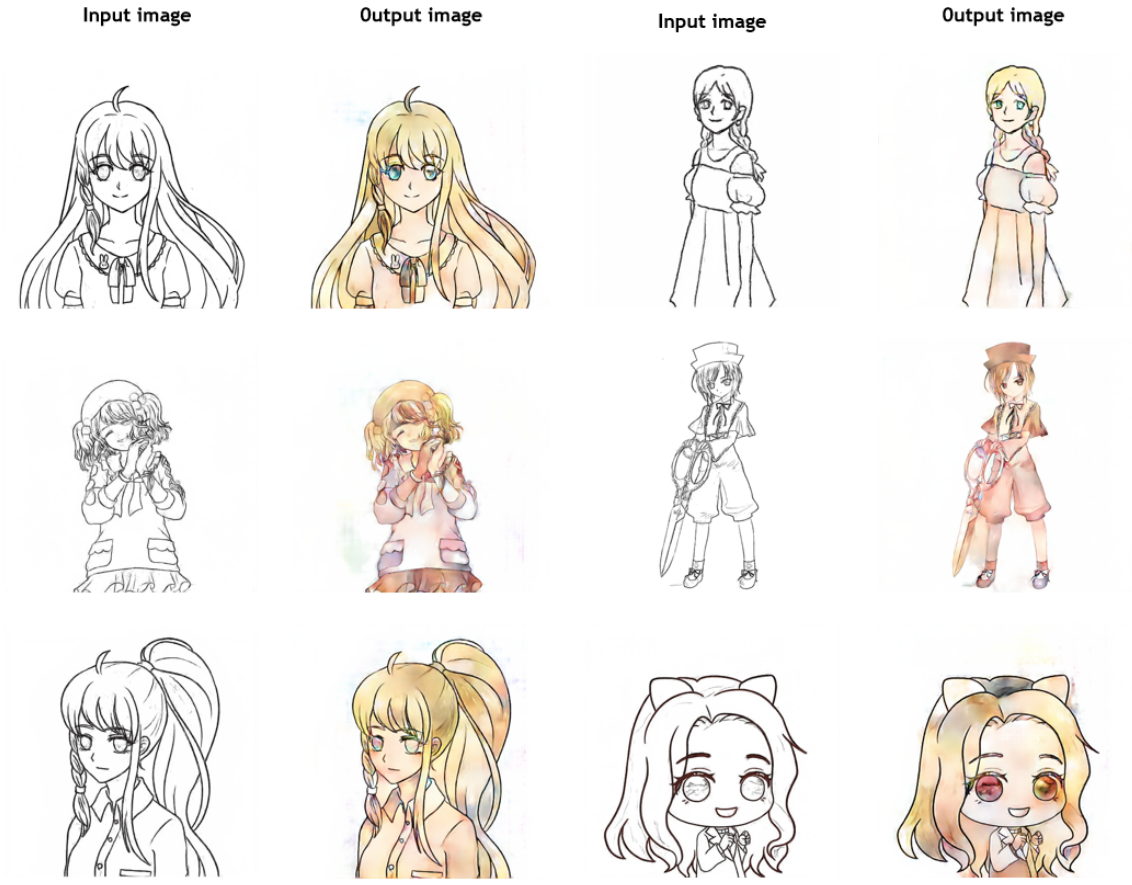


Figure 4.4: Results of the Pix2Pix coloring network trained from classified data

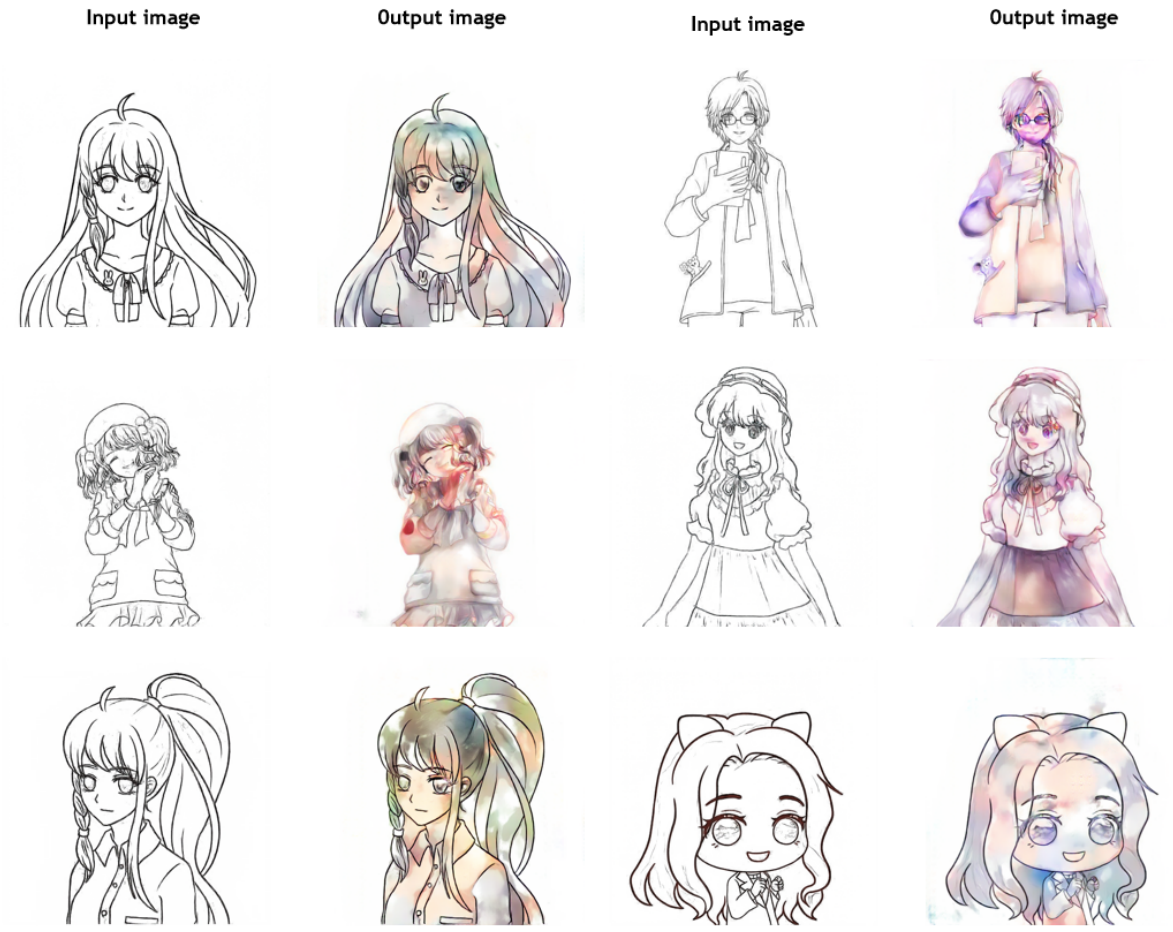


Figure 4.5: Results of the Pix2Pix coloring network trained from random data

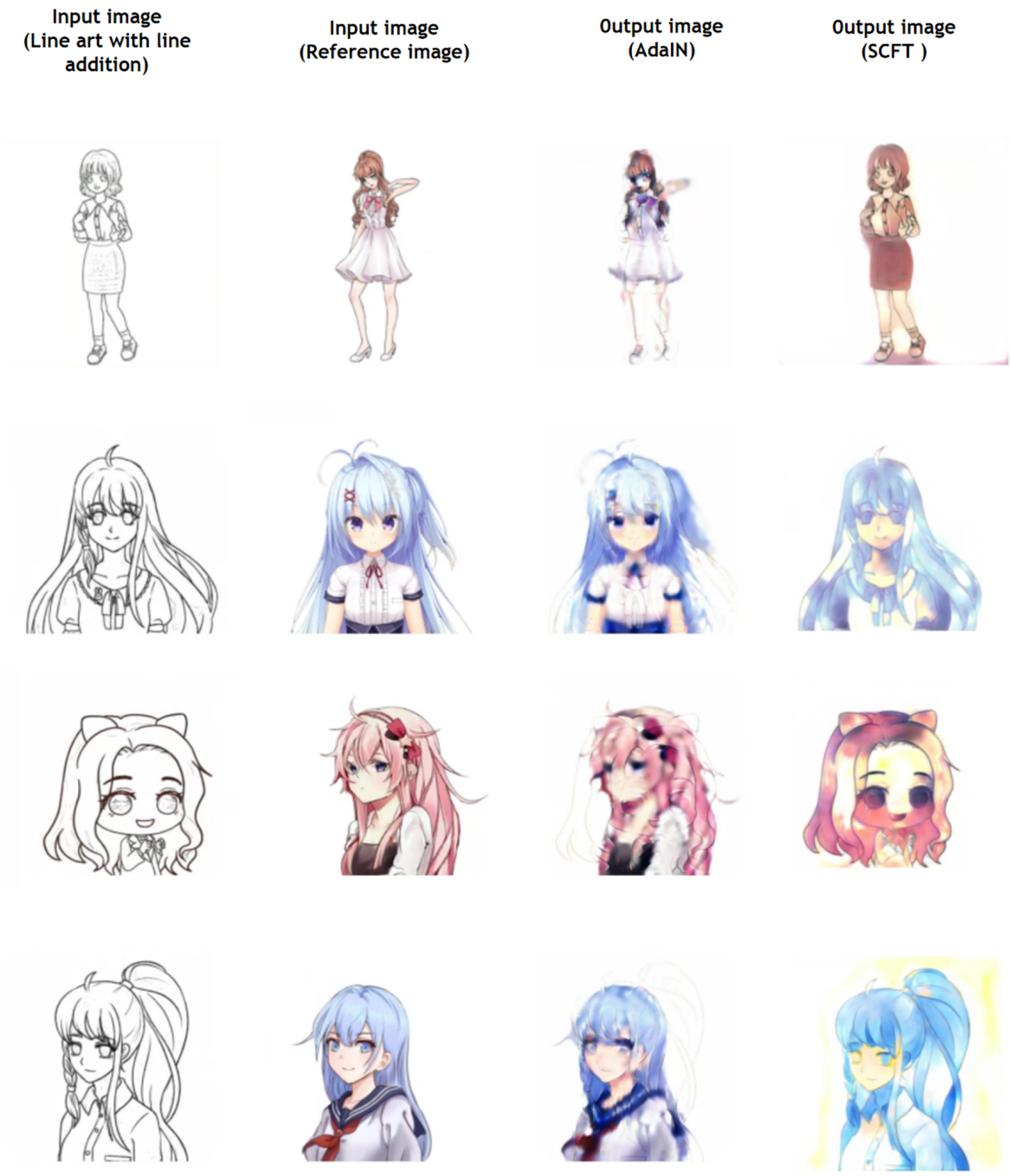


Figure 4.6: Results of the SCFT network and the AdaIN network

4.3.3 Line addition comparative experiment

In this paper, we propose the idea of adding line to manually drawn animation line art, aiming to get better coloring effect. In order to confirm the effectiveness of adding line, we conducted a set of comparative experiments by taking manually drawn animation line art without added line and line art with added line as the inputs to the coloring network. The results we get are shown in Figure 4.7.

It can be seen that when line art with the line addition is used as input, the output reflects the exact structure of the eyes. For example, in the first column of the picture of the girl with long hair, the girl's eyes are successfully colored blue, and the pupils and highlights can be seen, compared to line art without the line addition, where this is ignored, and the eyes are either white or close to the skin color. In addition, the line art with the line addition is more concentrated than the line art without the line addition, and the color overflow is slightly better than the line art without the line addition.

To summarize, with line addition, the detailed parts of the line art were better focused. Without it, some of the details were simply left blank, making the coloring less effective.

In addition, we also tested the SCFT network in the same way and got the results as shown in Figure 4.8. Through the resultant images we can find that when the line art with the line addition is used as input, the model can successfully distinguish the various parts of the character, such as hair, eyes, skin and so on. Migrating the color of the reference image, for example, the girl with long hair in the second column, as an output image successfully obtains the blue hair and blue eyes features of the reference image and has a similar skin color as the girl in the reference image. However, when the line art without added line is used as input, the result is not able to achieve the color migration well, such as the long-haired girl in the same second line, the blue hair almost covers the whole figure and it is not possible to distinguish the parts of the figure.

Therefore, in the case of the same reference image, the line art with the line addition enables the SCFT network to colorize more accurately, while the one without the line addition faces problems such as overflow of coloring results.

Therefore, from the results of the two coloring networks we prepared, the addition of line is necessary to improve the coloring accuracy and allow the coloring task to be done better. We believe that this method of handling line art can make the line art drawn by people more convenient to be handled by the network, and it can be applied not only to the two coloring networks we prepared, but also to more coloring networks theoretically.

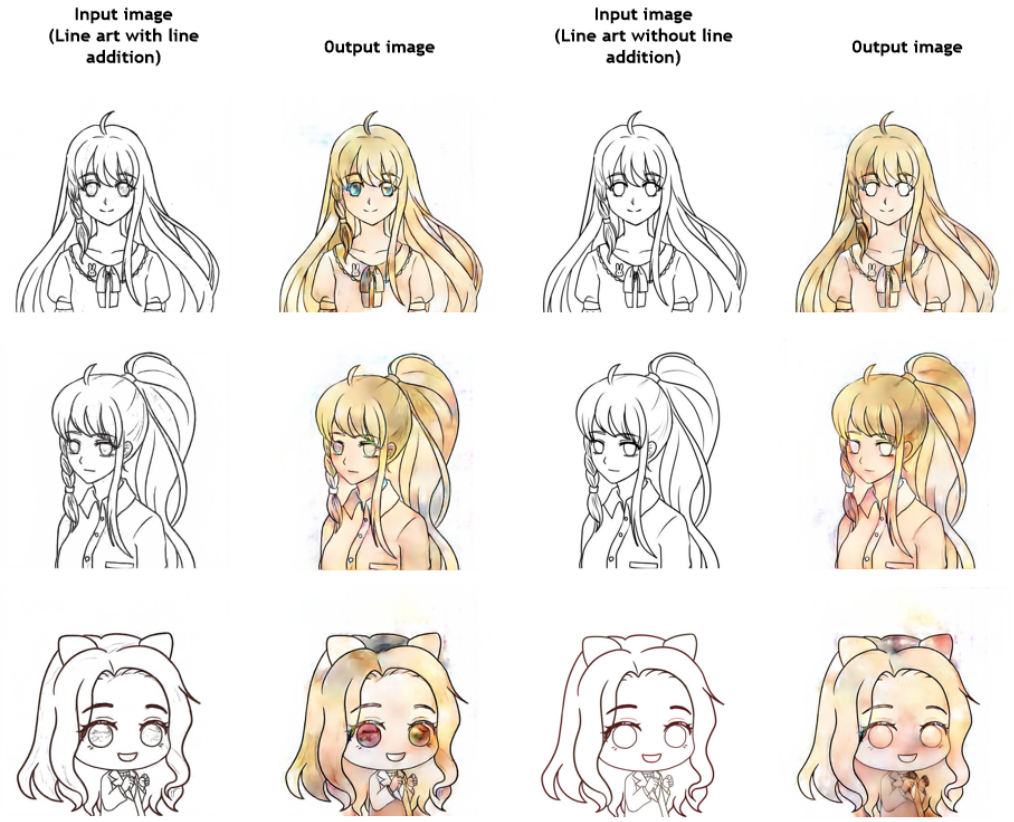


Figure 4.7: Results of the coloring network when line art with added lines is used as input and line art without added lines is used as input

Figure 4.8: Results of the SCFT network when line art with added lines is used as input and line art without added lines is used as input

4.3.4 Comparison of results of different coloring networks

To test the validity of our proposed method, we tested the results of several coloring models. We selected CycleGAN [20], Petalica Paint [19] and the original Pix2Pix model for experiments. The experimental results are shown in Figure 4.9.

For the Pix2Pix network, as described in the previous section, when a line art is directly taken as input, the output will be more confusing. As shown in the second column of the figure, not only will there be a chaotic effect of fusion in terms of color, but there will also be a lot of color overflow, which will not be able to complete the coloring task well.

For CycleGAN, that is, the image in the third column of the figure, we can find that although it generates a more uniform image color, the effect is relatively bland, the color is not rich enough, and the overall tendency is skin color and red. And it results in a lot of white parts have not been successfully colored, including the eyes and other details can not all be colored, accompanied by black noise of unknown origin.

For Petalica Paint, the fourth column of the coloring results, we can see that this scheme is relatively cleaner, but has the same problems as CycleGAN, there is a lot of unnatural white space in the coloring parts, and the details, such as the eyes, do not have a three-dimensional sense of the coloring, but are simply filled with a piece of color.

The fifth column shows the results of our proposed method. Compared to other coloring methods, our method is richer, although the color tends to be yellow. Moreover, our method can pay better attention to the details, and the coloring of the hair and eye parts is more comprehensive and three-dimensional. However, our image also has the problem of noise, which will be our future improvement goal.

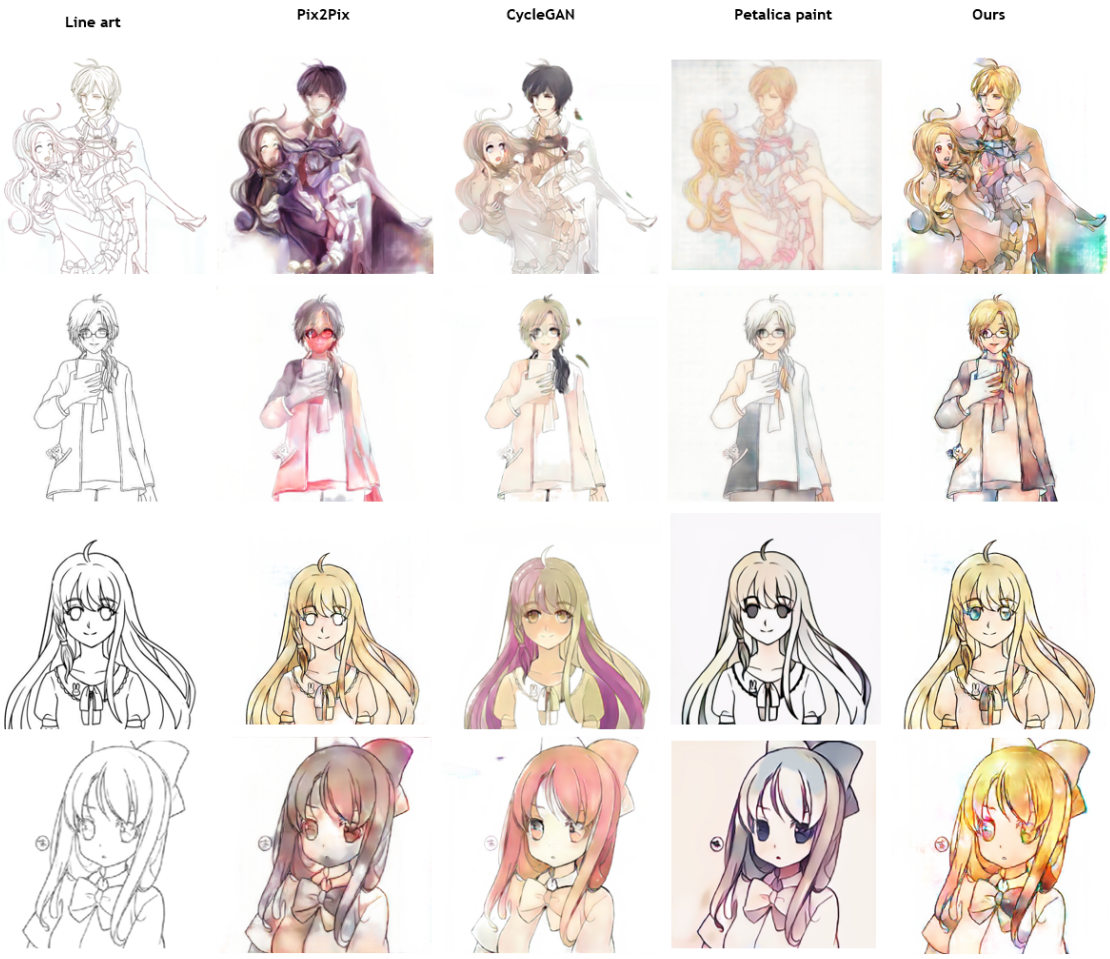


Figure 4.9: Results of different coloring networks

Chapter 5

Conclusions

In this paper, we introduce an animation line art coloring method based on four networks, and getting a better coloring effect by adding lines to the line art to be colored.

First we use the line addition dataset production network based on Pix2Pix network to generate a pair dataset for training the line addition network. This dataset is then used to train the Pix2Pix based line addition network, which is used for the purpose of adding lines to the line art. Finally, we use another Pix2Pix based coloring network for final coloring. In order to make the coloring results more selective, we also prepared the SCFT network, which allows the image to be colored according to the reference image provided. The experimental results show that our proposed scheme is effective.

In the future, we consider to further optimize the network and work on generating higher quality and less noisy coloring images.

Acknowledgement

I would like to express my sincere gratitude to Prof. Yasuaki Ito, Prof. Koji Nakano and Associate Prof. Sayaka Kamei for their enthusiastic guidance, valuable comments and suggestions during their busy schedule. I would also like to express my sincere gratitude to all the members of the laboratory for making my research environment comfortable.

References

[1] H. Tan, "Cartoon Animation and Architectural Animation Art Expression form Research," *Chinese Journal of Radio and Television*, p. 3, 2017.

[2] Z. Huang, "On the Artistic Creation Characteristics of the "Cartoon Generation"," *New West: Theory Edition*, p. 2, 2011.

[3] G. Law, "Quantitative Comparison of Flood Fill and Modified Flood Fill Algorithms," *International Journal of Computer Theory and Engineering*, pp. 503–508, 2013.

[4] J. Hays and A. A. Efros, "Scene Completion Using Millions of Photographs," p. 4 – es, 2007.

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, p. 24, 2009.

[6] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," vol. 5. PMLR, 2009, pp. 448–455.

[7] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations." Association for Computing Machinery, 2009, p. 609 – 616.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," 2014.

[9] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations*, 2013.

[10] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2016.

[12] C. Li and M. Wand, "Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks," 2016.

[13] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization," 2017.

[14] J. Lee, E. Kim, Y. Lee, D. Kim, J. Chang, and J. Choo, "Reference-Based Sketch Image Colorization using Augmented-Self Reference and Dense Semantic Correspondence," 2020.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 2017.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015.

[17] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," jun 2015.

[18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," 2017.

[19] "Petelica Paint." [Online]. Available: https://petalica.com/index_ja.html

[20] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," 2017.