Dalian University of Technology
Undergraduate Capstone Project (Thesis)

Textual Description Based Colorization of Scene Sketches

Department:	International School of Information Science & Engineering
Major:	Software Engineering
Name:	Chengxiao Li
Student Number:	201894076
Supervisor:	Associate Prof. Yu Liu
Review Teacher:	Associate Prof. Qi Jia
Completion Date:	2022-06-13

大连理工大学

Dalian University of Technology

原创性声明

本人郑重声明：本人所呈交的毕业设计（论文），是在指导老师的指导下独立进行研究所取得的成果。毕业设计（论文）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究成果做出重要贡献的个人和集体，均已在文中以明确方式标明。

本声明的法律责任由本人承担。

作者签名：李承潇　　日　期：2022/06/04

关于使用授权的声明

本人在指导老师指导下所完成的毕业设计（论文）及相关的资料（包括图纸、试验记录、原始数据、实物照片、图片、录音带、设计手稿等），知识产权归属大连理工大学。本人完全了解大连理工大学有关保存、使用毕业设计（论文）的规定，本人授权大连理工大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业设计（论文）。如果发表相关成果，一定征得指导教师同意，且第一署名单位为大连理工大学。本人离校后使用毕业毕业设计（论文）或与该论文直接相关的学术论文或成果时，第一署名单位仍然为大连理工大学。

论文作者签名：李承潇　　日　期：2022/06/04

指导老师签名：刘宇　　日　期：2022/06/04

摘 要

草图着色是一个非常有趣和实用的研究方向。同时自然语言是一个非常有前途的输入方式，结合自然语言输入与别的模态的用户输入，可以获得更加实用的系统。

在之前基于 GAN 生成网络的草图着色的系统中，包含了实例匹配，前景着色和背景着色三部分，使得系统能够分别对前景的物体进行上色并在前景上色后对背景进行着色，提高了系统上色的准确性。因此本论文继续沿用实例匹配，前景着色和背景着色的结构。

但是原结构仍然存在一些问题，在原结构中使用了 LSTM 进行文本编码和文本-图像信息融合，但是因为着色时的文本通常较短，因此不需要使用 LSTM 进行长时间的记忆，此外使用 LSTM 使得系统在训练时所消耗的时间较长。因此本论文主要在背景着色和前景着色部分针对这两部分进行提升。在文本编码部分使用 Bert 对文本描述进行整句的编码，在文本-图像信息融合部分使用 Residual Block 进行文本-图像信息融合，最终使草图着色可以获得更好的上色效果并能够更快地训练。

本论文使用了一个大规模的场景草图和卡通风格的彩色图像库以及对应的文本描述。前景着色和背景着色分别通过输入的前景草图和背景草图以及对应的文本描述来生成上色的图片。最终和原有系统的上色结果进行对比，本文中提出的结构获得了更好的上色结果以及拥有更快的训练速度。

关键词：文本描述；草图着色；GAN；Bert；Residual Block

Textual Description Based Colorization of Scene Sketches

Abstract

Sketch coloring is a very interesting and practical research direction. Also natural language is a very promising input method, and combining natural language input with user input from other modalities can lead to more practical systems.

In the previous system of sketch coloring based on GAN generative network, it contains three parts: instance matching, foreground coloring and background coloring, which enables the system to color the objects in the foreground and color the background after the foreground coloring respectively, which improves the accuracy of the system coloring. Therefore, this thesis continues to follow the structure of instance matching, foreground coloring and background coloring.

However, there are still some problems with the original structure. In the original structure, LSTM is used for text encoding and text-image information fusion, but since the text is usually short when coloring, there is no need to use LSTM for long time memory, and in addition, using LSTM makes the system consume longer time when training. Therefore, this thesis mainly focuses on the background coloring and foreground coloring part to improve these two parts. In the text encoding part, Bert is used to encode the whole sentence of the text description, and in the text-image information fusion part, Residual Block is used to fuse the text-image information, so that the sketch coloring can get better coloring effect and can be trained faster.

A large-scale dataset of scene sketches and cartoon-style color images with corresponding text descriptions is used in this thesis. Foreground coloring and background coloring are generated from the input foreground and background sketches and the corresponding text descriptions to colorize the images. In comparison with the coloring results of the original system, the proposed structure in this paper achieves better coloring results and faster training speed.

Key Words: Textual Description; Sketch Colorization; GAN; Bert; Residual Block

テキストの記述に基づくスケッチカラーリング

概要（日本語）

スケッチカラーリングは、非常に興味深く、実用的な研究分野です。また、自然言語は非常に有望な入力手段であり、自然言語入力と他のモダリティからのユーザー入力を組み合わせることで、より実用的なシステムを実現することができる。

これまでの GAN 生成ネットワークに基づくスケッチ色付けシステムでは、インスタンスマッチング、前景色付け、背景色付けの 3 つのパートが含まれており、前景オブジェクトを別々に色付けし、前景色付け後に背景を色付けすることで、システムの色付けの精度を向上させることができました。そこで、本論文では、引き続き、インスタンスマッチング、前景シェーディング、背景シェーディングの構造を踏襲することとした。

しかし、テキストエンコーディングとテキスト　画像情報融合に LSTM を使用する独自の構成にはまだ問題があります。しかし、通常、テキストは色付けすると短いため、長時間記憶のために LSTM を使用する必要はなく、LSTM を使用するとシステムの学習時間が長くなってしまいます。そこで、本論文では、背景色と前景色の部分に着目し、この 2 つの部分を改善することにしました。テキスト符号化部では、Bert を用いてテキスト記述を完全な文章で符号化し、テキスト　画像情報融合部では、Residual Block を用いてテキスト　画像情報を融合し、最終的にスケッチの色付けを良くし、学習速度を向上させる。

この論文では、シーンスケッチや漫画風のカラー画像とそれに対応するテキスト説明の大規模なライブラリーを使用しています。前景色付けと背景色付けは、それぞれ入力された前景と背景のスケッチと対応するテキスト記述からカラー画像を生成するために使用される。オリジナルシステムの着色結果と比較すると、提案構造はより良い着色結果とより速い学習速度を達成しています。

キーワード: テキスト記述、スケッチ着色、GAN、Bert、Residual Block

Catalogue

摘要 i

Abstract ii

摘要（日语） iii

1 Introduction 1

1.1 Research Background and Significance 1

1.2 The Research Status of Textual Description Based Colorization of Scene Sketches 6

1.2.1 International Status for Theoretical Research 6

1.2.2 Domestic Status for Theoretical Research 9

1.2.3 Existed Method of Textual Description Based Colorization of Scene Sketches 11

1.2.4 Limitations of Existing Methods 12

1.3 Research Target and Main Content 12

1.4 Summary 13

2 Preliminary 14

2.1 Text Encoding Method 14

2.1.1 Bidirectional Encoder Representation from Transformers 14

2.1.2 Long Short Term Memory 17

2.1.3 Gate Recurrent Unit 18

2.2 Summary 19

3 Network Architecture 20

3.1 Foreground Colorization 20

3.2 Background Colorization 27

3.3 Summary 29

4 Experiment 30

4.1 Dataset 30

4.1.1 Foreground Colorization Dateset 30

4.1.2 Background Colorization Dataset 31

4.2 Experimental Equipment 33

4.3 Training Parameters 34

4.3.1 Training Parameters of Foreground colorization 34

4.3.2 Training Parameters of Background colorization 35
4.4 Experimental Results and Analysis 35
4.4.1 Experimental Results of Forground Colorization 35
4.4.2 Experimental Results of Background Colorization 37
4.5 Summary 39
5 Limitation and discussion 40
5.1 Foreground Colorization 40
5.2 Background Colorization 42
5.3 Summary 43
Conclusion 45
References 46
修改记录 50
Acknowledgement 51

1 Introduction

Sketch coloring is a very interesting and practical research direction. Given a grayscale sketch, existing methods use deep neural networks to estimate the color information of objects. However, due to the lack of prior knowledge, the color variability of the image is too large. For example, the clothes worn by people can be filled with various colors. In order to constrain the solution space, this paper aims to improve the method of textual description based colorization of scene sketches, so that the entire sketches can gradually colored by natural language interaction.

1.1 Research Background and Significance

In recent years, comics have become one of the spiritual food of more and more people, and are deeply loved by young people. Under this general trend, more and more people choose to engage in the comics production industry, and more and more cartoonists devote themselves to comics creation, producing one after another well-known works. However, coloring comics has always been a very troublesome step for cartoonists. On the one hand, a pleasing comic work must have very high requirements in terms of color. On the other hand, comic coloring is a very complicated and technical content. Especially high work. Therefore, if the automatic coloring of comics can be realized, the work steps of comic creators will be greatly reduced, so as to achieve the purpose of efficient production of comics, and allow comic creators to focus more on conceiving the plot of comics. Due to its intuitive and easy-to-draw features, people widely use hand-drawn sketch data to express and summarize their knowledge and understanding of objective things: For example, cartoonists use sketches to express a certain point of view or idea, and children use sketches to create stories, designers use sketches to capture moments of inspiration in their minds. Humans' knowledge and understanding of objective things can usually be reflected in the sketches they draw. Research in the field of psychology believes that "The way children draw at the age of four may be a predictor of later intelligence", that is, the way children draw can be used to predict their future level of intelligence. Psychologists have shown that cognition is an important way to help children's intellectual development. While a language-based interface may not provide good or direct control over results, it does offer unique properties, including naturalness for everyone, enabling interesting applications that are difficult or even impossible to achieve with existing interfaces. Recent research has also found its unique benefits for children. Sound input can inspire children to explore and have fun. Furthermore, by recording children's storytelling, drawing can provide a more effective system to support children's literacy development.

Most Japanese manga are mainly in black and white, and there are few pages in a manga magazine that are called "color pages" by the Japanese. Although Japanese comics are popular around the world, adapting animation and games, but there may always be a question lingering in the minds of all comic readers, Japanese comics lead the world, why is always black and white? U.S. comics such as "Marvel", "DC" such as the behemoth aside, the slower start of Korean comics, national comics are also in full color. Although Japan has superior black and white comic skills, color comics are obviously more expressive than black and white comics, and readers will have a better reading experience, which seems to be something that has a hundred advantages but never becomes a reality. The reason for this is the difference in drawing tools and media. In Japan, due to the early start of manga, it was customary to draw on paper, and coloring was extremely laborious; printing paper was of poor quality to save costs, and black and white printing did not magnify this shortcoming. Only the popular works in the weekly magazines were rarely painted in color, and this habit has been preserved. It's like how China jumped into the era of cell phone payments without going through the credit card era, while Japan still lags behind in digital payments, so it is still accustomed to credit cards and banknotes. The later ones do not have the trouble of "accumulation of heavy weight", and the new wind of technology has blossomed on the layers of China. Dragon Ball", the Saiyan transformation after the setting of golden hair is also to reduce the amount of painting, in black and white comics, the golden color can be left white processing, rather than the need to paint Goku's hair all black. Even to the character hair color time needs to be saved, which can also be a side testimony to the huge workload of Japanese cartoonists. However, hair color and the character's combat power linked to the setting but also coincidentally became a major classic in the ACG works. To this day, Japanese manga are still mostly drawn by hand in the traditional way. The manga artist is the main person, and the assistants draw efficiently according to the setting and style written by the manga artist. At the same time, the assistants have a more detailed division of labor, such as drawing scenes and buildings, which is also a division of labor arising from black and white manga. In the case of color manga, the time spent by the individual manga artist is doubled in the division of labor. Most Japanese manga magazines are published once a week and have a huge circulation, so the black and white system not only reduces the time needed to draw, but also the money spent on printing in black and white and color. Therefore, the Japanese manga industry is mainly black and white. However, it is not that the habit of drawing in black and white is not good because of the limited human and material resources. In black and white manga, the author's skills are tested in terms of drawing. The lines, strokes, and sense of space are no less important than a black and white sketch. At the same time, black and white cartoons without excess color, in the visual more impactful. Most domestic comics are mainly in color, which is more appealing to most people's visual senses. Today, with the rapid development and popularization of Chinese network, the

prevalence of web comics is no less than that of online literature, and it is more graphic than online literature. And with the variety of drawing software, only one person can finish a small colorful work in a short time. (This is not to belittle the quality of the many bad works on the Internet, but to emphasize the speed of the popularity of cartoons). Most domestic manga artists are used to using software to draw, so the popularity of color manga does not require as much effort to consider human and material resources as the Japanese manga industry. However, on the major manga websites, most of the manga are published in the name of studios, if not by famous manga artists, so it is clear that even for color manga drawn by computer, it still depends on the division of labor. If we talk about the disadvantages of this division of labor system, the works drawn under the name of a studio are more likely to lack their "personal style" than black and white manga, which emphasizes drawing skills, and even with the support of color, they still look monotonous and ordinary. However, a good color manga, with computer brushes, color adjustments, etc., can often express the atmosphere of art better than pure black and white.Secondly, the way of "looking". Like Japanese books, Japanese manga, mainly paper magazines, are read in the same way as the customary printing in the Japanese printing industry. That is, it is read from right to left. The grids are concentrated on each page, and except for a small number of gaps, not a single inch of the picture is wasted. (Perhaps this is one of the reasons why black-and-white comics are visually impactful.) The fact that domestic comics do not consume paper and have a large number of color combinations makes it more important to pursue the sense of gaps in the layout. Such a sense of space may also be called white space. The most frequently used white space in all kinds of art works is actually Chinese painting, which makes the picture more coordinated and not depressing, and at the same time can lead to the viewer's imagination. At the same time, in a page of comics, each sub-panel is separated from the other by a small amount of white space, which makes the whole page look like a long strip. The top-down view is very suitable for both computer pages and cell phones. Under the same conditions, it is true that the cost of color comics is higher, but as color comics, consumers are more receptive to color comics, even if the paper used for printing is particularly poor, printing color comics are not ugly, consumers can accept, while black and white comics are different, black and white comics due to its fundamental reasons, if the degree of consumer acceptance is very high, and because the national comics have not formed a mature industry chain, around The printing factory for the printing of comics is not uniform, it is difficult to print high quality, can be accepted by consumers of black and white comics, which leads to the printing of black and white comics in the country, the cost is higher. Based on this, the national comic has gradually formed the form of mainly color comics, while the Japanese comic is still mainly black and white comics industry, which is the reason why the Japanese comic is mainly black and white comics, while the national comic is mainly color comics.

The serialization of anime is often years and decades, but no matter what the identity background, is a wealthy or poor characters, in people's memory they rarely change clothes. This is the difference between anime and TV series, the female protagonist of the TV series seems to be performing a fashion show, while the main character of the anime is the same dress from the first episode to the last. Although in the anime do not count those who do not meet the reality of the place, but everything as long as there is his reason, for example, Silverthorne once said he had four identical kimono, Crayola also gave us to see him full of a large closet of red T-shirts and yellow shorts. Although the unchanging clothes can be used to emphasize the character's personality, cheerful and lively people rarely wear dark clothes, and introverted people with low self-esteem rarely wear pink clothes. And it can be used as an anchor point to make people remember the character, but at the same time it can also make the picture tend to be monotonous. An anime is never a one-man show for the protagonist, who will inevitably be surrounded by a lot of pals. Most of the early Japanese comics are about adventure, and often a group of friends go on a trip together. The early anime has a lot of fight scenes, so the early artists will focus on the fight scenes, and no time to take into account the design of clothing and color. Another point is that an anime will not have only one artist. For example, when we look at "Crayon Shin-chan", it is obvious that the style of drawing is different. Cat and Mouse" is an even more obvious example, sometimes even the breed of cat is not the same. But if automatic coloring of manga is possible, then instead of the huge workload of coloring, manga workers can just draw the outline of the clothes and color them by computer with simple language descriptions, which will greatly enrich the picture of manga and make the characters closer to reality, and this work can also be left to the readers, bringing the readers closer to the manga workers. This work can also be done by the readers, bringing them closer to the cartoonist and thus creating a character that is more relevant to the public.

Therefore, if the black and white manga in Japanese manga can be turned into color manga through the machine and simple description, then it will greatly reduce the workload of manga workers, and the coloring work does not need a professional artist to describe, and even through the reader to color the black and white manga, such as in the reader's imagination which colors this character or object should be composed of, this can not only unknowingly This can not only complete a large amount of coloring work but also enhance the interaction between readers and comics, which has a strong sense of participation and immersion of readers.

In addition, comics have many benefits for children. Children are much more sensitive to the perception of pictures than other forms of information. The rich content of comics makes children feel happy while looking at them. The picture books that people read as children are in the form of pictures more than words. Comics through the vivid character image of specific

scenes of the portrayal, can let everyone produce a sense of immersion, can be immersive to feel the content of the scene, can go to heaven and earth, omnipotent. Very representative of the United States is the DC Comics and Marvel Comics, part of these comics have been moved to the movie screen, comics portrayed a wonderful, different universe, where the characters also have a strong personality charm, attracting many people. So if a language-based sketch coloring system can be implemented, it can facilitate a child's development. Coloring comics through verbal descriptions can develop children's imagination and help them improve their verbal skills and learn more about coloring and drawing while having fun.

The animation industry is a "sunrise industry" in China, and the total output value of China's animation industry has exceeded 150 billion yuan in 2018, and the scale of online animation users has also exceeded 200 million. As a form of art creation, cartoons are drawings depicting characters and events in a realistic or exaggerated way. As the provider of content, manga artists and manga studios have great value for their creativity and original works drawn. Depending on the carrier, comics can be divided into two categories: paper comics and digital comics". Traditionally, cartoons are presented in the form of original drawings on paper by cartoonists and published by publishers or magazines. Nowadays, with the popularity of digital media such as cell phone and internet, more and more cartoonists use digital boards, iPads and other tools to create cartoons and use various online platforms to disseminate and read them, and the more common drawing software are Photoshop, SAI and so on. Paper comics need to be photographed or scanned to get the electronic version of the images, but digital comics do not need this step. With the automatic coloring method of sketching in this thesis, the work of cartoonists can be simplified. For beginners, it can help them learn coloring faster by text-based sketch coloring, and it also lowers the threshold of the comics industry so that more and more people can tell stories through comics and join the comics industry.

Sketch coloring is generally divided into two ways, unguided coloring and guided coloring. Unguided coloring refers to fully automatic coloring by algorithms. Users only need to provide a line draft to be colored, and then a finished coloring draft can be generated by calculation. Guided coloring means that there are some color prompts provided by the user during the coloring process, such as color prompts for specific areas , color style reference maps textual description and so forth to generate corresponding coloring images with restrictions. Sketch coloring is a very interesting and practical direction of research. At the same time, natural language is a very promising input method, and a more practical system can be obtained by combining natural language input with other modal user input. Coloring sketches through language can provide a new way of cognitive development in children, or a reference scheme for grouping black and white comics into cartoon books. It can save the manga author's workload, and can also provide help in learning drawing and coloring.

1.2 The Research Status of Textual Description Based Colorization of Scene Sketches

1.2.1 International Status for Theoretical Research

In instance-based automatic colorization techniques, grayscale images are colored by a reference image. Welsh et al.[1] proposed a colorization method based on image brightness and texture information to automatically colorize the target gray image by passing color information between the reference image and the target gray image. This method does not require interaction, only a color image needs to be provided. Instead of randomly selecting colors from the palette to render to a certain area of the target gray image, the global color information of the reference image is passed to the target gray image, while retaining the original brightness of the target gray image. The color image texture and brightness generated by this method are clear enough, and the image quality is relatively high. Schillings et al.[2] proposed a method to colorize a target grayscale image by dividing it into two parts. The first part is a grayscale image separation algorithm based on Bayesian structure, which efficiently processes complex images with slightly different boundaries by introducing alpha distribution and gradient operator in Bayesian structure. In the second part, a smoothing algorithm for grayscale images is combined with a color transfer method in the feature space to produce high-quality color images. The algorithm proposed in this paper can be successfully used for portrait rendering.

With the development of the research of automatic image coloring without any prompt information, grayscale images can be fully auto-colored by auto-color technology. J Rock et al.[3] proposed to use the LEARCH framework to train the quadratic function relationship between gray images and color images. The model learns colorization methods from large batches of paired data, and then colorizes the input gray image by minimizing the objective function. G Larsson et al.[4] implemented a fully automatic black and white image colorization system, using deep network to learn the detailed features of color images, and training the model to predict the color information of each pixel in the black and white image, so as to realize the colorization of black and white images. S Iizuka et al.[5] proposed a new automatic coloring method for comics. The features of the image include global features and local features. The global features include the overall outline of the image, and the local features include some details of the image. The global feature and local feature of the image are combined to train the model. Based on the convolutional neural network, the network includes two sub-networks, the local feature extraction network and the global feature extraction network, so as to achieve the purpose of processing comic images of any resolution. .

In recent years, with the popularity of deep learning, the mainstream of cartoon coloring

that has gradually developed is realized by two different methods, including simple coloring algorithms and deep learning models[6] [7] [8] [9]. In the method based on deep learning, various deep learning models have been tried to complete the task of image translation. Among them, many researchers have tried to use artificial intelligence to solve the task of automatic coloring of comics. A Radford et al.[10] proposed that there are few related papers using unsupervised learning CNNs network for image processing, so they proposed a deep convolution network DCGAN to realize CNN's attempts in the field of supervised learning and unsupervised learning respectively. How to accurately cut the image area is also a major factor affecting the accuracy and quality of the final image coloring. Many researchers have done related research on image area segmentation.

In 2014, Goodfelows et al.[11]. proposed the Generative Adversarial Network (GAN). The Generative Adversarial Network is a simple and efficient network that is completely different from the previous deep learning model. The general deep learning model contains an overall neural network, and GAN breaks this. It consists of two sub-networks, the generator sub-network and the discriminator sub-network. The generator can be used to extract the features of the image and generate fake images: the discriminator can be used to identify real and fake images and give a probability to conclude that the image is more likely to be a real image or a fake image. In this process, both the generator model and the discriminator model are constantly retraining. With the increase of the number of iterations, the fake image forgery ability of the generator model is getting stronger and stronger, and the real and fake images of the discriminator can be accurately discriminated. The ability is also getting stronger and stronger, and eventually tends to converge. Therefore, with this training method, GAN models are widely used for Image processing which is one of the most commonly used models in the field of comic coloring. Seo Chang et al.[12] proposed a new GAN model for image region segmentation combined with semantics - the Seg2pix model. This model is an improvement on the pix2pix model, which is used to reproduce the image after semantic region segmentation, so as to provide a reference for artists. Arjovsky M et al.[13] proposed wGAN and its improved method. By changing the loss function and discriminator of the original GAN, the model training is more stable and the quality of the generated images is higher. Radford et al.[10] proposed DCGAN, which combines GAN and CNN, replacing pooling layers with convolution and deconvolution, and adding batch normalization layers to improve the performance and stability of DCGAN. Zhang H et al.[14] proposed StackGAN to generate high-resolution images based on descriptive texts. StackGAN is improved based on cGAN. The algorithm is divided into two stages: first, the first step depicts the basic outline of the image and the relevant color distribution, and the second step is based on the given image. Text and low-resolution images generated in the first stage, correct the defects in the low-resolution images, and finally generate high-resolution images. StackGAN divided the text and image

synthesis process into two steps and successfully created real 256 Œ 256 images. Compared to previous image fusion models built using GAN, our proposed approach can be seen as a variant of conditional GAN, but based on information about both the image and the text.After Zhang H et al. improved StackGAN and proposed a new model StackGAN++[15], which further improved the stability of training and the quality of generated images. In 2014, Mirza M et al.[16] proposed a conditional generative network cGAN, which proved that the model can generate MNIST digits conditioned on class labels. Isola P et al.[17] proposed a general framework pix2pix for image translation tasks based on cGAN. This framework requires two pairs of images from different image domains for training, and has been tested on multiple datasets with good conversion results. In order to solve the problem that the resolution of the images generated by pix2pix is not high enough, Wang et al.[18] proposed the pix2pixHD model, using a multi-scale generator and discriminator. Zhu et al.[19] proposed CycleGAN, an image translation model trained using images from two unaligned image domains, which solves the problem of difficult alignment data acquisition in image translation tasks. Zhang R et al.[20] propose an image colorization system using a convolutional neural network (CNN) that maps a grayscale image along with a sparse user-specified colormap to the output, with colorization in as little as one minute. Kataoka et al.[21] used GAN to automatically colorize black and white comics. The algorithm first used edge detection operators to remove the screen pattern of black and white comics, then extracted color style information from the color image, and colorized the black and white image according to the extracted color style information.

Reed et al.[22] were the first to propose a deep neural structure based on the conditional GAN system, which successfully generated real images (64 Œ 64) from natural language descriptions. The images can be synthesized from a given query image and a textual description, which the paper calls "style transfer". This is the only generative model capable of solving the image synthesis problem described in this paper. They further developed another model[23] that can synthesize 128 Œ 128 images using constraints such as text descriptions, object locations, and other annotations. Dumoulin et al.[24] proposed a model for cooperative learning of a generator and encoder network using adversarial learning. The discriminator jointly performs the discrimination task in both latent space and data space. Mahzani et al.[25] proposed adversarial autoencoders in which the discriminator is trained to discriminate between arbitrary prior and posterior distributions of latent feature representations. The trained encoder is able to match the distributions of the data with the distributions of the candidate representations. Larsen et al[26] directly combined VAE and GAN into one network by decomposing the GAN generator and VAE decoder, replacing the elementary metric of the original VAE with the feature metric determined by the discriminator. AttnGAN[27] develops details using a pre-trained model of visual and semantic similarity. Although it uses visual-semantic attention at the word level, similar to our approach, it relies mainly on sentence vectors to generate

images. All methods generate a variety of high-quality images from text, but their focus is on generating a completely new image rather than manipulating certain parts of the input image with text.

As for the research of text-guided Image manipulation, in 2018, Seonghyeon Nam et al.[28] propose a novel text-conditioned visual attribute manipulation method, called Text-Adaptive Generative Adversarial Networks (TAGAN). The key idea is to split a sentence-level discriminator into multiple word-level discriminators, such that each word-level discriminator is attached to a specific type of visual attribute. In this way, the discriminator is able to provide fine-grained training feedback to the generator to change only specific visual properties. To this end, we introduce a text-adaptive discriminator consisting of word-level local discriminators. It classifies images as true or false by aggregating all word-level matching scores from local discriminators with textual attention. Using a text-adaptive discriminator, our generator learns to alter parts of the image while preserving other content.

1.2.2 Domestic Status for Theoretical Research

There are also many domestic scholars who have carried out research on automatic grayscale image coloring methods. In the past few years, many excellent automatic image coloring papers have been published. Bin Sheng et al.[29] used a rotation-invariant Gabor filter bank and optimized it in the feature space. Pixels of similar texture features provide similar colors. He M et al.[30] proposed the use of convolutional neural network to automatically colorize uncolored images. After giving an uncolored black and white image, the model will be calculated to directly obtain a lustrous color image. This method can learn end-to-end how to select and transfer color information from batch image data. This method is very robust, and the model can learn the style of the image effective even if the color reference image is very different from the gray image style. Xiao C et al.[31] proposed a dense encoding pyramid network, which defined the coloring problem of target gray images as a polynomial classification problem. By analyzing the color distribution of the color reference map, the relevant distribution was used to predict each color on the gray image. This method considers the global and local color information and semantic information of the image at the same time in the coloring process, which can effectively extract the color information in the color reference image and render it to the target grayscale image. Gang Liu et al.[32] proposed an improved manga line draft coloring method based on U-net. The proposed swish layer and swish gated residual blocks (SGRBs) effectively filter the information transmitted by each layer and speed up The convergence speed of the M path can be controlled more effectively.Xiangguo Liang et al.[33] used convolutional neural networks, alternating two loss functions in the model to reduce the training error and using guided filtering to solve the problem of unclean backgrounds, blurred

or clumpy color blocks in images; Wu et al.[34] proposed a distortion function to create a smooth and clear transformation between two images, and for each pigment of the output image, selecting a most appropriate input image for each pigment of the output image, to prevent blurring and ghosting in the generated color image.

L Zhang et al.[35] developed an AI coloring tool, style2paints, to automate the coloring of line drawings to the finished manga. The model training process takes an unsupervised form without adding artificially defined rules, and coloring with style2paints can achieve specific style transformations for automated and fast coloring. Zhang R et al.[36] proposed that previous methods for automatic coloring of comics either required too much precision in scribbling lines during user interaction, or less demanding interactive coloring would lead to unacceptable quality of the colored finished manuscript and could not serve as a substitute for manual coloring, so this paper considers the uncertainty of the coloring problem as a classification task in deep learning and adopts a class rebalancing approach to increase the generated color diversity in the generated manga finished drafts, and the method has passed the graph evaluation chamber and fooled humans in 32 % of the experiments, taking the manga automatic coloring research a big step forward.

As for the research of Text-Guided Image Manipulation, Hao Dong et al.[37] propose an end-to-end neural architecture that utilizes a GAN network to automatically learn an implicit loss function, which should be able to separate semantic information from two modalities (image and text) and generate from the combined semantics new image. It is optimized to match the target text description while being realistic and maintaining other image features that are not related to the text description when given a source image and a target text description. The model addresses the problem of how to manipulate images with semantically different textual descriptions. Their approach trains a conditional GAN to synthesize a given input image and a target text description. However, the effectiveness of both approaches is limited because of the simple sentence modeling discussed above. In this paper, we address this problem in a different way by creating a subtle discriminator to learn how to distinguish between the visual attributes of images and text. Bowen Li et al.[38] proposed a novel generative adversarial network (ManiGAN) that consists of two key components: a text-image affine combination module (ACM) and a detail correction module (DCM). ACM selects image regions relevant to a given text, and then associates these regions with corresponding semantic words for efficient manipulation. At the same time, it encodes raw image features to help reconstruct text-independent content. DCM corrects the mismatched attributes and completes what is missing from the composite image.

1.2.3 Existed Method of Textual Description Based Colorization of Scene Sketches

In 2018 Changqing Zou et al.[39] proposed the first large-scale sketch scene dataset to improve sketch understanding at the feature and scene level. This dataset is created using a well-designed new crowdsourcing pipeline that allows users to efficiently create a large number of realistic and varied scene sketches. Sketch Scene contains over 29,000 scene-level sketches, over 7,000 pairs of scene and photo models, and over 11,000 object sketches. All objects in the scene sketch have the basic truth and semantics of the appropriate mask. The dataset is also highly scalable and scalable, making it easy to improve and / or change the composition of the scene. This database is undoubtedly an important work in sketching understanding.

On the basis of this research, Zou Changqing et al.[40] first proposed a language-based interactive coloring system for scene sketches based on semantic understanding. The system is based on deep neural networks, trained on the large-scale dataset of scene sketches mentioned above and textual descriptions. Given a scene sketch, the system allows the user to interactively locate specific foreground object instances through language-based instructions, satisfying various shading requirements in a progressive manner. In order to reduce the difficulty of model reasoning, the paper decouples the coloring problem into two sub-problems: segmentation and coloring, and colorizes the foreground and background of the entire sketch step by step. Specifically, as shown in Figure 1.1, a novel Reference Instance Segmentation algorithm is first used to automatically segment the foreground objects of interest according to the user's text input, and then in Figure 1.2 a generative adversarial network (GAN) and the model of the image-text feature fusion algorithm color the segmented foreground objects according to the user's language instructions, and this process will automatically perform part segmentation and color management. After completing the coloring of some or all objects, finally the system as shown in Figure 1.3 uses a two-channel adversarial network to simultaneously learn the segmentation and coloring of background areas including blank areas.

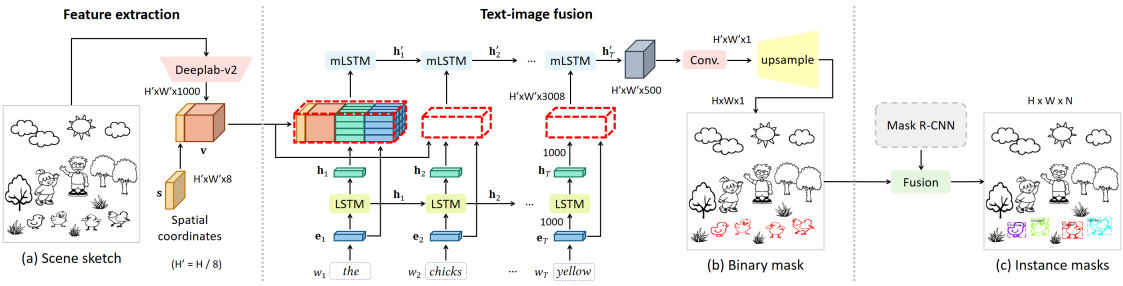

Figure 1.1. Network architecture of the instance matching model[40].

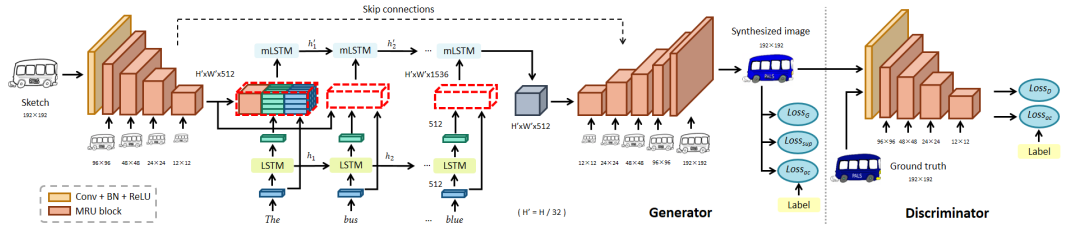

Figure 1.2. Network architecture for foreground colorization [40].

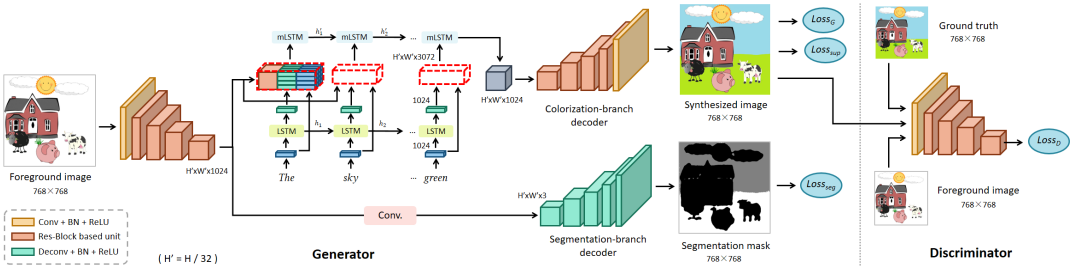

Figure 1.3. Network architecture for background colorization [40].

1.2.4 Limitations of Existing Methods

For sketch coloring, existing methods use deep neural networks to estimate the color information of objects. However, due to the lack of prior knowledge, the color variability of the image is too large. For example, the clothes worn by people can be filled with various colors. While traditional interactive sketch coloring solutions support precise control through a scribble-based interface, they require explicit color selection and direct target selection using color selection tools. In contrast, when text-based descriptions are made through voice input, the system is more easily adopted by novices and is friendly to patients with upper extremity injuries. But at the same time, there are some deficiencies in language-based sketch coloring. In terms of image coloring, there will still be some flaws in coloring, as well as modifying areas that are not specified by natural language. Besides, training the network normally takes a long time. For example,The language-based interactive coloring system proposed by Changqing Zou et al.[40] takes one to two days to train one network.

1.3 Research Target and Main Content

Sketch coloring is a very interesting and practical research direction. Given a grayscale sketch, existing methods use deep neural networks to estimate the color information of objects. However, due to the lack of prior knowledge, the color variability of the image is too large. For example, the clothes worn by people can be filled with various colors. In order to constrain the solution space, this paper studies the coloring of sketches guided by language description, and gradually completes the coloring of the whole image through natural language

interaction. The system proposed by Changqing Zou et al.[40] divides the sketch coloring into three parts: instance matching, foreground coloring, and background coloring, which can color each instance in the sketch separately and improve the accuracy of the language-based sketch coloring system. . Therefore, this graduation project will continue to use the structure of instance matching, foreground coloring, and background coloring in this paper, and use some other methods to improve on the original basis to improve the training speed and improve the coloring results.

1.4 Summary

This chapter introduces the research background of text-based sketch coloring, the development of Chinese manga industry in recent years, and the progress and influence brought by language-based sketch coloring in the manga industry in the context of the current situation of Japanese manga and the reason why Japanese manga is mainly black and white. By automatic coloring, the workload of manga workers can be significantly reduced, and more color manga can be produced as a result.

After that, some domestic and international methods and papers related to comics coloring, as well as existing text-based sketch coloring systems are introduced, and the problems in existing text-based sketch coloring systems are presented. Finally, the main research content and objectives of this paper are described.

2 Preliminary

2.1 Text Encoding Method

2.1.1 Bidirectional Encoder Representation from Transformers

In 2018 Jacob Devlin et al.[41] proposed Bidirectional Encoder Representation from Transformers which is a pre-trained language representation model. It emphasizes that the traditional one-way language model or the method of shallow splicing two one-way language models for pre-training is no longer used as before, but a new masked language model (MLM) is used to generate deep Bidirectional Linguistic Representation. As shown in Figure 2.1 Bidirectional Encoder Representation from Transformers(BERT) employs MLM to pre-train bidirectional Transformers[42] to generate deep bidirectional language representations. After pre-training, it is only necessary to add an additional output layer for fine-tune to achieve state-of-the-art performance in a variety of downstream tasks. No task-specific structural modifications to BERT are required in this process.

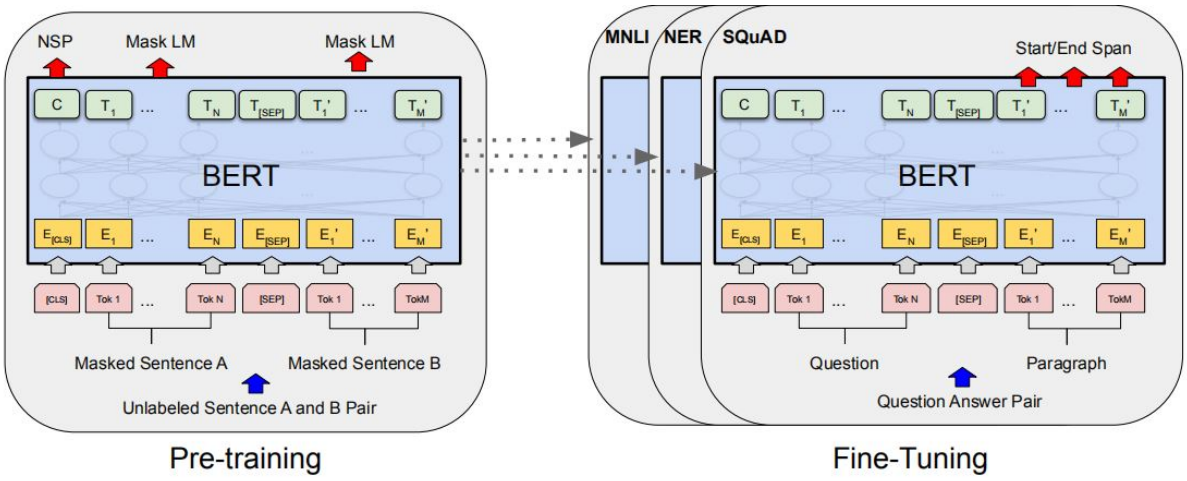

Figure 2.1. Overall pre-training and fine-tuning procedures for BERT[41]. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token.

As shown in the Figure 2.2, the input of BERT is the representation corresponding to each token and the representation is composed of three parts, which are the corresponding token embeddings, segment embeddings and position embeddings, and the word dictionary is constructed using the WordPiece algorithm. In order to complete the specific classification task, in addition to the token of the word, the author also inserts a specific classification token ([CLS]) at the beginning of each input sequence, and the last Transformer layer output cor-

responding to the classification token is used to play the role of The role of aggregating the entire sequence characterization information.

Since BERT is a pre-training model, it must be adapted to a variety of natural language tasks, so the sequence input to the model must be capable of containing one sentence (text sentiment classification, sequence labeling tasks) or more than two sentences (text summarization, natural language inference and question answering tasks). BERT uses two methods to distinguish which range belongs to sentence A and which range belongs to sentence B:

- Insert a segmentation token ([SEP]) after each sentence in the sequence tokens to separate different sentence tokens.
- Add a learnable segmentation embedding to each token representation to indicate whether it belongs to sentence A or sentence B.

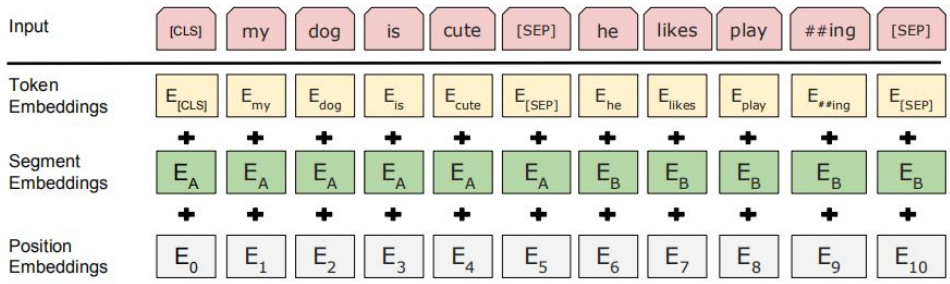

Figure 2.2. BERT input representation[41].

As for the output of BERT, as shown in the Figure 2.1, because the characteristic of Transformer is that there are as many inputs as there are corresponding outputs, the outputs and inputs are also one-to-one correspondence. C is the classification token ([CLS]) corresponding to the output of the last Transformer, and T_i represents the output of other tokens corresponding to the last Transformer. For some token-level tasks (such as sequence labeling and question answering tasks), the T_i is input into an additional output layer for prediction. For some sentence-level tasks (such as natural language inference and sentiment classification tasks), C is input into an additional output layer, which explains why a specific classification token is inserted before each token sequence.

Although the field of NLP does not have high-quality human-annotated data like ImageNet, the self-supervised nature of large-scale text data can be used to construct pre-training tasks. Therefore, BERT builds two pre-training tasks, Masked Language Model and Next Sentence Prediction.

Masked Language Model is the reason why BERT is not limited by one-way language models. Simply put, it randomly replaces the token in each training sequence with a mask token ([MASK]) with a probability of 15%, and then predicts the original word at the [MASK] position. However, since [MASK] does not appear in the fine-tuning stage of the downstream task, there is a mismatch between the pre-training stage and the fine-tuning stage (it is well explained here that the pre-training target will make the Linguistic representations are sensitive to [MASK], but not to other tokens). Therefore, BERT adopts the following strategy to solve this problem: first, a certain token position is randomly selected for prediction with a probability of 15% in each training sequence. If the i-th token is selected, it will be replaced by the following three one of the tokens

- 80% probability of [MASK]. For example, my dog is hairy——>my dog is [MASK]
- 10% probability of a random other token. For example, my dog is hairy——>my dog is apple
- 10% probability of the original token. For example, my dog is hairy——>my dog is hairy

Then use the T_i corresponding to the position to predict the original token (input to the full connection, then use softmax to output the probability of each token, and finally use cross entropy to calculate loss). This strategy makes BERT no longer only sensitive to [MASK], but to all tokens, so that the representation information of any token can be extracted.

Some tasks such as question answering and natural language inference need to understand the relationship between two sentences, while MLM tasks tend to extract token-level representations, so sentence-level representations cannot be directly obtained. In order for the model to be able to understand the relationship between sentences, BERT uses the NSP task for pre-training, which is simply to predict whether two sentences are connected together. The specific method is: for each training example, we select sentence A and sentence B in the corpus to form, 50% of the time sentence B is the next sentence of sentence A (marked as IsNext), and the remaining 50% of the time Sentence B is a random sentence in the corpus (labeled NotNext). Next, input the training example into the BERT model, and use the C information corresponding to [CLS] to make two-class predictions.

2.1.2 Long Short Term Memory

lstm has the ability to memorize long and short term information. LSTM was first proposed by Hochreiter et al.[43] in 1997. Due to the rise of deep learning in 2012, LSTM has undergone several generations of development, thus forming a relatively systematic and complete LSTM framework, and obtained in many fields. a wide range of applications.

The motivation proposed by LSTM is to solve the long-term dependency problem. That is, after the nodes of the neural network have undergone many stages of computation, the features of the previous relatively long time slice have been covered. Traditional RNN node outputs are determined only by weights, biases, and activation functions. RNN is a chain structure, each time slice uses the same parameters.

The reason why LSTM can solve the long-term dependence problem of RNN is that LSTM introduces a gate mechanism to control the circulation and loss of features. LSTM is composed of a series of LSTM units. Its chain structure is as shown in the Figure 2.3 and the meaning of the symbol is shown in Figure 2.4.

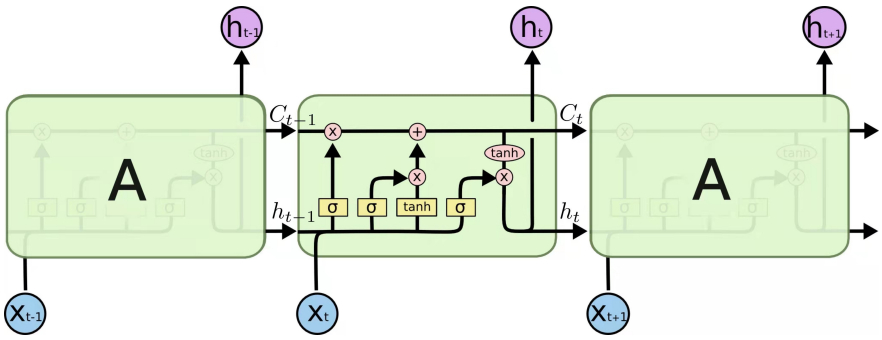

Figure 2.3. LSTM Unit[43].

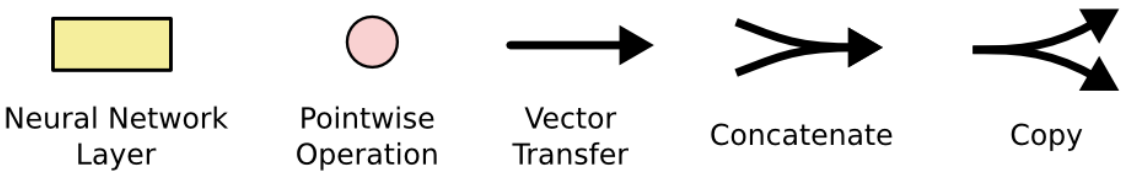

Figure 2.4. Symbolic meaning of LSTM[43].

There are three inputs: Cell state C_{t-1} , the hidden layer state h_{t-1} and the input vector X_t at time T. The outputs including cell state C_t and the hidden layer state h_t which is the output at time t. The calculation process of LSTM can be summarized as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{2.1}$$

As shown in formula 2.1 the f_t is called the forget gate, it indicates which features of C_{t-1} are used to calculate C_t. f_t is a vector where each element of the vector is within the range of [0, 1]. Usually we use sigmoid as the activation function, the output of sigmoid is a value in the interval of [0, 1], the gate values are overwhelmingly very close 0 or 1, and the rest are few and far between.

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \tag{2.2}$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \tag{2.3}$$

As shown in the formula 2.2 and formula 2.3 , $\tilde{C}_t$ represents the unit state update value, which is obtained from the input data X_t and hidden nodes h_{t-1} through a neural network layer, and the activation function of the unit state update value usually uses tanh . i_t is called the input gate. Like f_t, it is also a vector whose elements are within the range of [0, 1]. It is also calculated by X_t and h_{t-1} through the sigmoid activation function.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.4}$$

In formula 2.4 i_t is used to control which features of $\tilde{C}_t$ are used to update C_t , in the same way as f_t.

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \tag{2.5}$$

$$h_t = o_t * tanh(C_t) \tag{2.6}$$

Finally, h_t is obtained by calculation as the predicted value and the input of the next layer.

2.1.3 Gate Recurrent Unit

Gate Recurrent Unit(GRU) is proposed by Cho et al[44] which is a type of Recurrent Neural Network (RNN). Like LSTM (Long-Short Term Memory), it is also proposed to solve problems such as long-term memory and gradients in backpropagation. Compared with LSTM, using GRU can achieve comparable results, and it is easier to train and can greatly improve training efficiency, so GRU is often preferred. Compared with LSTM, GRU has one less "gate" and fewer parameters than LSTM, but it can also achieve the same function as LSTM. The architecture of Gate Recurrent Unit is shown in Figure 2.5

The input of GRU contains a current input x_t and the hidden state h_{t-1} passed down by the previous node, which contains the relevant information of the previous node. Combining x_t and h_{t-1}, GRU will get the output of the current hidden node h_t and the hidden state h_t passed to the next node. The calculation process of LSTM can be summarized as:

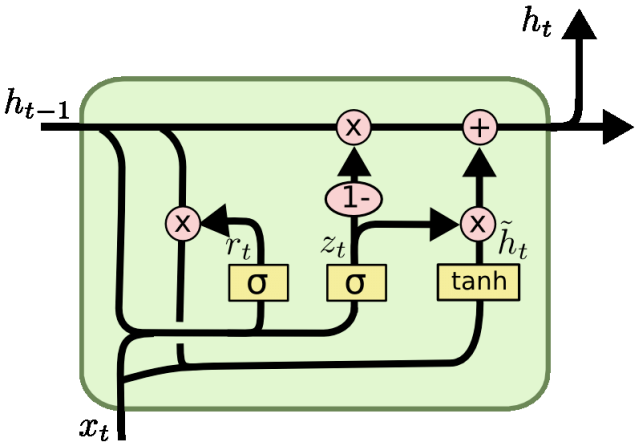

Figure 2.5. GRU unit[44].

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{2.7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{2.8}$$

As shown in formula 2.7 and formula 2.8, two gated states are obtained through the previous transmitted state h_{t-1} and the input x_t of the current node. r_t controls the reset gate, and z_t controls the update gate.

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{2.9}$$

As shown in formula 2.9, The $\tilde{h}_t$ here mainly contains the currently input x_t data. Adding $\tilde{h}_t$ to the current hidden state in a targeted manner is equivalent to "memorizing the state of the current moment".

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{2.10}$$

Finally, as shown in the formula, to update the memory, GRU uses the same gate z_t to forget and select memory at the same time (LSTM uses multiple gates). Compared with LSTM, GRU has one less "gate" and fewer parameters than LSTM, but it can also achieve the same function as LSTM. Considering the computing power and time cost of hardware, more "practical" GRUs are often chosen.

2.2 Summary

This chapter mainly introduces several text encoding methods that will be used and compared in this paper, and introduces the characteristics, basic structure and calculation process of these methods in detail.

3 Network Architecture

In the Language-based Colorization of Scene Sketches proposed by zou et al.[40], the sketch coloring system is divided into three parts: instance matching, foreground coloring and background coloring. Each instance in the sketch can be colored separately, which improves the language-based sketch coloring system accuracy. Therefore, in this paper, improvements will be made on the basis of this structure.Among the three parts of the colouring system: instance matching, foreground colouring, and background colouring, the instance matching and background colouring results are better, while the colouring results of the foreground colouring part still have many shortcomings, and all three parts have the disadvantage of long training time. Therefore, this chapter will focus on improving the network structure of foreground shading and background in the shading part to improve the shading results of foreground shading and reduce the time consumed for training.

3.1 Foreground Colorization

As shown in the Figure 3.1 the foreground object instance coloring network is essentially a generative adversarial network (GAN) consisting of a generator G and a discriminator d. Different from traditional generators, only single-modal visual image data is accepted as input, and foreground object instances are colored The generator G of the network needs to model the interaction between the textual description and visual information, since the generated colorization results should be constrained by the textual information.

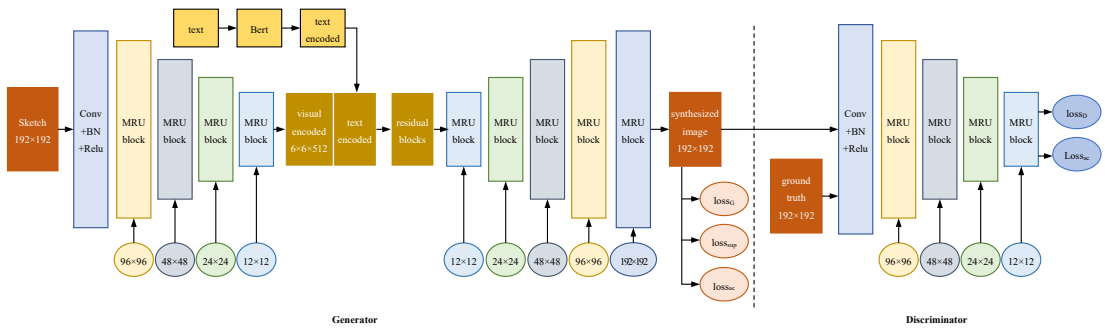

Figure 3.1. The architecture of foreground colorization.

The generator G consists of three modules: an image encoder, which encodes the fea-

tures of the image sketches of the H*W segmented object instances generated in the instance matching phase, and a fusion module, which combines the textual information of the natural language expressions with the features generated by the image encoder. The convolution layer consists of a convolution, a batch normalization and a Relu activation function. In mathematics, convolution is an operation. $(f*g)(n)$ is called the convolution of f and g whose continuum is defined as

$$(f*g)(n) = \int_{-\infty}^{\infty} f(\tau)g(n-\tau)\mathrm{d}\tau \tag{3.1}$$

whose discrete definition is

$$(f*g)(n) = \sum_{\tau=-\infty}^{\infty} f(\tau)g(n-\tau)\mathrm{d}\tau \tag{3.2}$$

For two functions $f(x)$ and $g(x)$, the convolution operation $(f*g)(n)$ is the summation (continuous for integration, discrete for summation) of such a product of $f(x)g(y)$, where x and y need to satisfy the condition that they can only fall on a line in a two-dimensional coordinate system shaped as $y = n - x$, which is a line with slope -1 and intersection with both the y and x axes at n. In a convolutional neural network, the convolutional operation is performed on two matrices, the input matrix and the filter (also called the convolutional kernel). The convolutional kernel moves over the input matrix in certain steps and performs a dot product operation to obtain the output matrix on the right-hand side. In this thesis the convolution layer uses a convolution kernel of size 7 and a step size of 2. When using a convolutional neural network for image recognition, the input is transformed image data, an image with a width of w, a height of h and a depth of d, denoted as h*w*d. In this paper it is a black and white sketch of 192*192 and the depth is the number of bits used to store each pixel of the image, for example a colour image, which has three components of RGB in one pixel, has a depth of 3.

The BatchNorm was proposed by Ioffe et al.[45] in 2015, and the role of the BatchNorm is to keep the inputs to each layer of the neural network equally distributed during the training of the deep neural network. If the input image is whitened in image processing - by whitening, we mean transforming the input data distribution to a normal distribution with zero mean and unit variance - then the neural network will converge faster, and the BN authors Then the BN authors start to reason: the image is the input layer of the deep neural network, and doing whitening will speed up convergence, so in fact for a deep network, the neurons in one of the hidden layers are the input to the next layer, meaning that in fact every hidden layer of the deep neural network is an input layer, but only relative to the next layer, so the original

idea that inspired BN is to do whitening for every hidden layer. Therefore BN is a simplified version of whitening the activation values of each hidden layer neuron of a deep neural network. Because the activation input values of the deep neural network before the nonlinear transformation gradually shift or change in distribution as the network gets deeper or during training, the reason for the slow convergence is generally that the overall distribution gradually moves closer to the upper and lower limits of the value interval of the nonlinear function (for the Sigmoid function, this means that the activation input values WU+B is a large negative or positive value), so this leads to the disappearance of the gradient of the lower layer of the neural network when back propagation, which is the essence of the slow convergence of training deep neural networks, and BN is to force the distribution of the input value of any neuron in each layer of the neural network back to the standard normal distribution with a mean of 0 and a variance of 1 through certain normalization means, which actually means forcing the increasingly skewed distribution back to a more standard distribution. This makes the activation input values fall in the region where the non-linear function is more sensitive to the input, so that small changes in the input will lead to larger changes in the loss function, meaning that this makes the gradient larger and avoids the problem of gradient disappearance, and a larger gradient means faster learning convergence, which can greatly accelerate the training speed. For each hidden layer neuron, the input distribution, which is gradually mapped to the nonlinear function towards the limit saturation zone of the value range, is forced back to a more standard normal distribution with a mean of 0 and a variance of 1, so that the input values of the nonlinear transformation function fall into a more sensitive region to the input, thus avoiding the gradient disappearance problem. Because the gradient is always kept relatively large, it is obvious that it is more efficient to adjust the parameters of the neural network, that is, it is more variable, that is to say, it is a large step towards the optimal value of the loss function, that is to say, it converges quickly. But if it all passes BN, then it has the same effect as replacing a non-linear function with a linear one. This means that the expressiveness of the network is reduced, which means that the depth is no longer meaningful. So in order to ensure that the non-linearity is obtained, the transformed x that satisfies a mean of 0 with a variance of 1 is then subjected to a scale plus shift operation ($y = scale * x + shift$), with two additional parameters for each neuron, the scale and shift parameters, which are learned through training, meaning that the value is shifted by scale and shift from This is equivalent to moving the value of the non-linear function from the linear region around the positive centre to the non-linear region. The BatchNorm is calculated as shown in the table

An image feature map and finally an image decoder that uses the fused features and

Algorithm 1: Calculation process of BatchNorm.

Input: value of x over a mini-batch: $\beta = \{x \ldots_m\}$; Parameters to be learned: γ , β

Output: $\{y_i = BN_{\gamma,\beta}(x_i)\}$

1 $\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$; `// mini-batch mean`

2 $\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$; `// mini-batch varience`

3 $\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$; `// normalize`

4 $y_i \leftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$; `// scale and shift`

produces a H*W*3 result. The MRU block will continue to be used as the backbone of the encoding and decoding modules in the foreground shading network. The MRU block, first proposed in 2018 by Chen et al.[46] accepts additional input images and generates new feature maps by dynamically incorporating the information from the additional images into the original input feature maps. The architecture of MRU is shown in Figure 3.2. The results show that the MRU block outperforms the naive convolution method on sketch data. A convolutional layer and four cascaded MRU are used in the foreground colouring system to encode the HŒW input object sketch image into a 6*6 feature map in the encoder. The MRU module allows a ConvNet to conditionally iterate over an input image. The module uses a learned internal mask to selectively extract new features from the input image and combine them with the feature maps computed by the network so far. The MRU block has two inputs:input feature map x_i and image I and output feature map y_i. $Conv(x)$ is the convolution on x and $f(x)$ is the activation function. We first have to merge the information in the input image I into the input feature mapping x_i. To enable the mru block to decide how much information it wishes to retain when a new image is received, the following method is used:

$$m_i = \sigma(Conv([x_i, I])) \tag{3.3}$$

$$z_i = f\left(\mathrm{Conv}\left([m_i \odot x_i, I]\right)\right) \tag{3.4}$$

m_i is the mask on the input feature map. Multiple convolutional layers can be stacked here to improve performance. We then wish to dynamically combine the information from the newly convolved feature map z_i with the original input feature map x_i, so we use another mask:

$$n_i = \sigma\left(\mathrm{Conv}\left([x_i, I]\right)\right) \tag{3.5}$$

$$y_i = (1 - n_i) \odot z_i + n_i \odot x_i \tag{3.6}$$

The second term in the Formula 3.6 is the residual connection. Since there is an internal mask that determines the flow of information, we call this structure a masked residual unit. We can

stack multiple such units, repeating the same image at different scales of input, so that the network can dynamically retrieve information from the input image along its computational path.The formulation of the MRU is similar to that of the recursive unit (GRU)[44]. However, the motivation of MRU is different and there are several important differences:

- GRU is designed to change the escape gradient in a recursive neural network. The motivation of MRU is to repeat the same input image to improve information flow.
- GRU blocks are periodic, so part of the output is redirected to the same block, while MRU blocks are cascaded, so the output of the previous block is redirected to the next block.
- The MRU shares the weights at each step, so it can only receive input of a fixed length. Two MRU blocks do not share weights, so it is possible to reduce or increase the size of the output feature map as in a conventional convolutional layer.

For the decoder, the 5 MRU are cascaded into a deconvolutional network. A skip connection was applied between the encoder and the decoder. The text encoding part is different from the structure proposed by Zou et al.[40] (LSTM) employing bert encoding. In the foreground colouring part, the input text information is usually short, e.g. the bus is yellow with blue window, and therefore does not need to be memorised for a long time using LSTM. Compared to Bert, the use of LSTM is slower and Bert does a better job of pre-training the model with a large amount of unsupervised data by designing a pre-training task. In the text image information fusion section, the same method as in the paper Text-Adaptive Generative Adversarial Networks: Manipulating Images with Natural Language[28], i.e. residual blocks[47], is used for the fusion of text image information. The architecture of residual block is shown in Figure 3.3. By using the residual edges in the residual blocks, the fusion of textual information can be improved more consistently. In the absence of residual edges, if the network is very deep, in order to update the weights of the bottom layer (near the input data), the gradient has to be multiplied forward according to the chain rule, and if any of the factors is too small, the gradient will be very small, and this small gradient will not help even if it is multiplied by a large learning rate. When there is a residual edge, the gradient can be found directly through the "highway" to the object we want to ask for the gradient, no matter how small the gradient is by the normal route through the chain rule, the result of the two routes will not be small, so the gradient can be updated very effectively. Therefore in this paper, five layers of residual blocks are used to better fuse the stitched text image information.

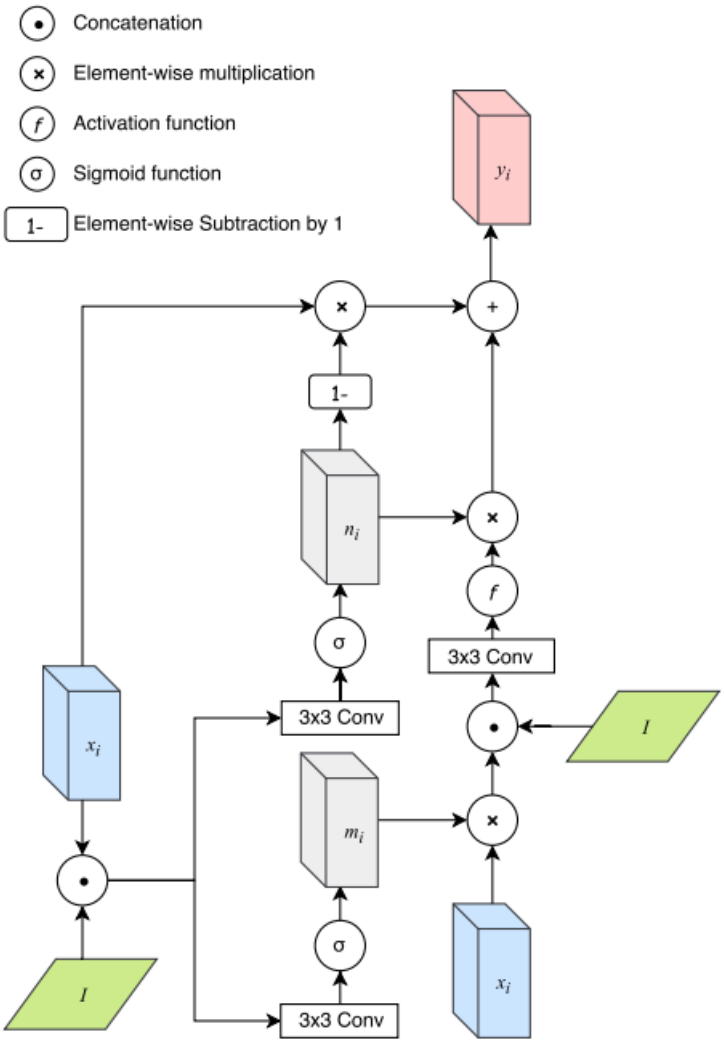

Figure 3.2. The architecture of MRU[46].

The structure of the discriminator is the same as that in the paper presented by Zou et al.[40]. Discriminator D takes a composite or real colour image as input and determines whether it is fake or real. d uses MRU blocks to form a discriminator. d outputs two logarithms: the GAN loss and the classification loss. The target function of GAN is similar to that of SketchyGAN[46] and can be expressed as $L_{GAN}(D,G)$. The exact $L_{GAN}(D,G)$ procedure is shown in Formula 3.7. x and y are images or sketches, z is the noise vector, and c is the class label.

$$\begin{aligned} L_{GAN}(D,G) = & \mathbb{E}_{y\sim P_{\text{image}}}[\log D(y)] + \\ & \mathbb{E}_{x\sim P_{\text{sketch}},z\sim P_z}[\log(1-D(G(x,z)))] \end{aligned} \tag{3.7}$$

As in SketchyGAN, auxiliary classification losses$L_{ac}(D)$ are introduced to classify D in order to maximise the log-likelihood of predictions and true labels. The exact algorithm of the

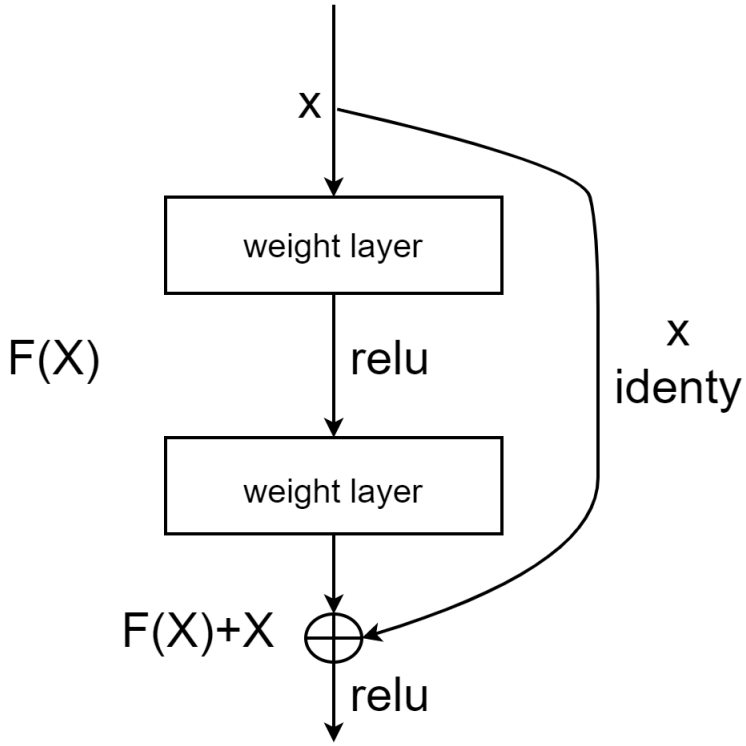

Figure 3.3. The architecture of residual block.

formula is shown in Formula .

$$L_{ac}(D) = \mathbb{E}[\log P(C = c \mid y)] \tag{3.8}$$

The generator maximises log-likelihood because $L_{ac}(G) = L_{ac}(D)$ and the discriminator is fixed, and the image is classified as synthetic. Unlike SketchyGAN, the goal in this paper is to generate colouring results according to user instructions, not to produce different results. Therefore, foreground colorization network do not use loss of perception or loss of diversity and replace the direct L1-distance in SketchyGAN with a smooth L1-distance as a controlled loss, as the colour term in our study may not correspond to a unique RGB value. The penalty should be less than a simple subtraction. The normative loss is defined as follows:

$$L_{sup}(G) = \begin{cases} \frac{1}{2}(G(x,s) - y)^2, & \|G(x,s) - y\|_1 < 1 \\ \|G(x,s) - y\|_1 - \frac{1}{2}, & \|G(x,s) - y\|_1 \geq 1 \end{cases} \tag{3.9}$$

where x is the drawing image of the input case, y is the cartoon image of the ground truth and s is the input text. $L_{sup}(G)$ is estimated at each pixel and summed to estimate the loss of the entire image. The foreground event shading loss function for discriminator D and generator G is defined as follows:

$$L(D) = L_{GAN}(D,G) + \lambda_1 L_{ac}(D) \tag{3.10}$$

$$L(G) = L_{GAN}(G) - \lambda_1 L_{ac}(G) + \lambda_2 L_{sup}(G) \tag{3.11}$$

in which λ_1 and λ_2 are coefficients. The discriminator D tries to maximise$L(D)$ and the generator G tries to minimise $L(G)$.

3.2 Background Colorization

The architecture of background colorization network is shown in Figure 3.4 background colorization network is a conditional GAN (cGAN)[17] based on the foreground image. It fills the background area of the input foreground image and generates a high resolution 768 x 768 colour image. The G-generator consists of a colour branch and a segmentation branch. A discriminator D is added to the colour branch, whose structure repeats the general structure of the cGAN[17] discriminators with slight adaptation. The whole network is trained from top to bottom. The colouring branch of the G generator has a similar structure to the foreground colouring grid. It shares the image coder with the segmentation branch and becomes the coder-decoder structure. The image coder of the G generator uses the remaining block of blocks as its backbone. The residual block was chosen instead of the MRU block used for foreground shading because it has a greater ability to produce large and graded textures than the MRU block and is therefore more suitable for background shading. The first block of the image coder uses a generic convolution layer followed by four cascaded residual blocks with corresponding numbered residual blocks 3, 4, 6 and 3.The text is encoded and the text-image information is fused using the same structure as the foreground colouring, i.e. the text is encoded using Bert, the text and image information is stitched together and the text and image information is further fused using a residual block.

The segmentation results are supplemented with explicit segmentation branches to obtain more accurate segmentation boundaries. The decoder that produces the segmented labelling branches consists of a generic chain of deconvolution levels. The D-discriminator takes the already shaded foreground image as a conditional input and decides whether the synthetic image or the base image is false. It uses a Res-Block based image encoder, similar to the one used in the G generator.

The form of the objective function cGAN is similar to the function proposed by Isola et al.[17] and is denoted as $L_{cGAN}(D,G)$:

$$\begin{aligned} L_{cGAN}(G,D) = & \mathbb{E}_{x,y}[\log D(x,y)] + \\ & \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))] \end{aligned} \tag{3.12}$$

The generator G is trained with three losses:$L_{cGAN}(G)$, $L_{seg}(G)$ and $L_{L1-sup}(G)$. $L_{cGAN}(G)$

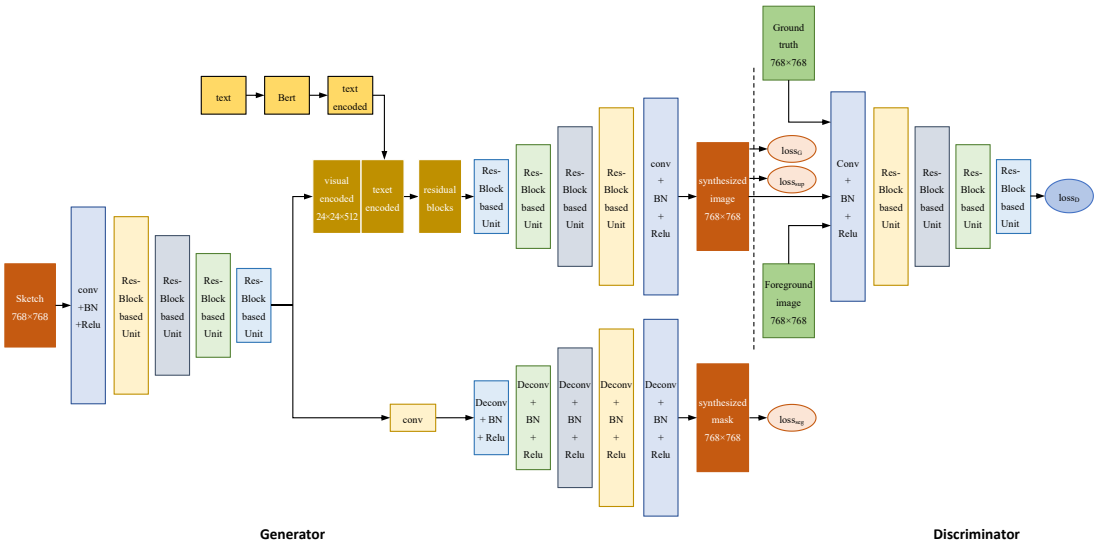

Figure 3.4. The architecture of background colorization.

is the conditional GAN loss, $L_{seg}(G)$ is the cross entropy segmentation loss and $L_{L1-sup}(G)$ controls the network with L1-distance being the distance between the generated image and the real image.

The Cross-entropy loss $L_{seg}(G)$ is essentially a log-likelihood function that can be used in binary and multiclassification tasks. Here, the cross-entropy loss function is used in the multi-classification task. The cross-entropy loss function can perfectly solve the problem of slow update of the square loss function weights, with the good property of "when the error is large, the weights are updated quickly; when the error is small, the weights are updated slowly".The cross-entropy loss function is specifically calculated as follows.

$$L_{seg}(G) = -\frac{1}{n}\sum_{i=1}^{n} y_i log a_i \tag{3.13}$$

where y_i denotes the pixels of the real mask image, a_i denotes the pixels of the generated mask image, and n is the total number of pixels.

The background coloring model fills the background area with color, leaving the foreground object unchanged. Therefore, $L_{L1-sup}(G)$ does not take into account the penalisation of the foreground object area. Therefore, the foreground object's contribution to the loss of oversight must be ignored. The binary mask M_{ij} is used to separate foreground and background areas ($M_{ij} = 1$ means background pixels, otherwise foreground pixels). If there are pairs of image data, i.e. foreground image x and true image y, and a natural language description s, the supervision loss is formulated as follows.

$$L_{L1-sup}(G) = \frac{1}{N}\sum_{i=1}^{W}\sum_{j=1}^{H}\left(M_{ij} * \left\|G(x,s)^{ij} - y^{ij}\right\|_1\right) \tag{3.14}$$

where N is the amount of pixels in the background of the input image.

$$L(D) = L_{cGAN}(D, G) \tag{3.15}$$

$$L(G) = L_{cGAN}(G) + \lambda_1 L_{seg}(G) + \lambda_2 L_{L1-sup}(G) \tag{3.16}$$

where λ_1 and λ_2 is the factor.

3.3 Summary

This chapter focuses on the structure of the improved foreground colouring module. Improvements have been made mainly in the text encoding part and the text image information fusion part. The original structure of fusing word by word with image information has been changed to one in which the whole sentence is encoded by Bert, stitched together with the image information and then fused with the text and image information by means of four residual blocks.

4 Experiment

In the experimental section the focus will be on comparing and experimenting with the foreground colouring section as the original structure has better results for the instance matching and background colouring sections.

4.1 Dataset

4.1.1 Foreground Colorization Dateset

The dataset will use the foreground colour data presented in the work published by Zou et al.[39] and will include colour objects, corresponding edge maps (thumbnails) and a linguistic description of the colour information of these colour objects. To collect this data, Zou et al. examined example cartoon images online and then used x-DoG[48] to extract an edge map as a corresponding thumbnail for each cartoon image. Next, colour annotations were manually created for each cartoon image and then colour descriptions were automatically created for each cartoon image based on a predefined sentence structure template. A total of 4587 cartoon images, borders and language descriptions were then created covering the same 24 object categories as the data, e.g. cars, butterflies, etc. The data were divided into two groups: 3822 groups for training and 765 groups for validation. In addition, to evaluate the effectiveness of our example colouring model on actual sketches rather than on board sketches, we created a test set of 1734 pairs of example sketches and descriptions. These example thumbnails, covering all 24 object classes, were extracted from the scene thumbnails of the test set. All cartoon and border/sketch images are scaled to 192 x 192. These descriptions cover 16 different colours. The number of colours in each object instance varied from 1 to 3. The number of words in each description ranged from 4 to 15, with an average of 6. Location information was added randomly to each collected description, according to a predefined sentence structure. This dataset is further referred to as the foreground dataset. The information about the dataset is shown in Table 4.1 and some representative samples are shown in the following Figure 4.1

Table 4.1. Parameters of the foreground colorization dataset.

Parameter name	Parameter values
amount	4587
number of classes	24
picture resolution	192*192
image format	png
number of channels	3

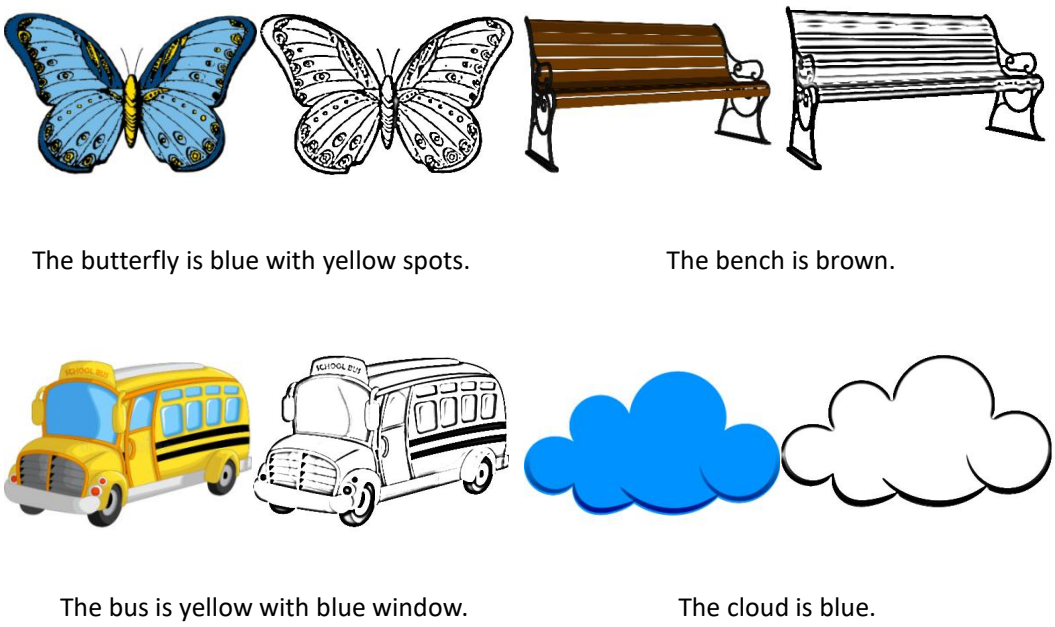

Figure 4.1. Representative samples of foreground colorization dataset.

4.1.2 Background Colorization Dataset

Four types of modal data are required to paint the background: a foreground image (with a blank background), a color image (with a color background), a text description, and a segmented marker map. Based on the object location information provided by the sketch template from the sketch scene dataset, the colored object instances were placed in 768 x 768 white images, thus forming foreground object images. The color images were then created by 24 users (manually drawing the background areas in solid color). For simplicity, only two types of backgrounds were considered in the dataset: sky and ground. By examining the different colors in the background area, background area segmentation information and the corresponding textual color description were extracted. For each foreground image, the data was augmented

by changing the color of the background area in the user picture to produce three additional pairs of color images and descriptions. A representative thumbnail template, the corresponding foreground image, a color image with a user-drawn background, a segmentation map, and three pairs of added color images and text captions are shown in the Figure 4.2. It is a manually hand-drawn background image.

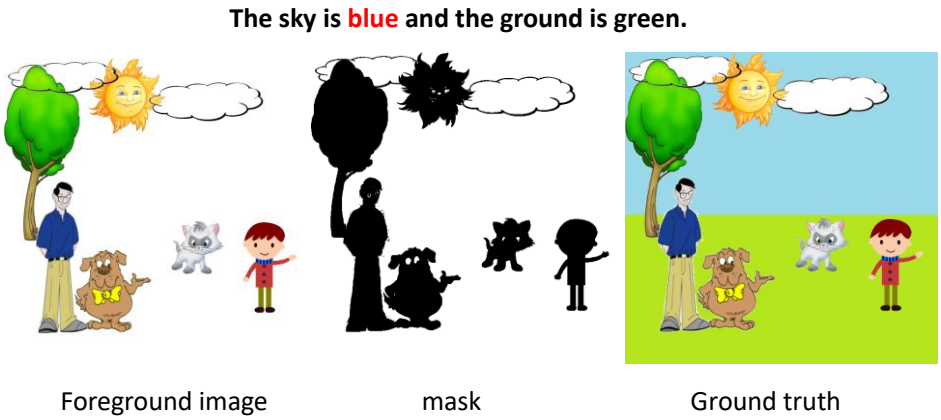

Figure 4.2. Representative samples of background colorization dataset made by hand.

As shown in the Figure 4.3, there are three sets of images generated by changing the colors of the sky and ground. The first line of the image replaces the sky and ground colors of the original hand-drawn image with purple and black. The second row of images replaces the sky and ground colors of the original hand-drawn image with pink and gray. The third row replaces the sky and ground colors of the original hand-drawn picture with bright green and yellow.These additions yielded a total of 15728, 1200, and 2908 sets for training, validation, and testing, respectively. The text descriptions covered a total of 11 colors. The average number of words in each description is 9. This dataset is further referred to as the background dataset. The specific background dataset is shown in the table 4.2.

Table 4.2. Parameters of the background colorization dataset.

Parameter name	Parameter values
amount	19908
Coloring object	sky and ground
picture resolution	768*768
image format	png
number of channels	3

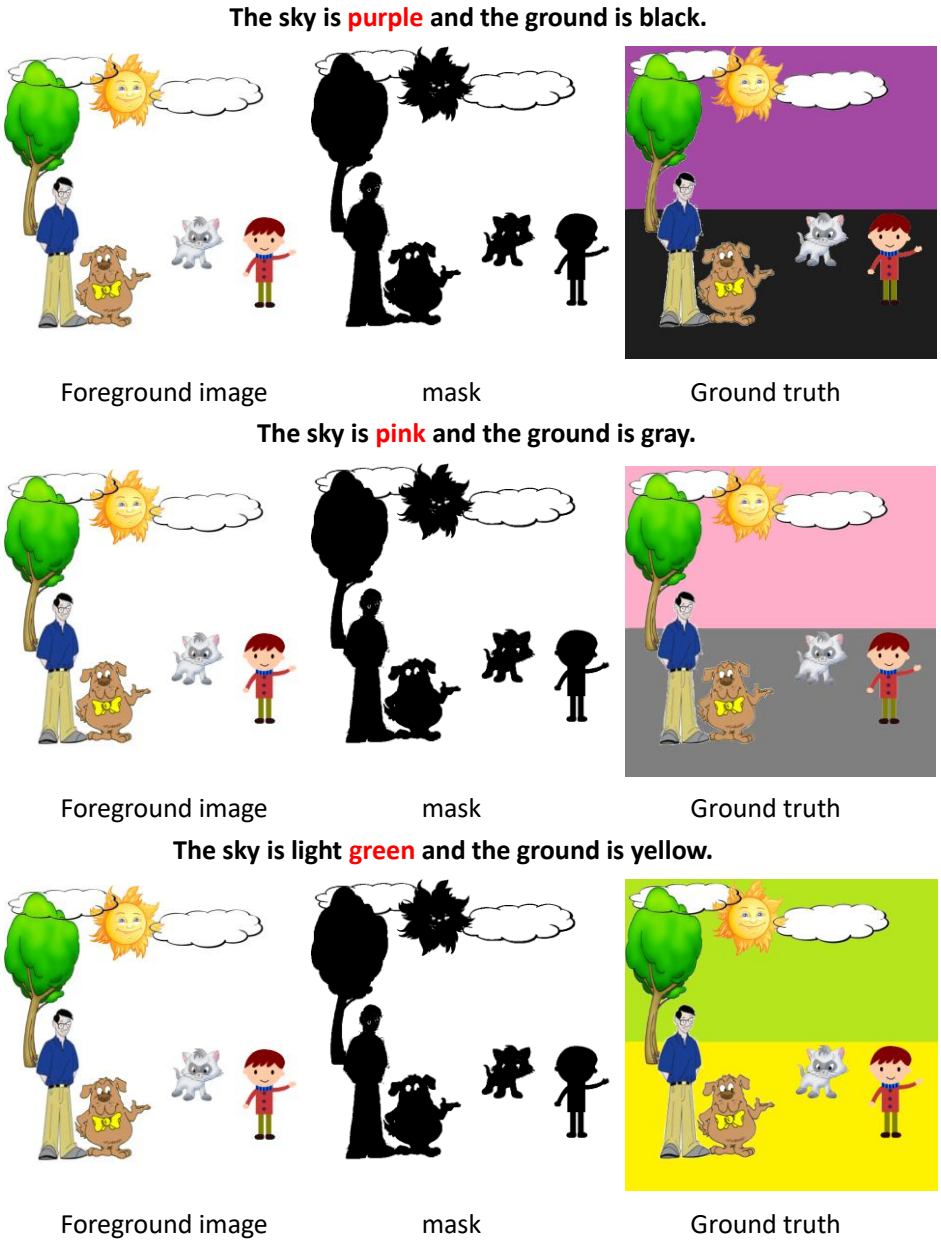

Figure 4.3. Representative samples of background colorization dataset made by replacing the color.

4.2 Experimental Equipment

Google Colab is a free cloud service and supports free GPUs. colab is developed on top of Jupyter. With Colab, you can use Jupyter notebooks without having to download, install or run any software, and you can share Colab notebooks by clicking the "Share" button in the top right corner of any Colab notebook. To use it, install colab on Google Drive and it will work. If you need to upload data, you can upload it to Google Drive, mount it in colab, and access

it directly. You can use either a p100 or v100 graphics card with 16g of memory. p100 and v100 values and power are shown in the Table 4.3.

Table 4.3. Parameters of the GPU.

Parameter name	Tesla V100	Tesla P100
Architecture	Volta	Pascal
Cores / GPU	5120	3584
GPU Boost Clock	1530 MHz	1480 MHz
Tensor Cores / GPU	640	NA
Memory type	HBM2	HBM2
Maximum RAM amount	32 GB	16 GB
Memory clock speed	1758 MHz	1430 MHz
Memory bandwidth	900.1 GB / s	720.9 GB / s
CUDA Support	From 7.0 Version	From 6.0 Version
Floating-point performance	14,029 gflops	10,609 gflops

4.3 Training Parameters

4.3.1 Training Parameters of Foreground colorization

The Foreground colouring training parameters are shown in Table 4.4. The training iteration of the foreground coloring part is 100000. the learning rate of the generator for the original structure and the improved network using Bert for text encoding and the residual block for text-image fusion is 0.0002, while the learning rate of the generator for the network using GRU instead of the LSTM in the original structure is 0.00002. in the discriminator In the discriminator section, the learning rate of all foreground coloring networks is 0.0001. The foreground coloring batchsize is 2.

Table 4.4. Foreground colouring training parameters.

Parameter name	Parameter values
iter	100000
batchsize	2
learning rate(G)	0.0002/0.00002
learning rate(D)	0.0001

4.3.2 Training Parameters of Background colorization

The training parameters of each network in the background coloring part are shown in the table. The learning rate of both the generator and discriminator of the background coloring part of the network is 0.0002. The batchsize of the original and the modified network using Bert for text encoding and residual block for text image information fusion is 1. The training iteration is 80000.

Table 4.5. Background colouring training parameters.

Parameter name	Parameter values
iter	80000
batchsize	1
learning rate(G)	0.0002

4.4 Experimental Results and Analysis

4.4.1 Experimental Results of Forground Colorization

The forground colouring results are shown in Figure 4.4. A total of three models were trained for comparison during the experimental phase. (a) is the structure proposed by Zou et al.[40] using LSTM for text encoding and for text image fusion. (b) is the original version was replaced using GRU and adjusting the learning rate to 0.00002, and (c) the structure proposed in Chapter 3 using Bert for text encoding and residual block for text image fusion.

As shown in the Figure 4.4, the coloring of the wings of the butterfly with GRU for text encoding and text image fusion and with Bert for text encoding and fusion of text image information using the residual block is closer to the ground truth than the structure with LSTM for text encoding and fusion of text image information. and compared to the structure with LSTM for text encoding and fusion of text image information, the coloring of the wings of the butterfly with GRU for text encoding and fusion of text image information is closer to the ground truth. The model proposed in Chapter 3, which uses Bert for text encoding and a residual block for text image fusion, shows the detail of the butterfly's yellow spots, which is better than the model proposed in Chapter 3 in terms of detailing and taking into account multiple colours when multiple colours are present.

In the Figure 4.5 it can be seen that the text encoding using Bert is better in terms of details such as the white window and the text at the top of the bus than the adapted model

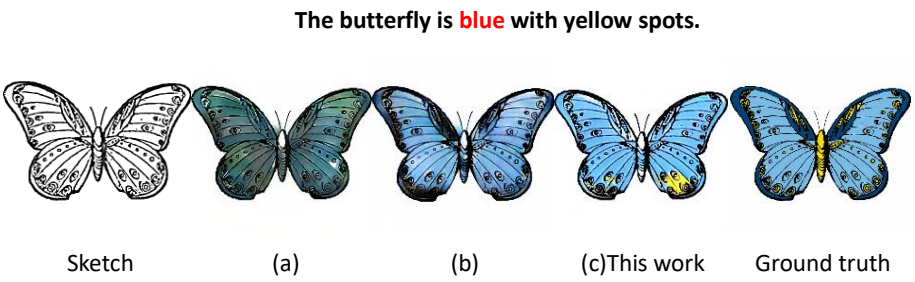

Figure 4.4. Experimental results of foreground colorization about detail.

using GRU for text encoding and text-image information fusion, and better in terms of colour than the model using LSTM for text encoding and text-image information fusion.

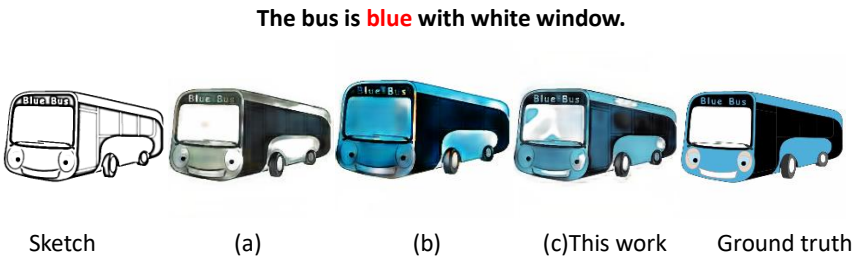

Figure 4.5. Experimental results of foreground colorization about coloring area.

In the Figure 4.6 , the model (c) proposed in chapter 3 is closer to the real image in terms of colour and window detail than the first two models.

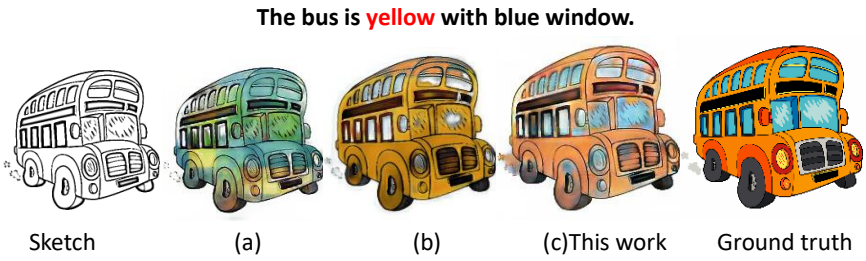

Figure 4.6. Experimental results of foreground colorization about the color Accuracy.

In the Figure 4.7, only the model proposed in Chapter 3 (c) reproduces the pink ears and scarf of the cat.

The use of Bert for text encoding has also been improved in terms of training speed. As shown in the Table 4.6, the speed of the model proposed in Chapter 3 using Bert for text

Figure 4.7. Experimental results of foreground colorization accuracy of multiple colors.

encoding and a residual block for text image fusion shows some improvement compared to the original model and the improved model using GRU.

Table 4.6. Training speed of foreground colorization.

Model name	Speed
Original System	0.34124 s/iter
Replacing LSTM with GRU	0.34033 s/iter
This Work	0.30886 s/iter

4.4.2 Experimental Results of Background Colorization

As shown in the Figure 4.8, (a) represents the input black-and-white sketch, (b) represents the colouring result of the network using Bert for text encoding and text-image information fusion via the residual block, and (c) represents the colouring result of the network using LSTM for text encoding and text-image information fusion before the improvement. Compared to the foreground colouring, the background colouring part is less enhanced. As shown in the Figure 4.8, the boundary between sky and ground is clearer for the network using Bert for text encoding and fusion of text-image information via the residual block (b), while the boundary is slightly blurred for the network using LSTM for text encoding and fusion of text-image information (c).

As shown in the Figure 4.9 the boundary between the sky and the ground is smoother for the network using Bert for text encoding and fusion of text-image information via the residual block (b), while the boundary is less smooth for the network using LSTM for text encoding and fusion of text-image information. However, as shown in the second row, the improvement in the background colouring part is very small and most of the colouring images do not change much. In terms of the training speed, as shown in Table 4.7 the training speed of the network

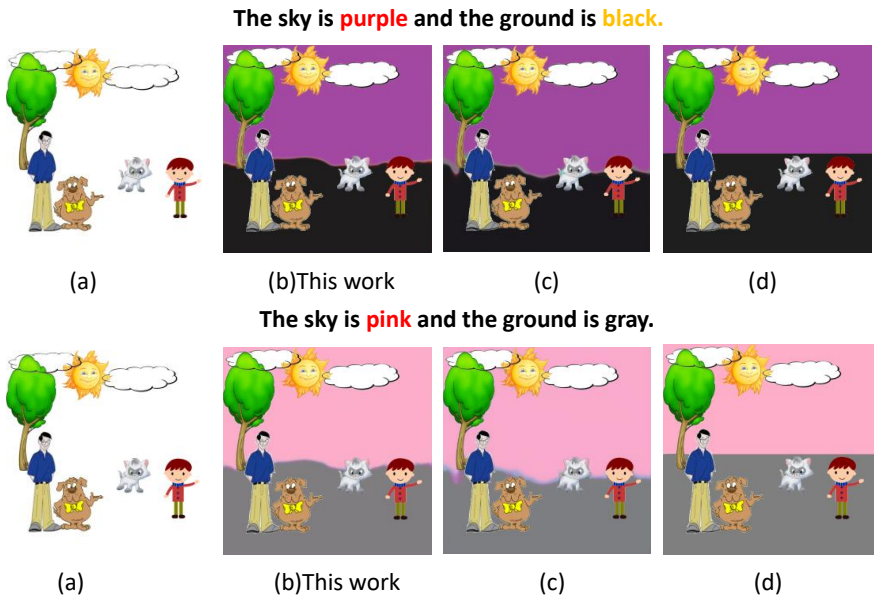

Figure 4.8. Experimental results of background colorization about boundaries.

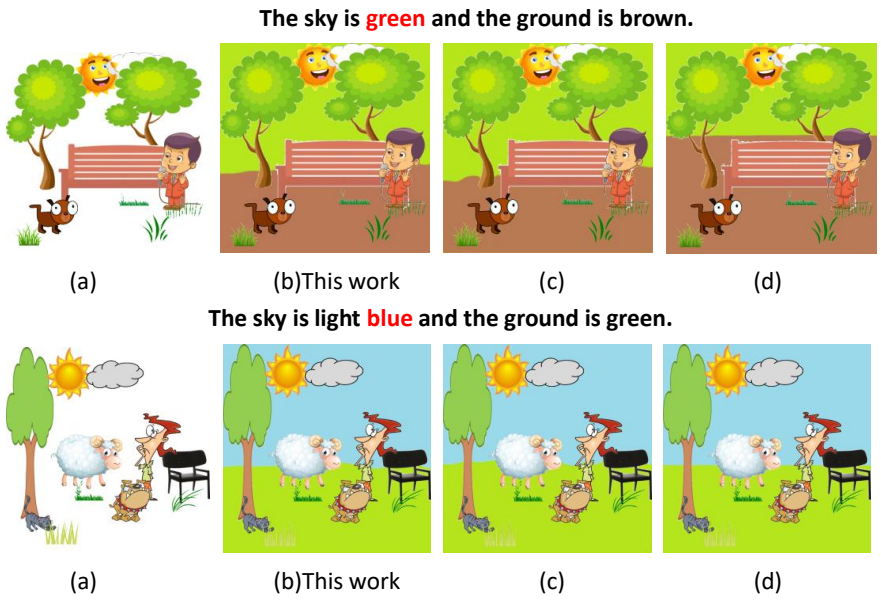

Figure 4.9. Experimental results of background colorization about the smoothness.

using Bert for text encoding and fusion of text-image information via the residual block is somewhat faster than that of the network using LSTM for text encoding and fusion of text-image information, but as with the colouring results, the improvement is not very significant.

Table 4.7. Training speed of background colorization.

Model name	speed
Original System	0.58653 s/iter
This Work	0.54736 s/iter

4.5 Summary

This chapter focuses on the foreground dataset and the background dataset on which the experiments were conducted, and introduces the details of the dataset such as the number of images, the composition of the dataset, the type and size of the images, etc. This was followed by a description of the colab platform used for training and the two graphics cards used. The training parameters such as batchsize, learning rate, etc. for foreground and background coloring are also introduced. Finally, the experimental results of foreground coloring and background coloring are shown and analyzed.

5 Limitation and discussion

5.1 Foreground Colorization

The current system still has many shortcomings. In terms of text, the individual foreground images are coloured separately, and it is common practice for people not to describe one object in a sentence, but to describe several objects in one sentence, for example by using the sentence "The chicken and the bus are yellow" rather than "The chicken is yellow. The bus is yellow". So there are still some linguistic limitations. In the future it would be more customary to use a single sentence to colour multiple images, which would require greater modifications in terms of dataset and model structure. In addition, colouring covers a smaller variety of objects, only 24 objects can be coloured, and it is not possible to depict more specific parts, for example the wheels of a car cannot be described individually. In the future, a larger and better data set would have to be created if the variety of colours covered is to be improved and the system is to be able to read more information in more detail.

In terms of the colouring results, as shown in the Figure 5.1, the model sometimes fails to colour in accordance with the colours provided. It color the red bench to the brown bench. Because most of the branch in the dataset arebrown, therrfore sometimes the model color the branch to brown even if the textual decvription is other color.

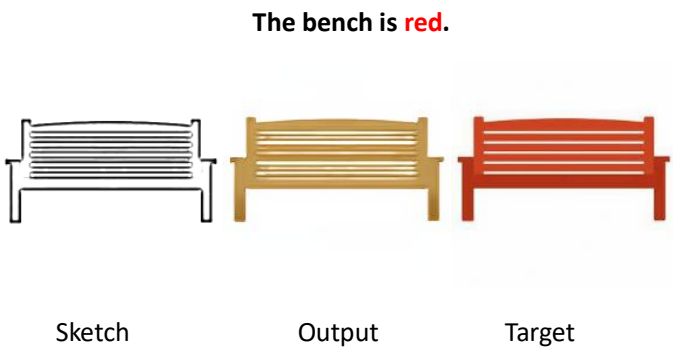

Figure 5.1. Limitation of foreground colorization.

As shown in the Figure 5.2 the foreground colorization model color the green branch to brown just lick the Figure 5.1 and there are some defects of the coloring result. The upper right part of the bench is not colored. As shown in the Figure 5.3 , the foreground colorization network color the green birt and the cloud to other colors. It can be seen that the accuracy

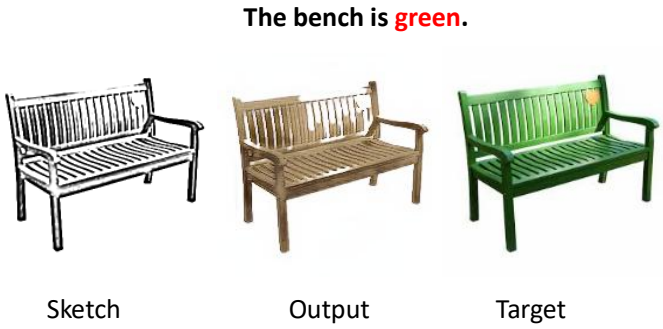

Figure 5.2. Limitation of foreground colorization.

of the proposed system for coloring green foreground objects is not ideal, and it does not extract and color the green keywords in the text information very well. It may be due to the fact that there are fewer green objects in the training set, so the results in green coloring are not very satisfactory. In the future, the accuracy of the model in terms of colouring could be

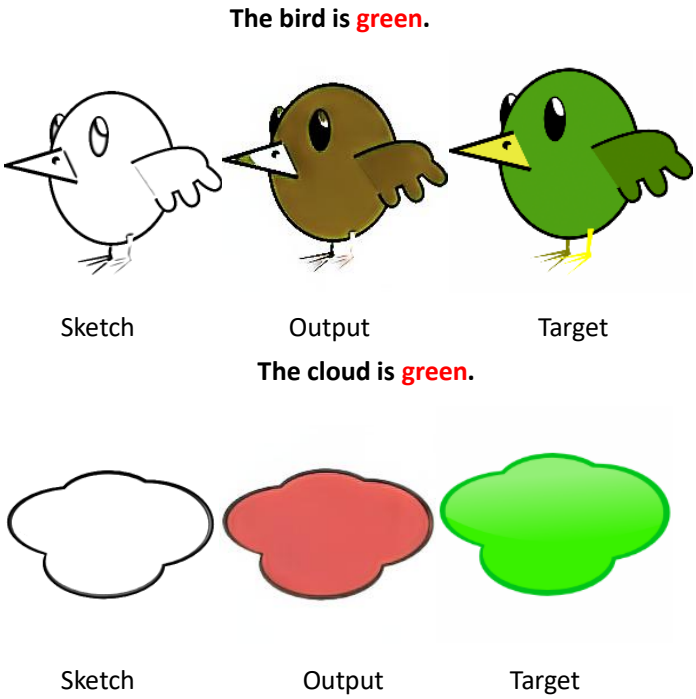

Figure 5.3. Limitation of foreground colorization.

improved by increasing the weight of colour information in the text coding or by further fusing the coding of colour words encoded by Bert with the text image information. In addition, the dataset can also increase the number of green and other color data in the foreground dataset to improve the accuracy of the model for green and other colors, in addition to objects such as

benches, dataset can not use a large number of brown color data, otherwise it is easy to make the model will be colored brown benches, ignoring the information in the text.

5.2 Background Colorization

There are also some shortcomings in the background colouring. As shown in the Figure 5.4, (a) represents the input black and white sketch, (b) represents the colouring result of the network output and (c) represents the real colour picture of the target. As can be seen from the picture in the first row, there is still a gap between the demarcation line of the background colouring map output by the model and the demarcation line of the target picture.

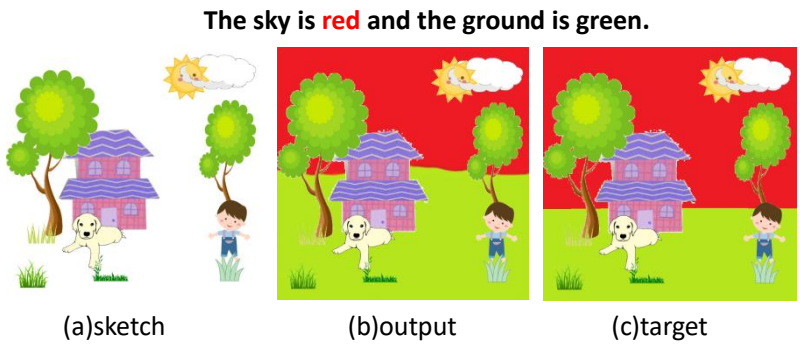

Figure 5.4. Limitation of background colorization about the height difference of the boundary.

As shown in the Figure 5.5, the demarcation lines of some of the images output through the model are still not smooth enough.These colored results still have some gaps that are more obvious.

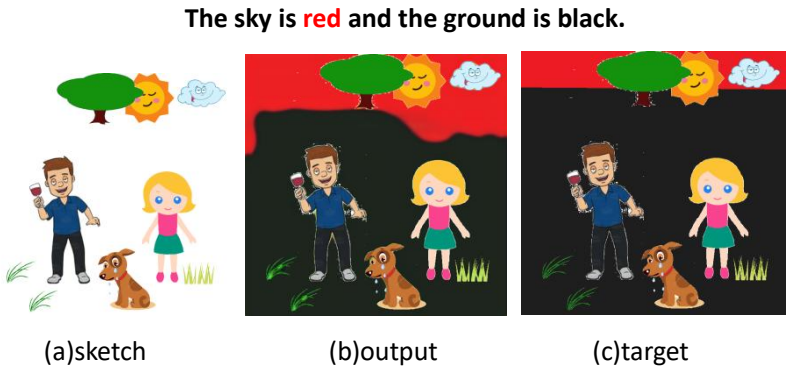

Figure 5.5. Limitation of background colorization about the smoothness.

In addition, as shown in the Figure 5.6 the demarcation line of the model output is still

somewhat unclear compared to the real image. The generated image has a white residue at the dividing line.

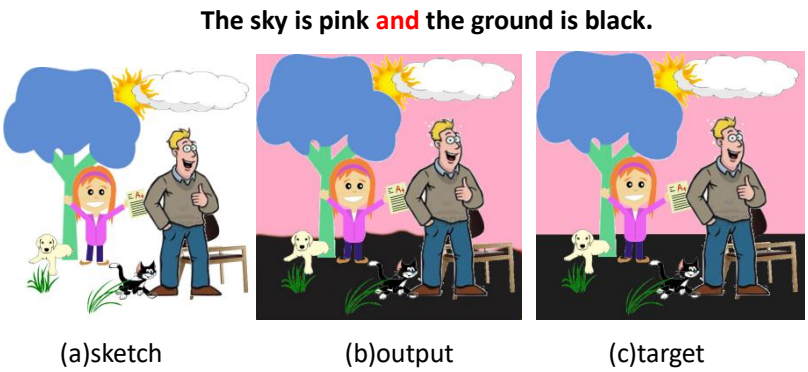

Figure 5.6. Limitation of background colorization about white residue in the boundary.

As shown in the Figure 5.7 , there is a certain colour difference between the demarcation line of the sky and the ground in the background coloured model output at the position of the demarcation line (the demarcation line of the background coloured model output in the fourth row is greenish).

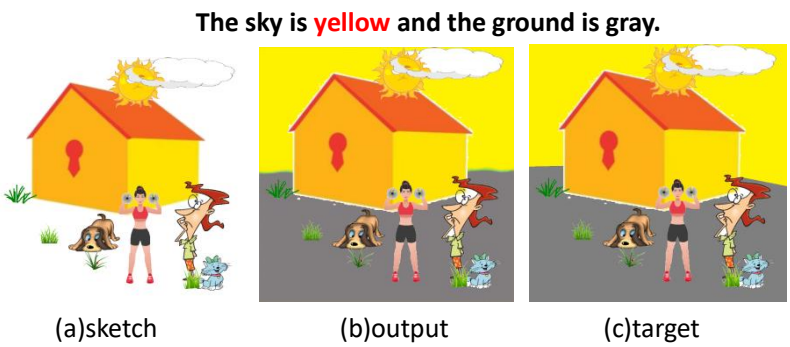

Figure 5.7. Limitation of background colorization about the color overlap at the boundary.

5.3 Summary

This chapter introduces the problems of the foreground coloring system and background coloring system proposed in this paper, and provides a specific analysis of the problems based on a specific coloring sample.

In the foreground coloring part, the coloring defects are mainly focused on color errors and some coloring noise. For objects with fixed colors such as benches, the model tends to

ignore the text information and color the object as its usual color directly, while for colors with small numbers in the data set such as green, the coloring network will often color it as other colors. Therefore, in order to solve this part of the problem, future improvements can be made to the dataset by adding more objects such as green, and the color of objects in a class should be as diverse as possible instead of mostly brown as in the bench dataset. It is also possible to increase the weight of color-related text in the network to better control color.

In the background coloring part, the structure in this paper is less enhanced, and the coloring defects are mostly consistent with the problems in the background coloring system proposed by Zou et al. i.e., there are still some blurred boundaries in the coloring results, sometimes the two colors at the boundaries overlap and make the boundaries other colors, and the boundaries are still not flat enough, sometimes there are some gaps, and the The height of the border is usually different from the real value. There is still a need for some improvement in the future.

Conclusion

This thesis focuses on improving an existing sketch colouring system by increasing the colouring results and training speed of the system. This thesis builds on the previous sketch colouring system by using Bert for text encoding and a residual block for text-image fusion. The main work in this thesis is as follows.

In the foreground colouring part, by using Bert for text encoding and the residual block for text-image information fusion, the sketch colouring results and training speed are improved. In particular, sketch coloring details and multiple color separation have been improved.

In terms of background colouring, by using the same improved method as for foreground colouring, i.e. text encoding by Bert and text-image information fusion by means of a residual block, a small improvement was achieved in the colouring of the background sketch, and although the boundary between sky and ground in the background was smoother and clearer than in the original system, the overall improvement was not significant. In terms of training speed, there is some improvement but not significant compared to the foreground colouring model.

References

[1] Welsh T, Ashikhmin M, Mueller K. Transferring color to greyscale images [C]. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques, 2002: 277–280.

[2] Chen T, Wang Y, Schillings V, et al. Grayscale image matting and colorization [C]. In Asian Conference on Computer Vision, 2004: 1164–1169.

[3] Deshpande A, Rock J, Forsyth D. Learning large-scale automatic image colorization [C]. In Proceedings of the IEEE international conference on computer vision, 2015: 567–575.

[4] Larsson G, Maire M, Shakhnarovich G. Learning representations for automatic colorization [C]. In European conference on computer vision, 2016: 577–593.

[5] Iizuka S, Simo-Serra E, Ishikawa H. Let there be color! Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification [J]. ACM Transactions on Graphics (ToG), 2016, 35 (4): 1–11.

[6] Aoki T, Nguyen V. Global Distribution Adjustment and Nonlinear Feature Transformation for Automatic Colorization [J]. Advances in Multimedia, 2018, 2018.

[7] Lee Y, Lee S. Automatic Colorization of Anime Style Illustrations Using a Two-Stage Generator [J]. Applied Sciences, 2020, 10 (23): 8699.

[8] Koleini M, Monadjemi S A, Moallem P. Automatic black and white film colorization using texture features and artificial neural networks [J]. Journal of the Chinese Institute of Engineers, 2010, 33 (7): 1049–1057.

[9] Lal S, Garg V, Verma O P. Automatic image colorization using adversarial training [C]. In Proceedings of the 9th International Conference on Signal Processing Systems, 2017: 84–88.

[10] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks [J]. arXiv preprint arXiv:1511.06434, 2015.

[11] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [J]. Advances in neural information processing systems, 2014, 27.

[12] Seo C W, Seo Y. Seg2pix: Few Shot Training Line Art Colorization with Segmented Image Data [J]. Applied Sciences, 2021, 11 (4): 1464.

[13] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks [C]. In International conference on machine learning, 2017: 214–223.

[14] Zhang H, Xu T, Li H, et al. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks [C]. In Proceedings of the IEEE international conference on computer vision, 2017: 5907–5915.

[15] Zhang H, Xu T, Li H, et al. Stackgan++: Realistic image synthesis with stacked generative adversarial networks [J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41 (8): 1947–1962.

[16] Mirza M, Osindero S. Conditional generative adversarial nets [J]. arXiv preprint arXiv:1411.1784, 2014.

[17] Isola P, Zhu J-Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks [C]. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017: 1125–1134.

[18] Wang T-C, Liu M-Y, Zhu J-Y, et al. High-resolution image synthesis and semantic manipulation with conditional gans [C]. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018: 8798–8807.

[19] Zhu J-Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks [C]. In Proceedings of the IEEE international conference on computer vision, 2017: 2223–2232.

[20] Zhang R, Zhu J-Y, Isola P, et al. Real-time user-guided image colorization with learned deep priors [J]. arXiv preprint arXiv:1705.02999, 2017.

[21] Kataoka Y, Matsubara T, Uehara K. Automatic manga colorization with color style by generative adversarial nets [C]. In 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2017: 495–499.

[22] Reed S, Akata Z, Yan X, et al. Generative adversarial text to image synthesis [C]. In International conference on machine learning, 2016: 1060–1069.

[23] Reed S E, Akata Z, Mohan S, et al. Learning what and where to draw [J]. Advances in neural information processing systems, 2016, 29.

[24] Dumoulin V, Belghazi I, Poole B, et al. Adversarially learned inference [J]. arXiv preprint arXiv:1606.00704, 2016.

[25] Makhzani A, Shlens J, Jaitly N, et al. Adversarial autoencoders [J]. arXiv preprint arXiv:1511.05644, 2015.

[26] Larsen A B L, Sønderby S K, Larochelle H, et al. Autoencoding beyond pixels using a learned similarity metric [C]. In International conference on machine learning, 2016: 1558–1566.

[27] Xu T, Zhang P, Huang Q, et al. Attngan: Fine-grained text to image generation with attentional generative adversarial networks [C]. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018: 1316–1324.

[28] Nam S, Kim Y, Kim S J. Text-adaptive generative adversarial networks: manipulating images with natural language [J]. Advances in neural information processing systems, 2018, 31.

[29] Sheng B, Sun H, Chen S, et al. Colorization using the rotation-invariant feature space [J]. IEEE computer graphics and applications, 2011, 31 (2): 24–35.

[30] He M, Chen D, Liao J, et al. Deep exemplar-based colorization [J]. ACM Transactions on Graphics (TOG), 2018, 37 (4): 1–16.

[31] Xiao C, Han C, Zhang Z, et al. Example-Based Colourization Via Dense Encoding Pyramids [C]. In Computer Graphics Forum, 2020: 20–33.

[32] Liu G, Chen X, Hu Y. Anime sketch coloring with swish-gated residual U-net [C]. In International Symposium on Intelligence Computation and Applications, 2018: 190–204.

[33] Liang X, Su Z, Xiao Y, et al. Deep patch-wise colorization model for grayscale images [M] // Liang X, Su Z, Xiao Y, et al. SIGGRAPH ASIA 2016 Technical Briefs. 2016: 2016: 1–4.

[34] Wu E, Liu F. Robust image metamorphosis immune from ghost and blur [J]. The visual computer, 2013, 29 (4): 311–321.

[35] Zhang L, Ji Y, Lin X, et al. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan [C]. In 2017 4th IAPR Asian conference on pattern recognition (ACPR), 2017: 506–511.

[36] Zhang R, Isola P, Efros A A. Colorful image colorization [C]. In European conference on computer vision, 2016: 649–666.

[37] Dong H, Yu S, Wu C, et al. Semantic image synthesis via adversarial learning [C]. In Proceedings of the IEEE international conference on computer vision, 2017: 5706–5714.

[38] Li B, Qi X, Lukasiewicz T, et al. Manigan: Text-guided image manipulation [C]. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 7880–7889.

[39] Zou C, Yu Q, Du R, et al. Sketchyscene: Richly-annotated scene sketches [C]. In Proceedings of the european conference on computer vision (ECCV), 2018: 421–436.

[40] Zou C, Mo H, Gao C, et al. Language-based colorization of scene sketches [J]. ACM Transactions on Graphics (TOG), 2019, 38 (6): 1–16.

[41] Devlin J, Chang M-W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv:1810.04805, 2018.

[42] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [J]. Advances in neural information processing systems, 2017, 30.

[43] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural computation, 1997, 9 (8): 1735–1780.

[44] Cho K, Van Merriënboer B, Bahdanau D, et al. On the properties of neural machine translation: Encoder-decoder approaches [J]. arXiv preprint arXiv:1409.1259, 2014.

[45] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift [C]. In International conference on machine learning, 2015: 448–456.

[46] Chen W, Hays J. Sketchygan: Towards diverse and realistic sketch to image synthesis [C]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 9416–9425.

[47] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C]. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 770–778.

[48] Winnemöller H. Xdog: advanced image stylization with extended difference-of-gaussians [C]. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, 2011: 147–156.

修改记录

（1）第一次修改记录：摘要，修改前：使用 Bert 和 Residual Block 对背景着色和前景着色进行改进，使草图着色可以获得更好的上色效果并能够更快地练。

修改后：但是原结构仍然存在一些问题，在原结构中使用了 LSTM 进行文本编码和文本-图像信息融合，但是因为着色时的文本通常较短，因此不需要使用 LSTM 进行长时间的记忆，此外使用 LSTM 使得系统在训练时所消耗的时间较长。因此本论文主要在背景着色和前景着色部分针对这两部分进行提升。在文本编码部分使用 Bert 对文本描述进行整句的编码，在文本-图像信息融合部分使用 Residual Block 进行文本-图像信息融合，最终使草图着色可以获得更好的上色效果并能够更快地训练。

（2）第二次修改记录：35 页图片 4.3 标题，修改前：Experimental results

修改后：Experimental results of foreground colorization

记录人（签字）：李承潇

指导教师（签字）：刘宇

Acknowledgement

Four years have gone by in a flash and I would like to thank the teachers and students who have helped and encouraged me during this time.

Firstly, I would like to thank my supervisor, Associate Professor Liu Yu, for providing me with a lot of constructive advice during my final year. When I encountered difficulties in my research, Associate Professor Liu Yu also helped me to find solutions and solutions in a timely manner, which benefited me a lot.

I would also like to thank my classmates. They were very helpful when I encountered difficulties. They have also helped me a lot in my life during the four years.

On the other hand, I would like to thank the Colab platform for providing me with a relatively inexpensive laboratory environment and equipment while I was doing my final research.

Finally, I would like to thank my parents who, although they do not know anything about computers, try their best to understand and encourage me when I am in trouble.

Once again, I would like to express my sincere gratitude to my teachers, family and friends.