creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

이학석사 학위논문

Grayscale Image Colorization using Modified FusionNet

(변형된 FusionNet을 이용한 회색조 이미지의 자연스러운 채색)

2021년 2월

서울대학교 대학원

협동과정 계산과학전공

좌 민 제

Grayscale Image Colorization using Modified FusionNet

(변형된 FusionNet을 이용한 회색조 이미지의 자연스러운 채색)

지도교수 강 명 주

이 논문을 이학석사 학위논문으로 제출함

2020년 10월

서울대학교 대학원

협동과정 계산과학전공

좌 민 제

좌 민 제의 이학석사 학위논문을 인준함

2020년 11월

위 원 장 국 윤 (인)

부 위 원 장 강 명 주 (인)

위 원 하 명 수 (인)

Grayscale Image Colorization using Modified FusionNet

by

Minje Jwa

A DISSERTATION

Submitted in partial fulfillment
of the requirements
for the degree of
Master of Science
to the faculty of the Graduate School of
Seoul National University
February 2021

© 2021 Minje Jwa

All rights reserved.

Abstract

Grayscale Image Colorization using Modified FusionNet

Minje Jwa

Computaional Science and Technoloty
The Graduate School
Seoul National University

In this paper, we propose a grayscale image colorizing technique. The colorization task can be divided into three main ways, the Scribble-based method, Exemplar-based method and Fully automatic method. Our proposed method is included in the third one. We use a deep learning model that is widely used in the colorization field recently. We propose Encoder-Docoder model using Convolutional Neural Networks. In particular, we modify the FusionNet with good performance to suit this purpose. Also, in order to get better results, we do not use MSE loss function. Instead, we use the loss function suitable for the colorizing purpose.

We use a subset of the ImageNet dataset as the training, validation and test dataset. We take some existing methods from Fully automatic Deep Learning method and compared them with our models. Our algorithm is evaluated using a quantitative metric called PSNR (Peak Signal-to-Noise Ratio). In addition, in order to evaluate the results qualitatively, our model was applied to the test dataset and compared with various other models. Our model has better performance both quantitatively and qualitatively than other models. Finally, we apply our model to old black and white photographs.

Key words: Deep learning, Grayscale image colorization, fully convolutional network, FusionNet
Student Number: 2018-21347

Contents

Abstract **i**

1 Introduction **1**

2 Related Works **4**
2.1 Scribble-based method 4
2.2 Exemplar-based method 5
2.3 Fully automatic method 6

3 Proposed Method **8**
3.1 Method Overview 8
3.2 Loss Function 8
3.3 Architecture detail 10
3.3.1 Encoder 11
3.3.2 Decoder 12
3.3.3 Bridge 12

4 Experiments **14**
4.1 CIE Lab Color Space 15
4.2 Dataset 16
4.3 Qualitative Evaluation 17
4.4 Quantitative Evaluation 18
4.5 Legacy Old image Colorization 20

5 Conclusion **23**

The bibliography **24**

Abstract (in Korean) **28**

Chapter 1

Introduction

Many old historical photographs are in black and white. When people look at the image, they cannot feel the liveliness when the picture was taken. So, colorizing grayscale images is a task that brings such old pictures to life. Image colorization refers to the technique that adds colors to grayscale images or videos. When a person looks at a grayscale image, color can be estimated through a number of known information. However, finding the perfect color is impossible even for humans. Colorization is a ill-posed problem and does not give a unique solution. It can be said that grayscale image colorizing is more about finding plausible colors rather than finding exact colors.

The grayscale image colorizing is an important task in many fields such as the film industry, historical data reviewing, photography technology and so on. In particular, as the mass production of webtoon is increasing all over the world these days, the field of automatic cartoon coloring is also in the spotlight [27]. Significant progress has been made in this area, but automatic image colorization remains a challenge.

In colorization fields, there are three methodologies : the Scribble-based method, Exemplar-based method and Fully automatic method. The first and second methods are largely classified as user-guided edit propagation method. And the third one is called data-driven automatic colorization method. In this paper, we propose a colorization technique using Fully automatic Deep Learning method. Deep learning has made a lot of progress over the past decade. In particular, in the field of image processing, people are getting very

good results using deep learning. Recently, a lot of researches on colorizing using deep learning technology based on Convolutional Neural Networks are being conducted, and the results are shown to be very good. Prior to the advent of deep learning technology, most effective methods relied on human intervention.

Therefore, we consider a method of grayscale image colorizing based on CNNs (Convolutional Neural Networks). Most of the colorization methods have a feature extraction part and a reconstruction part. And this structure is called Encoder-Decoder model in deep learning fields. There are various types of Encoder-Decoder model. When colorization was performed using U-net [20], the result was good [18]. The structure of U-net can be seen in detail in Figure 1.1. In particular, we have variously modified the FusionNet [19] since FusionNet has a similar structure to U-net. The structure of FusionNet can be seen in detail in Figure 1.2. FusionNet is an Encoder-Decoder model with very good performance. We adjust the number of kernels of the last layer and make various modifications to fit our purpose. Since the performance of the model using the basic MSE loss function is not good, training was carried out by defining a new loss function appropriate to the colorization goal.

In deep learning, it is important to properly select training, validation and test datasets. There are various datasets that have been previously disclosed that can be applied to deep learning networks. Among them, the ImageNet [21] dataset has about 1.3M images under 1000 labels. That is, there are so many different images. Thus, we use ImageNet dataset for training, validation and test dataset. This is because it is important to use various images for training models.

The outline of this paper is as follows. In Chapter 2, we reviews the related works and previous colorizing methods. We describe detailed information about our model and loss function in Chapter 3. The quantitative and qualitative results of the experiment can be seen in Chapter 4. Finally, we conclude our results and discuss further works in Chapter 5.

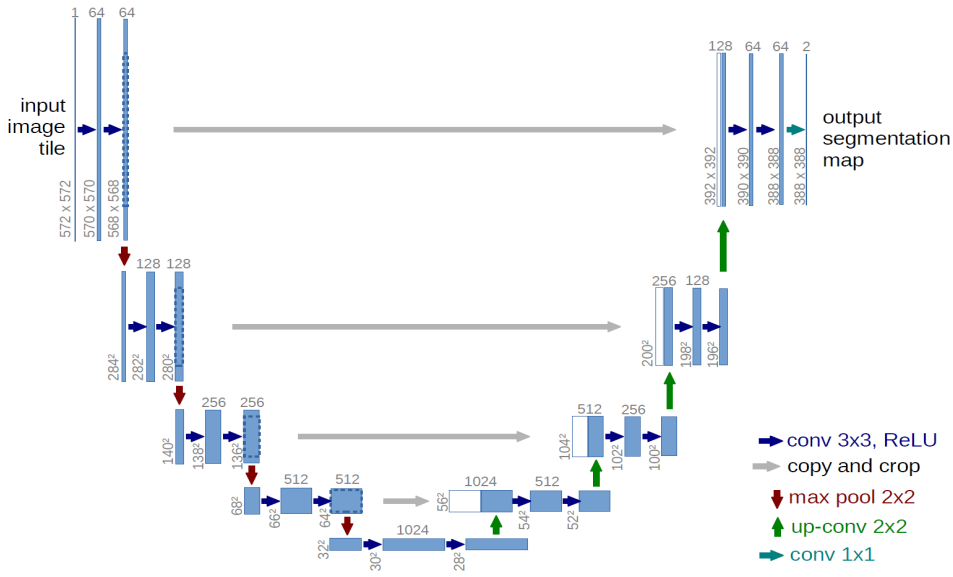


Figure 1.1: Detail description U-net

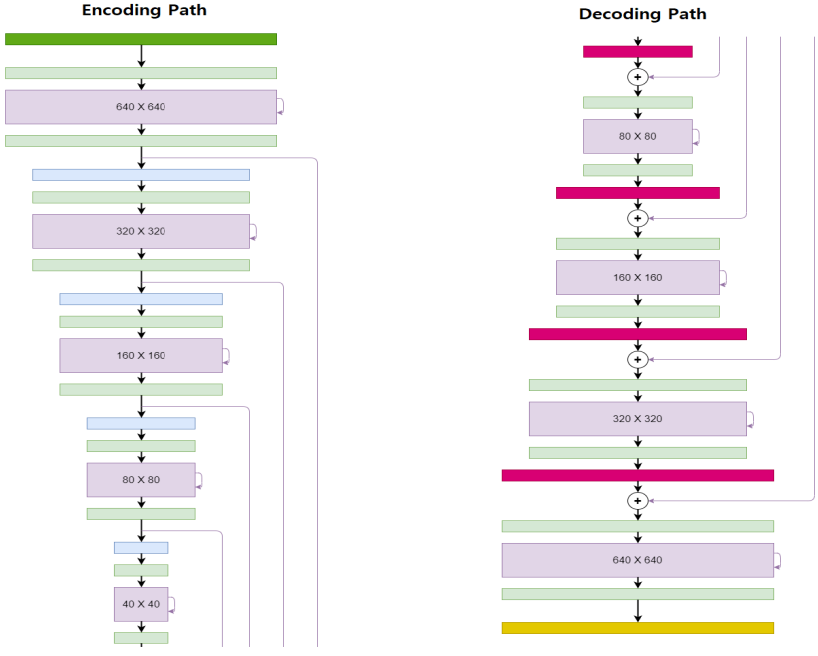


Figure 1.2: Detail description FusionNet

Chapter 2

Related Works

Several colorizing techniques have been proposed over the past decades. The field of grayscale image colorizing can be roughly divided into three : The Scribbled-based method, the Exemplar-based method and the Fully automatic method. The first two methods require human intervention. Thus, they are called as user-guided edit propagation methods. However, considering the application of the model, it is impossible to intervene for colorizing all grayscale images. This is because it takes too much cost and time. Therefore, we need an end-to-end automatic method without human assistance. And the third method was introduced. It produces a color image that fits the grayscale image with end-to-end without human help in any image.

2.1 Scribble-based method

In the Scribbled-based method, the user gives a color scribble and then the model receives a hint about the color. It is interactive way to colorize grayscale image by placing scribbles. Literally, the input value is the user's doodle drawing or point of the appropriate colors in the grayscale image. An example of the input of Scribble-based method can be seen in Figure 2.1. In Levin et al. [16], this method was first devised. These methods assume that spatial neighboring pixels of pixels with color information must also have similar colors. And after that, other methods were proposed [10, 26, 17]. In Zhang et al. [29], they made a model that randomly selects several pixels from

the training image and puts them together as scribble input. In the Xiao et al. [25], they developed the method of Zhang et al. [29], and made a model that could put the overall color theme of the image as a global input. However, these methods require a large amount of user input, especially when dealing with complex images. In addition, because choosing a color scheme involves human subjectivity, people can choose the wrong color.

Figure 2.1: An example of the input of Scribble-based method. This is from [29]. First image is grayscale image and the second image is a pixel scribble that humans put into iterative input.

2.2 Exemplar-based method

In Exemplar-based method, the grayscale image and another color image are entered together as input. The color image is an example image with colors that we expect to emerge as a result of grayscale images. This image is called an exemplar image or reference image. An example of the input of Exemplar-based method can be seen in Figure 2.2. Starting with Welsh et al. [24], various exemplar-based methods appeared. It transferred the colors by matching the statistic within the pixel's neighborhood. Kang et al. [14] proposed a better method by using a webtoon image as a dataset and putting an attention structure. And various methods have been devised [22, 5, 3]. This

type of method significantly reduces user's intervention than the previous methods, but it is difficult to process a large amount of images because it always requires a reference image that should be similar to a grayscale image.

Figure 2.2: An example of the input of Exemplar-based method. This is from [24]. First image is exemplar (or reference) image and the second image is a grayscale image and last image is colorizing picture that we expect.

2.3 Fully automatic method

The Fully automatic method is a fully automatic end-to-end deep learning based colorizing methods and has been widely used in these days. In particular, there are many ways to make models based on CNNs that receive only grayscale images as a input. So, they are often called data-driven automatic colorization. Cheng et al. [4] first created a fully-automatic colorization method using a deep neural network. As the ability to use large datasets has increased and methods of deep neural networks have evolved, there have been more ways to colorize using deep learning. By training a large amount of data, people can predict pluasible colors on a grayscale image. Most of the models are basically based on the structure of Encoder-Decoder models. The reason is that we basically need feature extraction part and reconstruction part and they have a similar structure to Encoder-Decoder models. In Iizuka et al. [11], local and global features are trained together, giving additional class information to the basic CNNs Encoder-Decoder structure. Baldassarre et al. [2] following Iizuka et al. [11] also trained semantic information by giving classification terms similarly. Zhao et al. [30] also uses semantic segmentation

information to colorize grayscale image. Zhang et al. [28] uses a very basic structured model, but learning the color distribution of every pixel was a very creative start. Recently, there are many ways to colorize grayscale images using GANs (Generative Adversarial Networks) [7]. Isola et al. [13] uses conditional GAN using U-net based generator. In addition to that, Nazeri et al. [18] was proposed to use a similar method. Recently, a method of using Improved WGAN [8] while giving class information like Iizuka et al. [11] was argued in Vitoria et al. [23]. Also, Variational autoencoders (VAE) [15] have also been used to learn a color embedding in Deshpande et al. [6].

Recently, new methods have emerged that deviate from these three big parts. In particular, some methods were influenced by the NLP (Nature Language Processing). There are methods for users to enter information about the target picture color in sentences or words [1]. Changqing et al. [31] created a dataset [33] themselves, and created a model in which the user can directly enter texts as the information about the sketch. And in [32], the model was further developed.

Chapter 3

Proposed Method

3.1 Method Overview

In this chapter, we will explain detail method how to colorize grayscale image with our model. Given a grayscale input image L, we obtain a, b values such that a and b are chrominance channel images in the CIE Lab Color Space. The model receives the input image (grayscale image), passes through an encoding part and a decoding part, and finally produces two channels. There are several Encoder-Decoder models. We select FusionNet [19] as the base model and make several modifications. Also, a new loss function is applied for the inferring point estimates of color from the predicted color distribution.

3.2 Loss Function

A description of the new loss function is as follows. For input lightness data $X \in \mathbb{R}^{h \times w \times 1}$ (This is a grayscale image), the model have to find the a, b channels $\hat{Y} = \mathcal{F}(X)$ ($\hat{Y} \in \mathbb{R}^{h \times w \times 2}$). In this form, the available loss function is MSE (Mean Square Error). MSE function is used in various tasks as well as in colorization. When MSE is used, it can be seen as an image-to-image (pixel-by-pixel) regression approach. The equation for MSE function can be written as a following equation.

$$L_{MSE} = \frac{1}{2} \sum_{h,w} \|Y - \hat{Y}\|_2^2. \tag{3.1}$$

However, the optimal solution of this loss function is the average of set for the image. If we apply this value, the saturation (i.e. a, b value) becomes almost intermediate, which cannot represent the vivid saturation found in nature. Therefore, if we use this loss function, we get a grayish and awkward color instead of multimodal colorization of natural saturation.

So, we would like to think about another way in terms of finding a certain label for each pixel of the output. This method was first proposed in Colorul Image Colorization, Zhang et al. [28], and we want to use this method to specify the loss function. In terms of multinomial classification, we can use a multinomial Cross-entropy loss function. We have to learn mapping $\hat{Z} = \mathcal{H}(X)$ where $\hat{Z} \in \mathbb{R}^{h \times w \times n}$ and $n = 313$. To compare predicted $\hat{Z}$ against ground truth Z, we have to convert ground truth color Y to Z.

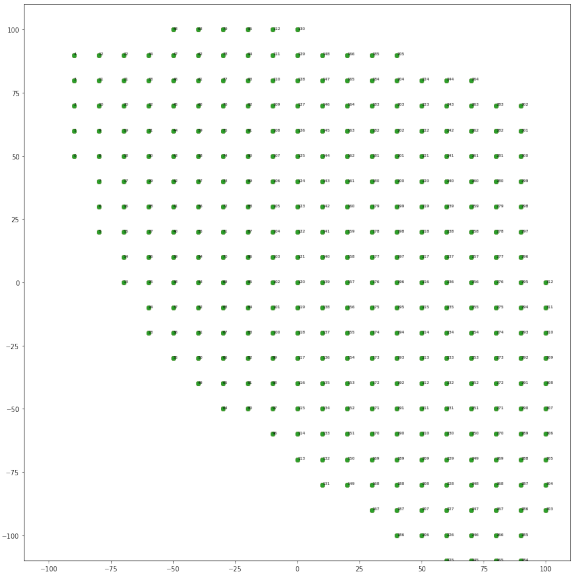


Figure 3.1: Quantized bin of 313 representative points in CIE Lab Color space for values of a and b

To explain this method, we quantize 313 representative points in the a, b channel like Figure 3.1. And for the a, b channel of a particular pixel, the

nearest representative point is designated as the label for that pixel. This method is called the k-nearest neighborhood method. Each pixels of ground truth image Y can be encoded as vector as common output of the classification problems. We do not make it one-hot vector, and find five quantized a, b bin in order close to that pixel. In other words, we find 5-nearest neighbors. And weight them proportionally to their distance from the ground truth using a Gaussian kernel with $\sigma = 5$. When we apply this method to each pixel, we have $h \times w$ vectors. This allows to apply cross-entropy loss function to each pixel of our model's output and ground truth. We specify the loss function in the expression below that adds all the values of each pixel. And by adding all those values of each pixel, we specify the loss function.

Then we can use multinomial cross entropy loss, defined as in Equation 3.2.

$$L_{cl} = -\sum_{h,w}\sum_{n} Z_{h,w,n} \log(\hat{Z}_{h,w,n}). \tag{3.2}$$

3.3 Architecture detail

A detailed architecture description for a model that infers a, b values of CIE lab color space by receiving input L containing grayscale lightness information is as follows. And our model is depicted in Figure 3.3. Architecture consists of 3 parts (Encoding part, Decoding part, Bridge). Overall, our model consists of 4 encoding blocks, 4 decoding blocks and middle bridge.

As mentioned earlier, we modify the Encoder-Decoder model called FusionNet [19]. The input size used in FusionNet is 640×640 , while the input size in this paper is 256×256. In FusionNet, the size of the feature map is reduced five times, but our model is reduced only four times. This is because input size of our model is smaller than FusionNet's input size, so if we reduce the size of the feature map a lot, we can lose a lot of information. We use 2-stride convolution layer to reduce the size of the feature map. In the FusionNet, Maxpooling is used to decrease the size of the feature map.

It is the same as FusionNet that ResBlock (Residual Block) is placed in the middle of the Encoder and Decoder blocks. ResBlock was first presented

in He et al. [9]. The ResBlock can be taken a variety of forms other than those seen in the Figure 3.2. We can create a ResBlock in various formats if we add the feature map to this previous one after going through a few Convolutional layers. It is designed to make a kind of shortcut (skip connection) so that gradient can flow well even when the layer is deep. There are long skip connections that connect the information of the previous feature map to the later feature map. And the filter size of all Convolutional layers was unified as 3×3.

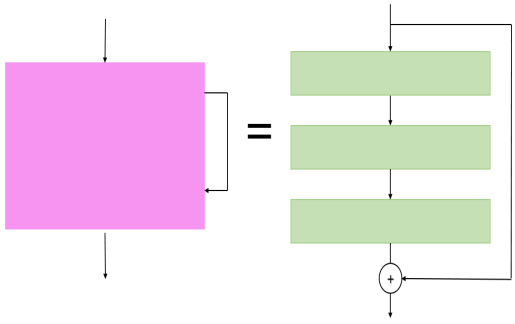


Figure 3.2: Detail description Residual Block from FusionNet [19]

3.3.1 Encoder

The encoder part is composed of 4 encoding blocks. To explain the encoding block in detail, the feature map is reduced in half through 2-stride convolution firstly. Then it goes through ResBlock (Residual Block). A detailed description of the Resblock used in our proposed architecture can be found in Figure 3.2. Resblock provides the result of adding the first entering feature map and the last feature map that comes out after Convolutional + Batch-normalization [12] three times. Lastly, the feature map from Resblock goes through the Convolutional + batch-normalization layer one more time and then goes into the next encoding block. Relu activation function was used for all Convolutional layers in the encoding block.

3.3.2 Decoder

Decoder has a similar structure to Encoder. Decoder is also composed of 4 decoding blocks. Firstly, the incoming feature map is doubled through Upsampling. Then it goes through Convolutional + Batch-normalization layer and enters the ResBlock. Resblock uses the same structure as used in the Encoding block. The feature map from ResBlock is concatenated with the feature map of the same size in the encoding block to create an expanded feature map (long skip-connection). After Convolutional + batch-normalization is performed once more, the next decoding block is entered. Like Encoder, all convolution layers use the Relu activation function, but the last layer uses the Softmax activation function.

3.3.3 Bridge

Bridge also follows a similar format to Encoder and Decoder. FusionNet [19], which we are based on, also has the middle bridge and they reduce the feature map size once more. However, our model receives the feature map from Encoder and sends it to the Decoder section in constant size (same resolution). After finishing with Encoder, the incoming feature map goes through Convolutional + batch-normalization layer and then enters to the Resblock. The feature map from the ResBlock is finally entered into another Convolutional + batch-normalization. And then the feature map goes into the Decoder part.

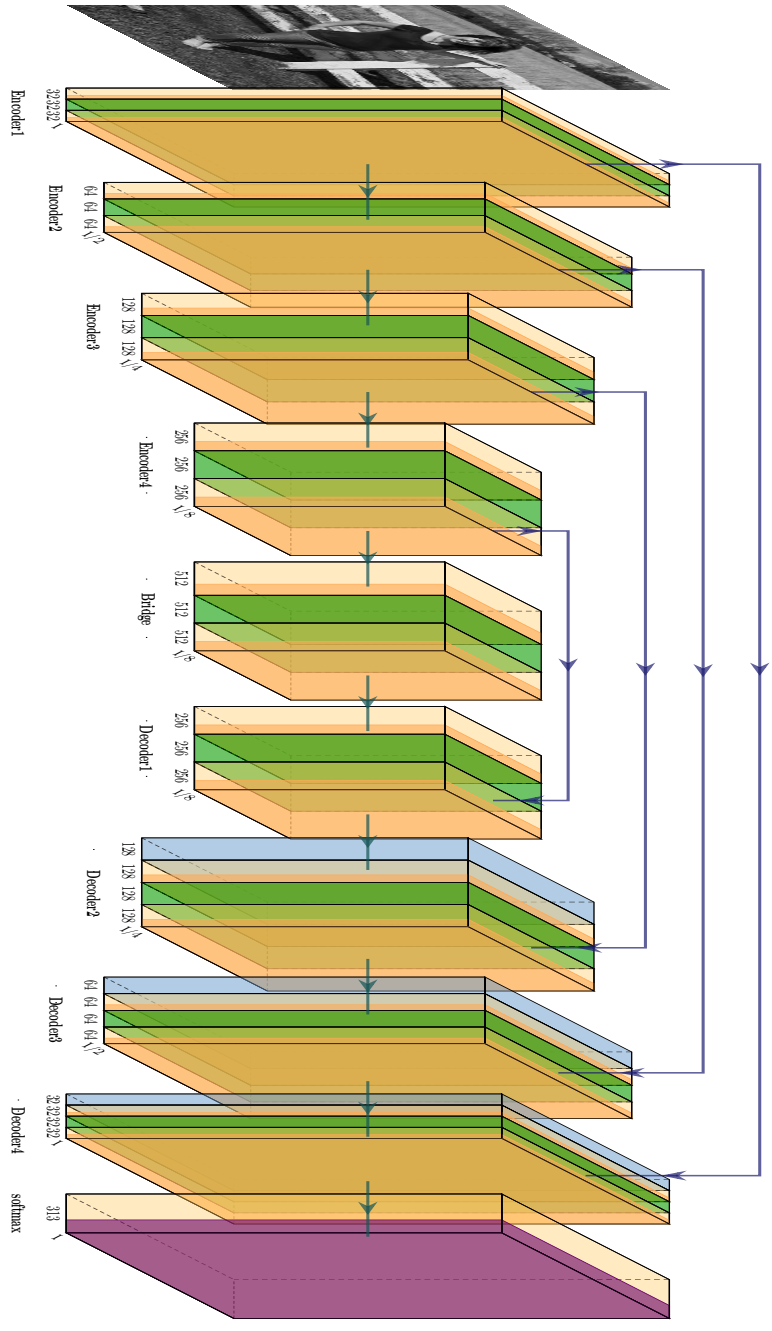


Figure 3.3: Proposed Architecture

Chapter 4

Experiments

Table 4.1: Experimental environment

Hardware Specification	
CPU	Intel Core i5-6500
GPU	TITAN RTX
Memory Capacity	16GB
Software Specification	
Python	Version : 3.7.6
TensorFlow	Version : 1.13.1
Operation System	Linux Ubuntu 16.04

We verified the validity of our end-to-end model by training it using the ImageNet [21] dataset. And the environment used for the experiment is the same as the Table 4.1. We implement the network model on the NVIDIA TITAN RTX GPU, using Google's TensorFlow architecture and Keras architecture. We train the model using a batch size of 32 and epoch 200 and takes about three days on the GPU. The loss function used for training is multinomial categorical cross-entropy as described in Chapter 3.2., and use SGD as optimizer.

In addition, Colorful Image Colorization [28] and Deep Kolarization [2] are selected as the comparison group of the proposed method in this paper. Those methods are also Fully automatic method using Deep learning networks. In

this paper, the evaluation metrics used to check the similarity between the Ground Truth image and the prediction result was selected as the Peak Signal-to-Noise Ratio (PSNR).

4.1 CIE Lab Color Space

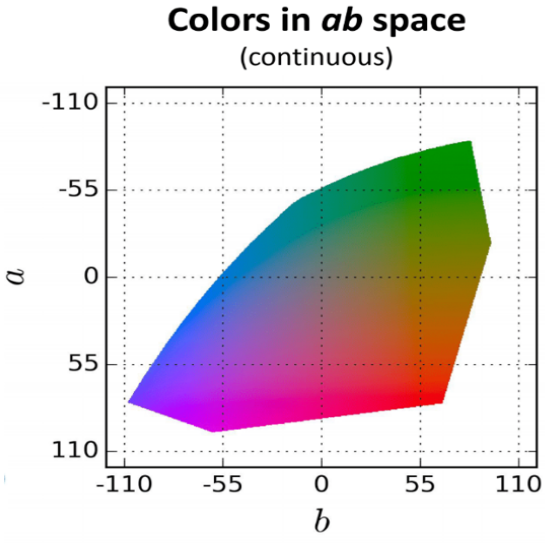


Figure 4.1: CIE Lab Color Space

The CIE Lab Color space is a color space defined by the International Commission on illumination (CIE). This is one of the types of Color Space such as RGB, YUV, etc. Other image processing process tasks use RGB a lot, but most of the image colorization field seems to use Lab color space. This is probably because there are two output kernels and less than three in RGB, which is advantageous for training. Also, because the grayscale kernel called L and the chrominance kernel called ab are related. Channel L stands for lightness of the color. L has a value from 0 to 100. 0 means black, 100 means white. Channel a specifies red/magenta and green color. If a is negative, it means the color with more green, and if a is positive, it can be seen as the color with more red. Channel b specifies yellow and blue colors. If b is negative,

it is blue, and if b is positive, it is yellow. Since the range of L is $[0, 100]$ and the range of ab channel is $[-128, 127]$, we normalize all channels to $[0, 1]$ using the following equations. This is to speed up the learning speed.

$$L = L/100. \tag{4.1}$$

$$ab = (ab + 128)/255. \tag{4.2}$$

4.2 Dataset

Table 4.2: Dataset Explanation

Original ImageNet dataset	
Training	1,281,167
Validation	50,000
Test	100,000
Selected dataset for training	
Training	80,000
Validation	20,000
Test	20,000

ImageNet [21] datasets are already widely used in image processing. In particular, image colorization also uses this dataset a lot. ImageNet has about 1.3M images and consists of 1000 labels. It is based on the name nodes contained in the word dataset WordNet That is, the dataset is composed of millions of pictures within a wide variety of sets. In Figure 4.5, we can see some examples of images from the ImageNet dataset. As we can see in the example, it contains a wide variety of images. Since there are such various images, we think this dataset is suitable for colorizing tasks. This is because the more various images are trained, the easier it is to predict the color of more grayscale images. In order to simplify training and reduce running times, only a small subset of images is used (train : validation = 8:2). Therefore, we randomly select 80,000 from the ImageNet for the training dataset, and 20,000

randomly from the ImageNet for the validation dataset. The description of Dataset can be found in Table 4.2.

4.3 Qualitative Evaluation

We used networks of Zhang et al. [28] and Baldassarre et al. [2] for comparison. And the results of those methods were compared with the results obtained using our proposed method. Zhang et al. [28] and Baldassarre et al. [2] used the ImageNet [21] dataset as we did. However, since we used a subset of the ImageNet dataset, we directly trained models based on our training dataset and validation dataset for fairness.

ImageNet dataset is also used as the test dataset. Among them, 20,000 images are randomly selected and used. Several colorizing results are shown on our validation set in Figure 4.6.

The second column is the grayscale image to be entered as the input of the models, the first column is the ground truth, the third column is the result of Zhang et al. [28], the fourth column is the result of Baldassarre et al. [2], and the last row shows the result of our proposed method. The network of Zhang et al. [28] and our network produce plausible results because they use the similar loss function as ours. Among them, our proposed model produces the most vivid colors, generating near-photo-realistic pictures. On the other hand, Baldassarre et al. [2] uses MSE loss function, resulting in relatively desaturated and grayish colors than other models. In particular, as we can see from the fourth and fifth rows, our model tends to colorize particularly well when the background is blue or green. And, like the last row, it seems to be able to find a specific bright color well like orange.

Although the result seems to be quite good in most images, we can see some examples of images that do not find color well in Figure 4.2. Looking at the example, we can see that even when humans can't conceive of colors, our model cannot even find colors. In particular, in our output image(third column), it seems that our model does not find the color very well. This is because it is almost impossible to imagine the exact color when looking at the buckets in the grayscale image.

Figure 4.2: Case of Failure

4.4 Quantitative Evaluation

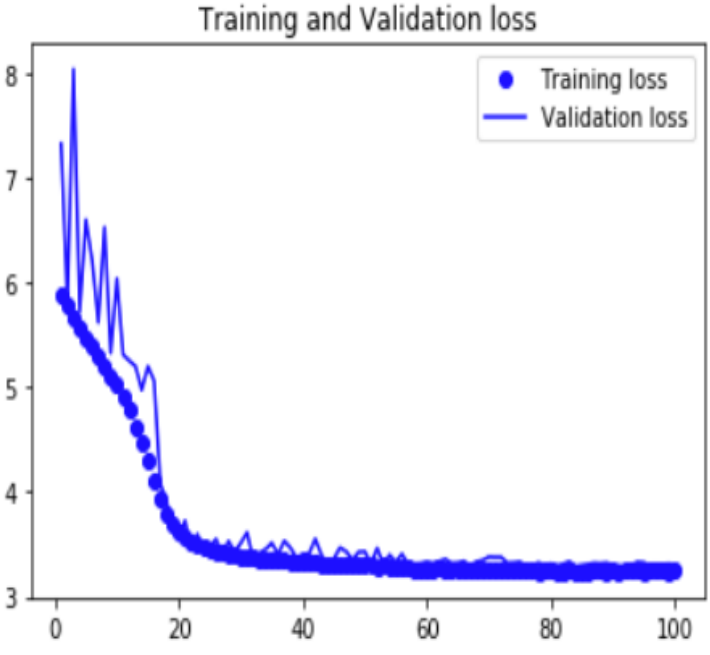


Figure 4.3: Training history on our proposed model

First of all, Figure 4.3. shows our training history. Our model converges well both on training and validation dataset. Therefore, our model trained

well.

Also, we use PSNR (Peak Signal-to-Noise Ratio) metrics for quantitative evaluation. PSNR is used to evaluate image quality loss information. This comes from calculating how much different between the ground truth image and the output image pixel by pixel. The equation for PSNR is as follows.

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}(\frac{MAX_i^2}{MSE}) \\ &= 20 \cdot \log_{10}(\frac{MAX_i}{\sqrt{MSE}}) \\ &= 20 \cdot \log_{10}(MAX_i) - 10 \cdot \log_{10}(MSE). \end{aligned} \tag{4.3}$$

PSNR metrics may be difficult to represent accurate performance in image colorization task. The reason is that the models in the field of image colorization are not aiming to find the exact color, but to find a plausible color. For example, if we are looking for the grayscale image of a tulip and the ground truth is yellow, but the colorization model can find it red. In this case, it cannot be said that the model find a completely wrong color. However, because the predicted image and Ground Truth are very different, PSNR is affected a lot. As such, PSNR cannot be a complete indicator for colorizing task. Nevertheless, it may contain important information to find pixels that predict different colors alone in the overall image or pixels that remain black and white because no color is found at all. And with PSNR, we can catch cases where the model is not properly trained and the result is predicted to be brownish or grayish. Therefore, PSNR metric is used in various colorization papers and we also use it.

Table 4.3. is the result of calculating the average PSNR after applying several models and our proposed model to the ImageNet test dataset. As mentioned earlier, 20,000 images are randomly selected and used. Table 4.3. shows that Zhang et al. [28] PSNR is 23.4657 and our proposed model's PSNR is 23.9219, which is higher. In other words, there is less loss compared to the ground truth at the result of our proposed architecture. In particular, the PSNR of Baldassarre et al. [2] using the MSE loss function was 21.9914, which was lower than that of other models. This is because the output color image

of Baldassarre et al. [2] model is grayish overall using MSE loss function.

Table 4.3: Comparing PSNR results with other networks.

Model	PSNR
Zhang et al. [28]	23.4657
Baldassarre et al. [2]	21.9914
Proposed method	23.9219

4.5 Legacy Old image Colorization

Since our model was trained based on the ImageNet dataset containing various images, we could apply images in various situations. In order to apply it to the actual situation, we searched and tested several old legacy images to see if it is possible to colorize using old black and white photos without ground truth as input, and how natural it is. Figure 4.4 shows some results.

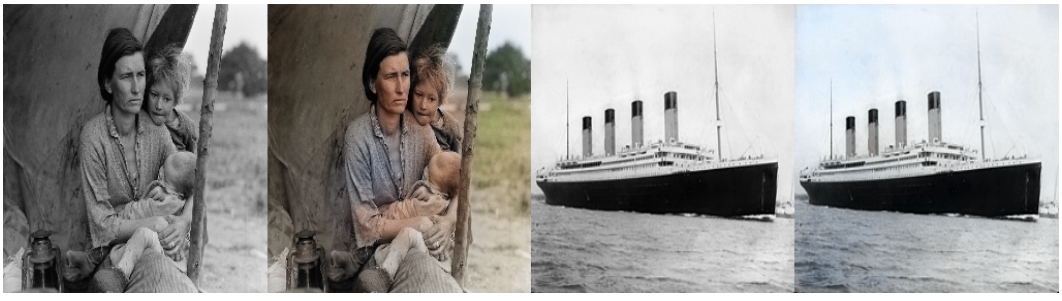


Figure 4.4: Old legacy image colorizing

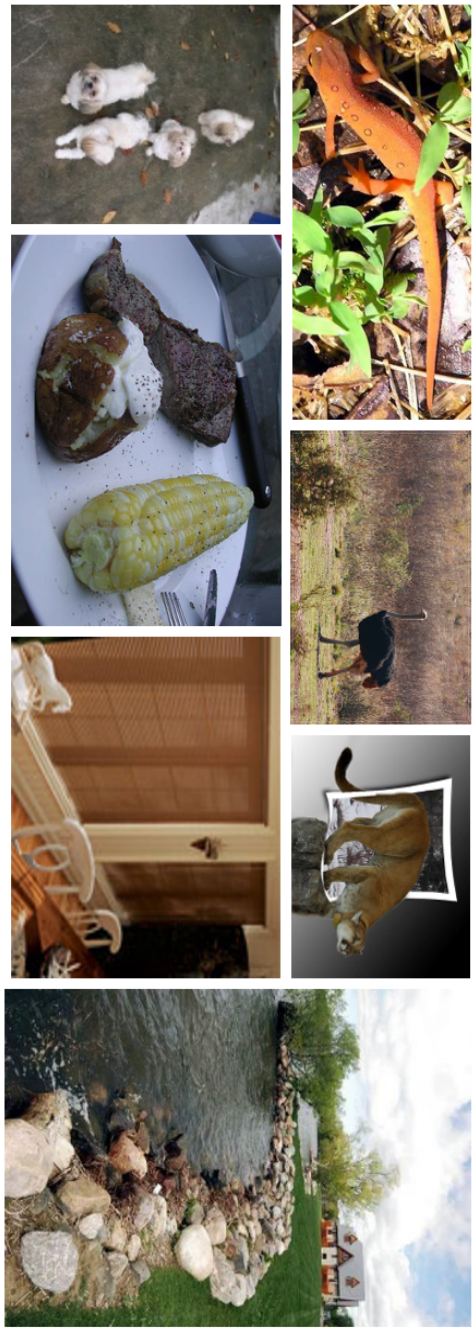
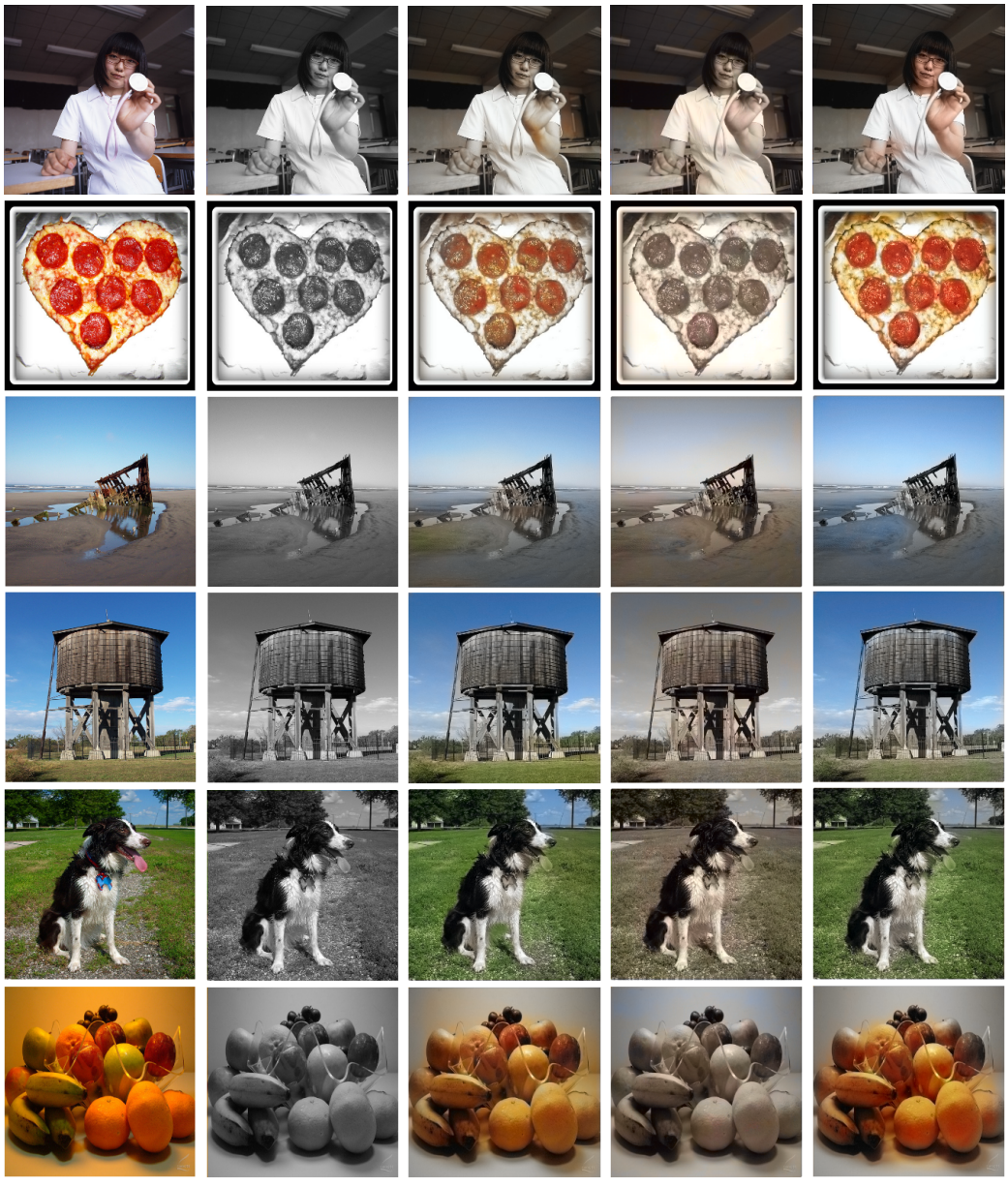


Figure 4.5: Some examples of ImageNet [21] dataset



(a) Original (b) Grayscale (c) Zhang (d) Baldassarre (e) Proposed

Figure 4.6: Qualitative Comparing results

Chapter 5

Conclusion

In this paper, a new Fully-automatic colorization model using Deep learning network was proposed based on FusionNet, which showed very good performance in the Encoder-Deocder models. Since we variously transformed FusionNet to fit the colorization task and used the appropriate loss function, good results were achieved. The validity of this model was compared with existing other deep learning models. The quantitative comparison to state-of-the-art methods show that our method outperforms them in terms of PSNR. Also, qualitative comparisons show visually high quality. Through this, it was confirmed that the proposed technique shows better performance than other deep learning models. Furthermore, our model itself may be put in the generator part of GAN to further increase the performance. Also, we can produce better performance by combining the features of the image (classification term, semantic information, segmentation information, etc.) in our model.

Bibliography

[1] H. Bahng, S. Yoo, W. Cho, D. Keetae Park, Z. Wu, X. Ma, and J. Choo, *Coloring with words: Guiding image colorization through text-based palette generation*, in Proceedings of the european conference on computer vision (eccv), 2018, pp. 431–447.

[2] F. Baldassarre, D. G. Morín, and L. Rodés-Guirao, *Deep koalarization: Image colorization using cnns and inception-resnet-v2*, arXiv preprint arXiv:1712.03400, (2017).

[3] A. Bugeau, V.-T. Ta, and N. Papadakis, *Variational exemplar-based image colorization*, IEEE Transactions on Image Processing, 23 (2013), pp. 298–307.

[4] Z. Cheng, Q. Yang, and B. Sheng, *Deep colorization*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 415–423.

[5] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin, *Semantic colorization with internet images*, ACM Transactions on Graphics (TOG), 30 (2011), pp. 1–8.

[6] A. Deshpande, J. Lu, M.-C. Yeh, M. Jin Chong, and D. Forsyth, *Learning diverse image colorization*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6837–6845.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Gen-*

erative adversarial nets, in Advances in neural information processing systems, 2014, pp. 2672–2680.

[8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, *Improved training of wasserstein gans*, in Advances in neural information processing systems, 2017, pp. 5767–5777.

[9] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[10] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, *An adaptive edge detection based colorization algorithm and its applications*, in Proceedings of the 13th annual ACM international conference on Multimedia, 2005, pp. 351–354.

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

[12] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, arXiv preprint arXiv:1502.03167, (2015).

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.

[14] S. Kang, J. Choo, and J. Chang, *Consistent comic colorization with pixel-wise background classification*, in NIPS'17 Workshop on Machine Learning for Creativity and Design, 2017.

[15] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114, (2013).

[16] A. Levin, D. Lischinski, and Y. Weiss, *Colorization using optimization*, in ACM SIGGRAPH 2004 Papers, 2004, pp. 689–694.

[17] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum, *Natural image colorization*, in Proceedings of the 18th Eurographics conference on Rendering Techniques, 2007, pp. 309–320.

[18] K. Nazeri, E. Ng, and M. Ebrahimi, *Image colorization using generative adversarial networks*, in International conference on articulated motion and deformable objects, Springer, 2018, pp. 85–94.

[19] T. M. Quan, D. G. Hildebrand, and W.-K. Jeong, *Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics*, arXiv preprint arXiv:1612.05360, (2016).

[20] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., *Imagenet large scale visual recognition challenge*, International journal of computer vision, 115 (2015), pp. 211–252.

[22] Y.-W. Tai, J. Jia, and C.-K. Tang, *Local color transfer via probabilistic segmentation by expectation-maximization*, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, IEEE, 2005, pp. 747–754.

[23] P. Vitoria, L. Raad, and C. Ballester, *Chromagan: Adversarial picture colorization with semantic class distribution*, in The IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 2445–2454.

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

[25] Y. Xiao, P. Zhou, and Y. Zheng, *Interactive deep colorization with simultaneous global and local inputs*, arXiv preprint arXiv:1801.09083, (2018).

[26] L. Yatziv and G. Sapiro, *Fast image and video colorization using chrominance blending*, IEEE transactions on image processing, 15 (2006), pp. 1120–1129.

[27] S. Yoo, H. Bahng, S. Chung, J. Lee, J. Chang, and J. Choo, *Coloring with limited data: Few-shot colorization via memory augmented networks*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11283–11292.

[28] R. Zhang, P. Isola, and A. A. Efros, *Colorful image colorization*, in European conference on computer vision, Springer, 2016, pp. 649–666.

[29] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, *Real-time user-guided image colorization with learned deep priors*, arXiv preprint arXiv:1705.02999, (2017).

[30] J. Zhao, J. Han, L. Shao, and C. G. Snoek, *Pixelated semantic colorization*, International Journal of Computer Vision, (2019), pp. 1–17.

[31] C. Zou, H. Mo, R. Du, X. Wu, C. Gao, and H. Fu, *Lucss: Language-based user-customized colourization of scene sketches*, arXiv preprint arXiv:1808.10544, (2018).

[32] C. Zou, H. Mo, C. Gao, R. Du, and H. Fu, *Language-based colorization of scene sketches*, ACM Transactions on Graphics (TOG), 38 (2019), pp. 1–16.

[33] C. Zou, Q. Yu, R. Du, H. Mo, Y.-Z. Song, T. Xiang, C. Gao, B. Chen, and H. Zhang, *Sketchyscene: Richly-annotated scene sketches*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 421–436.

국문초록

본 논문에서는 회색조 이미지들에 대한 채색 기법을 제안한다. 채색 작업은 크게 Scribble 기반 방법, Exemplar 기반 방법, 완전 자동 방법의 세 가지로 나눌 수 있다. 본 논문에서는 세 번째 방법을 사용했다. 최근에 채색 분야에서 널리 사용되는 딥 러닝 모델을 사용한다. Convolutional Neural Networks를 이용한 Encoder-Docoder 모델을 제안한다. 특히 기존에 image segmetation 분야에서 좋은 성능을 보이는 FusionNet을 자동 채색 목적에 맞게 다양한 방법으로 수정했다. 또한 더 나은 결과를 얻기 위해 MSE 손실 함수를 사용하지 않았다. 대신, 우리는 자동 채색 목적에 적합한 손실 함수를 사용하였다.

ImageNet 데이터셋의 부분 집합을 훈련, 검증 및 테스트 데이터셋으로 사용했다. 우리는 완전 자동 딥 러닝 방법에서 기존 방법을 가져와 우리의 모델과 비교했다. 우리의 알고리즘은 PSNR (Peak Signal-to-Noise Ratio)이라는 정량적 지표를 사용하여 평가되었다. 또한 결과를 정성적으로 평가하기 위해 테스트 데이터셋에 모델을 적용하여 다양한 모델과 비교했다. 그 결과 다른 모델에 비해 정성적으로도, 정량적으로도 좋은 성능을 보였다. 마지막으로 오래된 흑백 사진과 같은 다양한 유형의 이미지에 적용한 결과를 제시했다.

주요어휘: 딥러닝, 자동 채색, 합성곱 신경망, FusionNet
학번: 2018-21347

감사의 글

길었지만 짧게 느껴졌던 석사 과정을 마치고 학위 논문을 마무리하게 되었습니다. NCIA 랩실 안에서 그 동안 도와주신 많은 분들에게 감사의 말씀을 전합니다. 특히 랩실에 있는 동안 저에게 많은 도움을 주시고 다양한 조언을 해주셨던 강명주 지도 교수님 감사합니다. 또한, 갑작스럽게 부탁드렸지만 선뜻 심사위원장을 해주신 국웅 교수님, 바쁜 시간을 쪼개서 논문에 대한 조언을 해주시고 심사위원까지 해주신 하영수 박사님 감사합니다. 그리고, 랩실 안에서 저의 조언자도 되어주고 친구도 되어 주었던 수진언니에게도 감사하다는 말씀 전합니다. 저와 비슷한 시기에 랩실에 들어와서 서로 조언을 아끼지 않고, 이번 1년 동안 같이 졸업 준비를 한 정윤언니에게도 수고했고 고맙다고 말하고 싶습니다. 그리고 저와 분야도 다르고 랩실도 다르지만, 우연한 기회에 만나 제가 힘든 순간 항상 힘을 내게 도와 준 은실 언니도 너무 감사합니다. 마지막으로, 제가 어떤 길을 가던지 언제나 지지해주시는 부모님, 그리고 같이 살면서 다양한 도움을 주었던 저의 언니에게도 고맙습니다.

학위 논문을 쓰기 위해 걸었던 지난 1년, 사소한 실험 결과에 울고 웃던 저의 시간이 저를 성장시킨 거름이 되었습니다. 생각대로 잘 되던 날도, 잘 되지 않던 날도 있었습니다. 하지만, 학위 논문 주제를 잡는 처음 그 순간부터 혼자 모든 것을 다 하고 선택해야 했기에, 힘을 내서 여기까지 올 수 있었습니다. 대학원 생활을 하며, 딥러닝이라는 학문을 배운 것 뿐만 아니라 다양한 사람들을 만나고, 다양한 경험을 하면서 흔들리지 않고 바로 설 수 있는 방법을 배웠습니다. 그 3년이라는 시간이 헛되지 않게 배운 것에 더욱 정진하며 학문적으로도 인간적으로도 더욱 노력하는 사람이 되겠습니다.