COLORIZING GRAYSCALE PHOTOGRAPHS WITH CONVOLUTIONAL NEURAL NETWORK

Hanlin Ke, Boqian Fan & Kuan Yang
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{hkeab,bfanab,kyangao}@connect.ust.hk

ABSTRACT

Humans started to transform memory into photographs not until the invention of cameras. Photographs solve the problem that our memory is easy to lose and sometime may be unreliable, and they are also the best tool to capture the most unforgettable moment of our life. In old days, however, due to technological limitations, photographs are recorded in grayscale, which loses the information regarding colors. Therefore, our task is to colorize the grayscale photographs so that restores their original appearance to the greatest extent. We propose to use convolutional neural network to capture the features from the images with down-sampling technique and then apply deconvolutional layers to more advanced features with up-sampling technique. In general, the main task is to make the model learn the mapping relations between the lightness component of the input channel to the other two output color channels.

1 INTRODUCTION

Recently, a restored old video shot in 1920 introducing the Chinese fashion in time of Republic of China drew everyone's attention in video sharing websites (i.e. Bilibili). This high resolution video is claimed to be restored by AI. Afterwards, plenty of AI-restored old videos appeared in the internet. These old videos are usually in black and white tones, therefore every frames of the videos seems more or less similar to a grayscale photograph. We regard the process of restoring old videos as a problem of colorizing grayscale photographs. To colorize a grayscale photograph, the most important thing is to predict the color. Thus, the colorizing problem can be summarized as a color prediction problem.

Color prediction task has a cordial property that training data is practically free: in CIELAB color space, any colored photo can be used as a training example, simply by taking the image's L channel as input and its AB channels as the supervisory signal. Others have noted the easy availability of training data, and previous works have trained convolutional neural networks (CNNs) to predict color on large datasets (Zhang et al., 2016). In this paper, we follow the previous works and design a photograph colorizing model using the features of ResNet(He et al., 2016) to extract latent features of images' L channel then predicting the AB channel by up-sampling. In the experimental results, we manage to have nice performance similar to Zhang et al..

2 MOTIVATION

Due to the limitation of technology in old days, there has been plenty of grayscale photographs even though with the ability of capturing contents but lack of colors to enrich contents. More importantly, some of the photographs are keys for us to study the history. For instance, we may be interested in how various colors are used in old fashion and how blue the sky was.

Colorizing grayscale photographs is actually a difficult job because it requires not only certain background knowledge regarding the scene but also certain ability of imagination. There is not an absolute correct color for certain objects, and just like there are three main skin colors for humans such

as black, white, and yellow. To manually colorize grayscale photographs is both time-consuming and labor-consuming.

Thus, we are motivated to colorize the photographs supposed to be colorful while reducing human labors and time on colorizing the photographs. Inspired by DeOldify(Antic & Kelley), a famous GitHub repository that restores old photographs with high accuracy, we begin our research. However, the authors of DeOldify use NoGAN, a new and well-designed types of generative adversarial network (GAN) without publishing on any academic conferences or journals, as the main algorithm, we have no chance to reproduce a large system like DeOldify. Therefore, we look away to CNN which is less sophisticate than GAN. We do some researches on grayscale images colorization by CNN and find that there are already many successful researches done in computer vision community. Based on previous experience from Zhang et al. (2016), we successfully design our own CNN model to solve the problem mentioned in Introduction.

3 MODEL

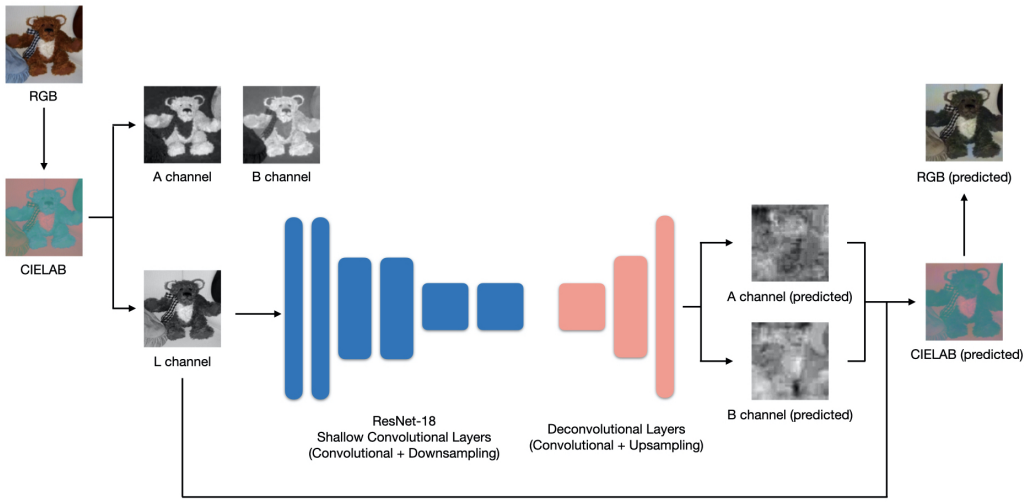

Figure 1: Overview of the proposed model: The L channel is the input of the neural network while A&B channel is the output. The model consists of a down-sampling part and a up-sampling part.

3.1 OVERVIEW

The overview of our model is shown above in Figure 1. In the pre-processing stage, where we build the training set and validation set, we firstly convert the images from RGB color space to CIELAB color space. As it is mentioned in the introduction section, CIELAB makes the color predicting process more smooth and convenient. Same as RGB images, CIELAB images contains 3 channels. The L channel means lightness, ranging from 0 (black) to 100 (white). The A channel ranges from red (above 0) to green (below 0), while B channel ranges from yellow (above 0) to blue (below 0). For display concern, all these three channels are shown in grayscale color map.

In this task, we take L channel from groundtruth as input, A&B channel are used to calculate the training loss and validation loss. Then, the input will be fed into a pre-trained ResNet-18 convolutional layers. In our model, we only extract the half of ResNet-18 model as the down-sampling part. The reason why we choose the shallow layers is that: since the input size is 224×224, the output size will be only 28×28 at the middle of the ResNet-18. If the down-sampling continues, the size will becomes less suitable for up-sampling. After we extract the latent features of the image in the down-sampling part, it is time to rebuild the color. We implement the up-sampling part by using the convolutional layers and up-sampling layers. Convolutional layers can guarantee the output is in 2 channels (i.e. A&B channels) by reducing the dimension of the feature map from down-sampling

part. After the A&B channels of prediction are generated from up-sampling part, the loss of the prediction will be computed. Note that we are new to the field of the image colorizaion. We simply choose the mean square error of A&B channels between the prediction and the groundtruth.

3.2 DOWN-SAMPLING

Before introducing the down-sampling technique, it is vital to introduce the architecture of the ResNet-18. The architecture of ResNet-18 is shown in Figure 2. As noticed, there are a total of eight layers; however, we only adopt the first three layers of the ResNet-18 (from conv_1 to conv3_x as depicted in Figure 2 in the column of Layer_Name) as the first half of our convolutional neural network. ResNet-18 is a pre-trained network which has been trained "on the set of images defined by the ILSVRC 2015 challenge" (Napoletano et al., 2018). The challenge is known for identifying the categories of object and scene described in a photograph (Napoletano et al., 2018). Considering its inherent capability of detecting categories of certain scene and object, we decide to apply it for the task of colorizing, which is a sort of transfer learning.

Down-sampling technique, in general, is to convert vectors from higher dimensions to lower dimensions. For example, from 128x128 to 64x64 with a scale factor of 2. Several popular operations could be applied to achieve it, such as average pooling and maximum pooling. In the context of ResNet18, maximum pooling is used as shown in Figure 2, which can be described as the operation shown in Figure 3.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 2: ResNet-18 Architecture (Napoletano et al., 2018)

3.3 UP-SAMPLING

Due to the ResNet-18 extracting main features from the grayscale input, the features are in smaller dimensions. In order to reconstruct vectors which still pertain same dimension as the input, the feature vectors are required to be expanded in two dimensions. Therefore, up-sampling, in the context of CNN, is a technique which expands a single value in a vector in two dimensions. As demonstrated below in the Figure 1, the original 1x1 dimension vector with an entry of 3 is expanded into a 2x2 dimension vector with four entries of 3.

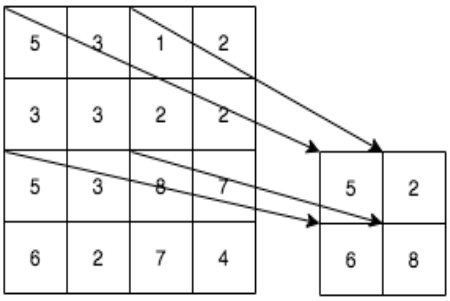

Figure 3: Down-sampling – 2x2 max pool, stride 2

We implement the up-sampling technique in PyTorch by simply calling the function "nn.Upsample(scale_factor=2)", which accomplishes exactly what the Figure 4 does.

Due to the properties of the up-sampling technique, the predicted result of the model is somehow similar to mosaics (shown in Figure 1).

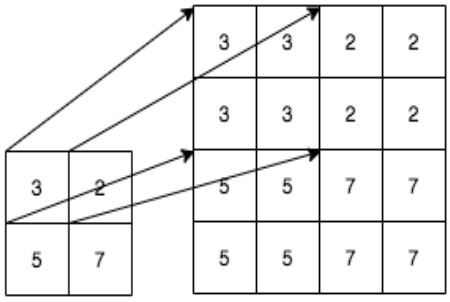

Figure 4: Up-sampling – 2x2 unpool, stride 2

4 DATASET

The dataset we use in this project is a subset of the MIT Places205 dataset which contain locations, landscapes, and buildings. The first 1000 images are used for validation, whereas the rest 40000 images are used for training. The images are 256×256 with three channels of RGB space, but we convert the color space of the images from RGB to CIELAB and the size of the images from 256×256 to 224×224.

5 EXPERIMENTAL RESULT

In the experiment, we compare our model with model from (Zhang et al., 2016) (we call it ModelZ in this section). We firstly trained our model for 100 epochs with Adam optimizer. We set the learning rate of our model at 0.01. It takes us for a few days to complete the training process. During the training process, we also discover that the loss of the training set is always fluctuating around 0.002 and 0.003 after 40 epochs. Despite the fact that our model seems to be not trained well, we encourage ourselves to have a comparison with the model that had been proposed on ECCV16. When we look into the code repository of (Zhang et al., 2016), we found that the authors trained their model for hundreds of epochs over a million images to get the best performance. Figure 5 is the experimental result of the model comparison with groundtruth images.

From image1 to image10, our model has a better outcome on landscapes, such as image1, image3 and image5. According to our dataset, which is mostly landscapes and other scenes of places, it is no surprising that our model can have such good outcome. Although some outputs of ModelZ can even

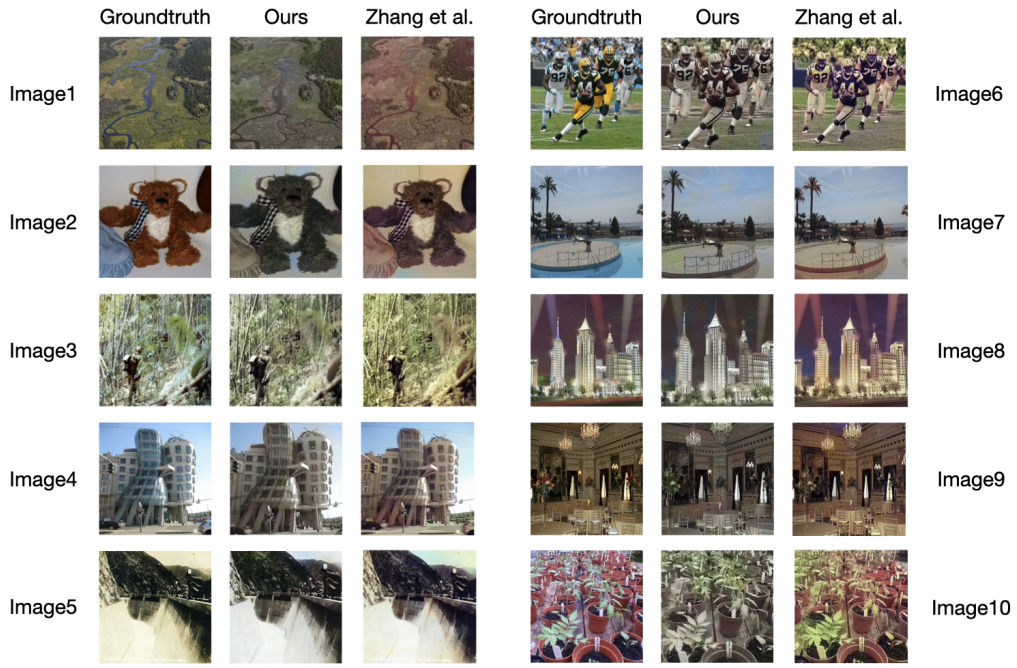

Figure 5: The experimental results of groundtruth and model comparison

fool our eyes like image8, image9 and image10, we are delighted to see the output of our model on image3 is almost the same with the groundtruth. It is worth to say that both our model and ModelZ are somehow precise on image4. Through these outputs, we find that ModelZ outputs its results with higher values on A channel that results in some reddish blur on images (i.e. image1, image2, image6 and image7). We also discover that some less-frequently appeared object, for example blue walls in image4, are not completely learned by the models. After all, we still satisfied with the output of our model.

6 Discussion

Figure 6: Some examples of the validation output from our model

Figure 6 shows some examples of colorized images. Obviously, compared with the landscapes (i.e. the first row), the effect of the algorithm is not satisfactory enough for other items such as architectures and people figures in the second row. In other words, for these images, the color is

not bright enough and the tone is dull. After analyzing this problem, we give a hypothesis that this phenomenon is attribute to Mean Squared Error(MSE) which we adopt as loss function. We adopt MSE since it can avoid making severe errors while determining colors. However, MSE will try to "average" the colors to get the minimum error, which will result in an "old", flat look.

Therefore, in the future work, one available direction is attempting other loss functions or considering this colorizing problem as multiple classification, which might be useful to solve dull color problem. The other direction is to try a more diverse dataset to improve the capability to paint other objects. Moreover, how to improve the efficiency of this model is also an unresolved problem since in our experimental it is really time-consuming to train the model until the output is good enough.

7 CONCLUSION

In this project, we have basically implemented the model of grayscale image colorization by deep learning. Firstly, we talked about our motivation and some background of image colorization. Then we introduced the model architecture and analyzed each layer in this model, especially convolutional neural network part. After that we finished programming work and carried out model experiment, whose output seems to be relatively successful. Finally we analyzed these results and give some future work directions to make improvement of this model.

Generally, although there are some aspects in this model need to be improved, this coloring model has basically met our expectations and it is capable to predict colors accurately to make those old photographs colorful again.

AUTHOR CONTRIBUTIONS

Hanlin Ke (20745412): Literature Review, Model Setup/Coding, Model Framework and Architect, Model Experiment, Video Presentation

Boqian Fan (20743139): Literature Review, Model Setup/Coding, Model Framework and Architect, Model Experiment, Video Presentation

Kuan Yang (20716605): Literature Review, Model Setup/Coding, Model Experiment, Video Presentation

REFERENCES

Jason Antic and Dana Kelley. Deoldify. `https://github.com/jantic/DeOldify`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly detection in nanofibrous materials by cnn-based self-similarity. pp. 6. Sensors MDPI, January 2018.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.