COLORIZE IMAGES WITH DEEP LEARNING

PROJECT 30

LIANCHENG GONG [GONGLC@SEAS.UPENN.EDU],
ZHIHAO CHEN [CZHCHEN@SEAS.UPENN.EDU],
JINYUN SHAN [CECISHAN@SEAS.UPENN.EDU],

ABSTRACT. This paper explores three different deep learning networks: ResNet, GAN, VAE to solve the problem of colorize greyscale images. We use the grey-scaled views as the input, and compare the colorized images produced by our models with the ground truth. The overall performances for all three models are great. It is hard to tell the generated images are fake although there are some difference between the ground truth and the output. ResNet and VAE generate less colorful images, and some images processed by GAN are overly colorful.

1. INTRODUCTION

The advancement of modern camera technology has enabled the capture and preservation of a wide range of colors in contemporary photographs. However, numerous black and white photos taken in the past, which do not contain color information, still exist. In an effort to restore the original colors of these historical images, the use of deep learning tools and principles has been proposed as a potential solution. Through the application of various deep learning techniques, it may be possible to colorize these black and white photos, thereby increasing their visual appeal and enhancing their historical value.

2. BACKGROUND

The problem of colorizing black and white photographs can be formalized as follows: given a grayscale image, denoted by I_{bw}, the goal is to predict a corresponding color image, denoted by I_{color}. The predicted image should accurately represent the original colors of the scene, as well as adhere to the inherent structure and layout of the image.

To tackle this problem, a number of deep learning approaches have been proposed, including the use of generative adversarial networks (GANs), convolutional neural networks (CNNs), autoencoders, and image translation techniques. These approaches typically involve the training of a model on a large dataset of black and white and color images, and the use of various optimization algorithms to minimize the difference between the predicted and ground truth color images. In this report, we use residual connection ResNet, generative adversarial networks, and autoencoders to reach our target.

In general, the colorization of black and white photographs is a challenging task due to the inherent ambiguity and subjectivity of color. There are often multiple valid colorizations of a given black and white image, depending on the specific colors and hues that are chosen. As such, the performance of a colorization model can be evaluated using a variety of metrics, such as the mean squared error (MSE) between the predicted and ground truth color images, the structural similarity index (SSIM) between the two images, and the perceptual similarity between the predicted and ground truth images as judged by human evaluators.

3. RELATED WORK

Cheng el at.[4] trained CNN to predict color while the results looked desaturated. Koleini el at.[2] use a Gabor filter bank to extract the texture features and replace CNN with an artificial neural network (ANN). The paper published by Zhang[1] uses the VGG with added depth and dilated convolutions as the basic model. They also tried to recognize the semantics of the scene and its surface textures. Hattab et at.[3] introduced ten simple rules to colorize the images including color space, color context, color deficiencies, etc. We can make more attempts on how to create color effectively.

4. Approach

4.1. **Data.**

The data we used are the Intel Imgae from Kaggle, which contains around 25k images including 6 categories: buildings, forest, glacier, mountain, sea, and street. We changed the original images into greyscale as the input to our model. In the rest of this section, we tried three different deep learning architectures on the data and we want to see how each method works.

4.2. **ResNet.**

ResNet (short for Residual Network) is a type of convolutional neural network (CNN) that was developed by researchers at Microsoft Research in 2015. It's made to solve the vanishing gradient problem, which affects very deep neural networks frequently and causes the signal to become weaker and weaker as it moves through the layers. The concept of residual connections, which ResNet uses to address this problem, enables the network to "skip" over some layers and gain direct access to earlier layers. By doing so, the network can learn more quickly and avoid the vanishing gradient issue. The figure below shows its basic structure.

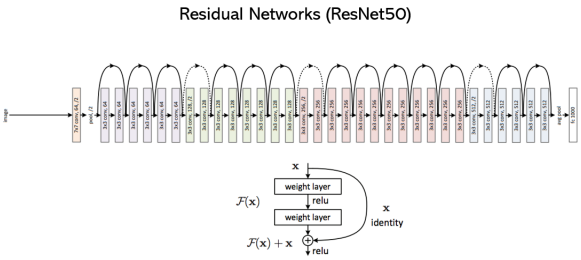
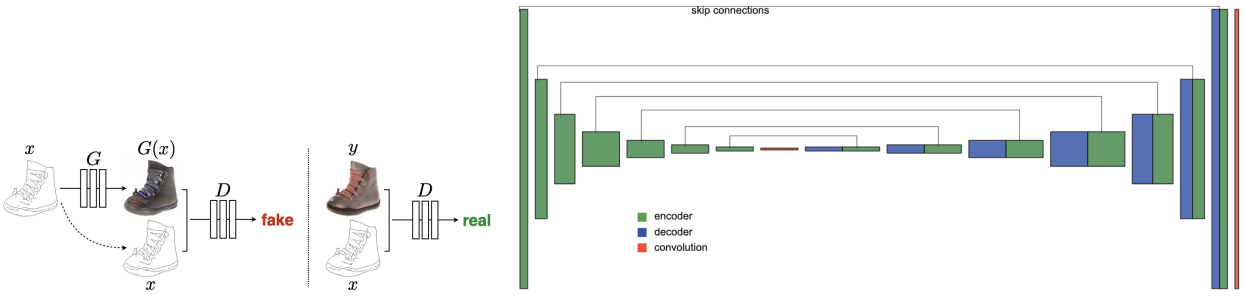


FIGURE 1. ResNet50 architecture

ResNets have demonstrated success in a variety of computer vision tasks, such as semantic segmentation, object detection, and image classification. This implies that they might also be useful for processes like image colorization.

4.3. **Generative Adversarial Network (GAN).**

GAN is a kind of models that generate a mapping from random noise to an output image. In this project, we are using a conditional GAN which is introduced by Isola et at. [5] which uses a generated image and random noise to predict an output image. Figure 1A shows how this conditional GAN works: a generator G will learn how to fool the discriminator, and the discriminator D learns how to classify between the real photo and the synthesized fake photo.



(A) Conditional GAN

(B) U-net

FIGURE 2

Our model uses a U-net structure show in figure 1B, where we add some skip connections between the ith encoder with the (n-i)th decoder. In this problem of the colorization of black and white images, the input and output shares some low-level information like edges, therefore, the U-net allows these kind of information to go across the network quickly.

4.4. **VAE.**
Autoencoders are a type of neural network that are commonly used for tasks such as image reconstruction and dimensionality reduction. They consist of two parts: an encoder and a decoder. The encoder takes an input image and converts it into a lower-dimensional representation, known as the latent representation or code. The decoder then takes the latent representation and converts it back into an output image that should be similar to the input image.

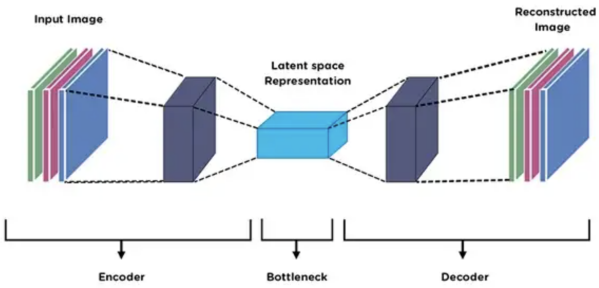


FIGURE 3. VAE architecture

The first 6 layers of resent 18 are used for the encoder part which transforms the input image into a hidden representation. Then in the decoder part, the hidden representation is upsampled back to the original input size but this time with 3 channels that is R-G-B. The loss function used while training the model is Mean squared error.

5. EXPERIMENTAL RESULTS

5.1. **ResNet.**
The downsampling method used by the original ResNet50 is inappropriate for our proposal. As a result, we modify the original structure in some ways. We cannot use the weights that have already been trained, which is a drawback. All of the max pooling layers are removed, and the stride in all of the convolutional layers is changed to 1. In addition, the input layer is changed to 1, and the convolutional layer with output channel 3 replaces the final fully connected layer for classification. Finally, we reduce the number of channels in the residual modules to reduce the number of network weights and save on computational and memory costs.

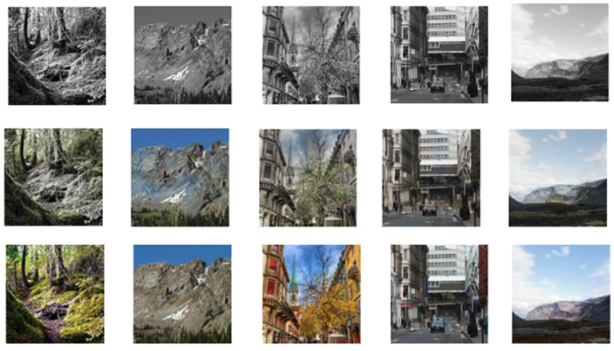


FIGURE 4. ResNet Predictions

The above figure is the output after 20 epochs training of our model. The first row is the greyscale image, the second row is the synthesized image, and the last row is the original image. The overall performance is satisfactory, but we can see that synthesized images are less colorful than the original ones. The validation loss decreases steadily here, and it is likely to drop with further training.

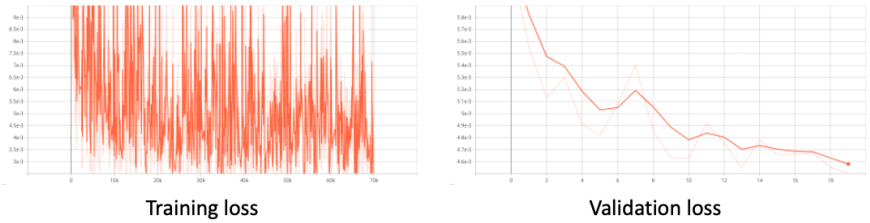


FIGURE 5. ResNet Loss

5.2. **GAN.**
Our conditional GAN model with a U-net architechture consists of 7 encoders and 7 decoders, each consists of a 4*4 kernel convolutional 2d layer, a batch normalization, and a Leaky ReLU nonlinearity layer. There are also two convolutional layers at the end of the encoder and decoder for the output.

FIGURE 6. GAN Predictions

The above figure is the output after 20 epochs training of our model. The first row is the greyscale image, the second row is the synthesized image, and the last row is the original image. We can see that most of the predictions are good and close to the real image, for example the blue sky, the green trees, and the yellow buuilding. However, we notice that there still exist several part may be too colorful or mis-predicted. For example, the building in the first image might be too yellow, the sunlight spot in the third image becomes red, and the sunlight in the fourth image shows white instead of golden color. These might due to some overfitting on colors, or underfitting for the sunlights, and all of these might due too the lack of images and training epochs. Due to the time and computational resource limitation, we here only trained for 20 epochs, but if we have time we will definitly use more data and train for longer time.

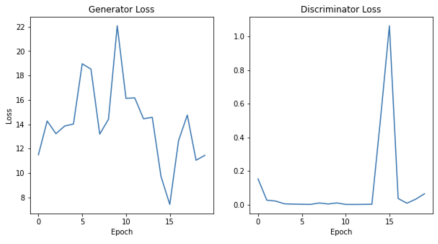


FIGURE 7. GAN Loss

We recorded the generator losses and discriminator losses in the above figure. We notice that the generator loss is unstable because of the adversarial network. And in the discriminator losses there's a sudden peak at around epoch 15 which correspond to the low point in generator loss. We know that an improvement in the generator will cause more error in the discriminator and vice versa, which can be shown in the plot.

5.3. **VAE.**
To apply VAE, we do the data preprocessing including applying a center crop transform to a fixed size of 128x128, converting the image to the LAB color space using the rgb2lab function from the skimage.color library, and normalizing the image to the range (-1, 1). The model consists of a pre-trained ResNet-18 model, which is a convolutional neural network (CNN) trained on the ImageNet dataset, and an upsampling network that takes the features extracted by the ResNet model and upsamples them to generate the colorized output.

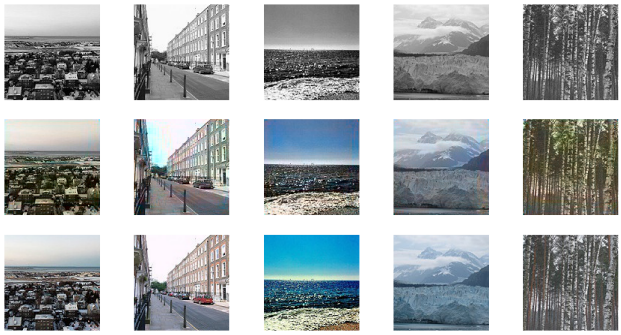


FIGURE 8. VAE Predictions

The above figure is the output after 20 epochs of training of model VAE. The first row is the greyscale image, the second is the synthesized image, and the last is the original image. The colorized images have a satisfactory result that the blue sky, the green trees, and the yellow building are predicted well. However, there are some misclassified pixels, where the generated color does not accurately match the color that would have been present in the original image. This can occur if the VAE is not trained on a diverse enough dataset, or if the latent space of the VAE does not contain sufficient information to accurately reconstruct the original image.

The MSE loss from the training weights updated and validation epoch updates decreases sharply from 0.5 to 0.002 in a shorter time, which means that the VAE is more effective. However, VAEs can be sensitive to the choice of hyperparameters, such as the size of the latent space and the learning rate. If these hyperparameters are not set correctly, the VAE may produce poor-quality results or may not converge during training.

6. DISCUSSION

ResNet is typically not the primary model used for this task. GAN and VAE are both generative models that have been widely used for image colorization, among other tasks. GAN can provide colorful images, but the quality varies because of the adversarial nature of the generator and discriminator, and sometimes the synthesized images are overly colorful.

For all the three methods, the synthesized color is still different from the real color, partly because we only trained for 20 epochs due to the time and computational resource limitations. All of the three losses will still decrease after that, so if we have more time and take good use of the AWS GPUs, we would definitely train for more than 100 epochs and compare the performance when they converges. When we are reading the related papers, we also found many other interesting models, for example we could incorporate self-attention networks.

REFERENCES

[1] Zhang, R., Isola, P.; Efros, A. A. (2016, October 5). Colorful image colorization. arXiv.org. Retrieved November 11, 2022, from https://arxiv.org/abs/1603.08511

[2] Mina Koleini, S. Amirhassan Monadjemi, Payman Moallem (2010) Automatic black and white film colorization using texture features and artificial neural networks, Journal of the Chinese Institute of Engineers, 33:7, 1049-1057, DOI: 10.1080/02533839.2010.9671693

[3] Hattab, G., Rhyne, T.-M.; Heider, D. (2020, October 15). Ten simple rules to colorize biological data visualization. PLoS computational biology. Retrieved November 11, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7561171/

[4] Z. Cheng, Q. Yang and B. Sheng, "Deep Colorization," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 415-423, doi: 10.1109/ICCV.2015.55.

[5] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[6] Zhang, Richard, et al. "Colorful Image Colorization." ArXiv.org, 5 Oct. 2016, http://arxiv.org/abs/1603.08511.

[7] Kingma, Diederik P, and Max Welling. "Auto-Encoding Variational Bayes." ArXiv.org, 10 Dec. 2022, http://arxiv.org/abs/1312.6114.