# GAN based colorization project technical report

Yuantao Bai

April 8,2022

## Contents

# 1 Introduction:

In this project we trained a GAN-based colorizer which takes a grayscale photograph as input, and generate a plausible color version of the photograph. In addition, we deploy the final algorithm model on AWS and use the Heroku platform to make the algorithm run on the web.

# 2 Motivation

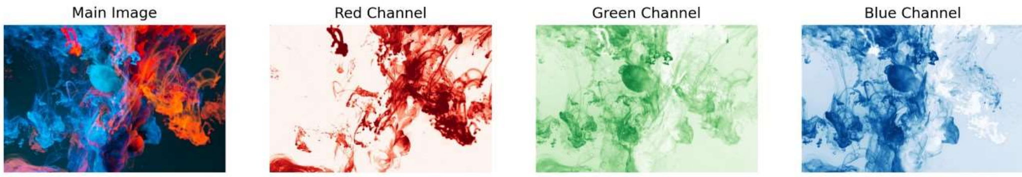
The main purpose of the GAN colorizer is to help people get the fun of painting. Although humans are very sensitive to color perception, it is relatively difficult to express color correctly. People often have to go through a lot of training to describe reality through painting relatively realistically, and this process is painful for most people. We wanted to implement a colorizer that would allow people to realize the great potential of their own pencil sketches.

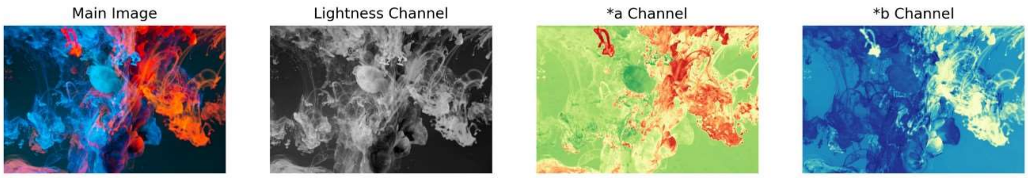
# 3 Dataset

Getting enough data of peoples' pencil drawing and the colorized version of the drawing is not reasonable as few people tend to share their draft of painting. However, we can use grayscale images to simulate the pencil drawing and use the original image as the colorized version. These method leads to a nice property that training data is practically free: any color 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. In this project, we used ImageNet as our dataset.

# 4 Preprocessing

In this work we utilize the L*a*b* color space for the colorization task. This is because L*a*b* color space contains a dedicated channel to depict the brightness of the image (which is highly related to grayscale value) and the color information is fully encoded in the remaining two channels. Using Lab color space prevents any sudden variations in both color and brightness through small perturbations in intensity values that are experienced through RGB. Visualization of RGB and Lab color space is as follows:




The translation from RGB color space to Lab color space needs to be done with the help of the XYZ color space, in which X, Y, Z describe the color stimulus considered and Xn, Yn, Zn describe a specified white achromatic reference illuminant. for the CIE-1931-2 standard colorimetric observer and assuming normalization where reference white = Y = 100. For Standard Illuminant D65:

$$
\begin{aligned}
X_{\mathrm{n}} &= 95.0489, \\
Y_{\mathrm{n}} &= 100, \\
Z_{\mathrm{n}} &= 108.8840
\end{aligned}
$$

$$
\begin{aligned}
X &= X_{\mathrm{n}} f^{-1}\left(\frac{L^\star + 16}{116} + \frac{a^\star}{500}\right) \\
Y &= Y_{\mathrm{n}} f^{-1}\left(\frac{L^\star + 16}{116}\right) \\
Z &= Z_{\mathrm{n}} f^{-1}\left(\frac{L^\star + 16}{116} - \frac{b^\star}{200}\right)
\end{aligned}
$$

$$
\begin{aligned}
L^\star &= 116\, f\left(\frac{Y}{Y_{\mathrm{n}}}\right) - 16 \\
a^\star &= 500\left(f\left(\frac{X}{X_{\mathrm{n}}}\right) - f\left(\frac{Y}{Y_{\mathrm{n}}}\right)\right) \\
b^\star &= 200\left(f\left(\frac{Y}{Y_{\mathrm{n}}}\right) - f\left(\frac{Z}{Z_{\mathrm{n}}}\right)\right)
\end{aligned}
$$

## 5 Postprocessing

The calculation of Gan colorizer is completely based on the Lab color space, so the color space conversion needs to be performed when the model is finally used to generate a color image. The transformation from Lab to RGB is contrary to preprocessing:

$$
\begin{aligned}
X &= X_{\mathrm{n}} f^{-1}\left(\frac{L^\star + 16}{116} + \frac{a^\star}{500}\right) \\
Y &= Y_{\mathrm{n}} f^{-1}\left(\frac{L^\star + 16}{116}\right) \\
Z &= Z_{\mathrm{n}} f^{-1}\left(\frac{L^\star + 16}{116} - \frac{b^\star}{200}\right)
\end{aligned}
$$

In which:

$$
f^{-1}(t) = \begin{cases} t^3 & \text{if } t > \delta \\ 3\delta^2\left(t - \frac{4}{29}\right) & \text{otherwise} \end{cases}
$$

# 6 GAN Model

Image colorization is an image-to-image translation problem that maps a high dimensional input to a high dimensional output. It can be seen as a pixel-wise regression problem where structure in the input is highly aligned with structure in the output. That means the network needs not only to generate an output with the same spatial dimension as the input, but also to provide color information to each pixel in the grayscale input image.

In 2014, Goodfellow et al. [1] proposed a new type of generative model: generative adversarial networks (GANs). A GAN is composed of two smaller networks called the generator(G) and discriminator(D). The generator produces results that are indistinguishable from real data while the discriminator classifies whether a sample is fake, i.e., generated by the generator's model or real, i.e., came from the real data. Both of these subnetworks are trained simultaneously until the generator is able to consistently produce results that the discriminator cannot classify.

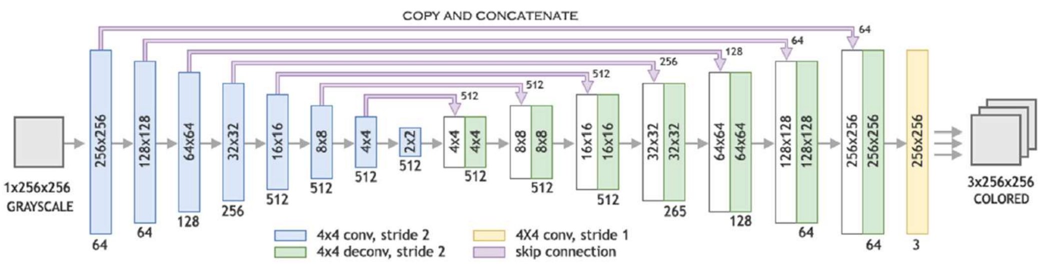
In a traditional GAN, the input of the generator is randomly generated noise data z. However, in the colorization problem, the colors are not randomly generated but generated under the guidance of the L channel. This problem was addressed by using a variant of GAN called conditional generative adversarial networks [PIXCOLOR: PIXEL RECURSIVE COLORIZATION]. Since no noise is introduced, the input of the generator is treated as zero noise with the grayscale input as a prior, or mathematically speaking, G(0z|x). In addition, the input of the discriminator was also modified to accommodate for the conditional network. By introducing these modifications, our final cost functions are as follows:

$$\min_{\theta_G} J^{(G)}(\theta_D, \theta_G) = \min_{\theta_G} -\mathbb{E}_z \left[\log(D(G(\mathbf{0}_z|x)))\right] + \lambda \|G(\mathbf{0}_z|x) - y\|_1$$

$$\max_{\theta_D} J^{(D)}(\theta_D, \theta_G) = \max_{\theta_D} \left(\mathbb{E}_y \left[\log(D(y|x))\right] + \mathbb{E}_z \left[\log(1 - D(G(\mathbf{0}_z|x)|x))\right]\right)$$

In which D is the discriminator, G is the generator.

In this project, we follow the structure of U-Net to train our GAN model, layers of U-Net is shown as follow:



# 7 GAN Colorizer Result:

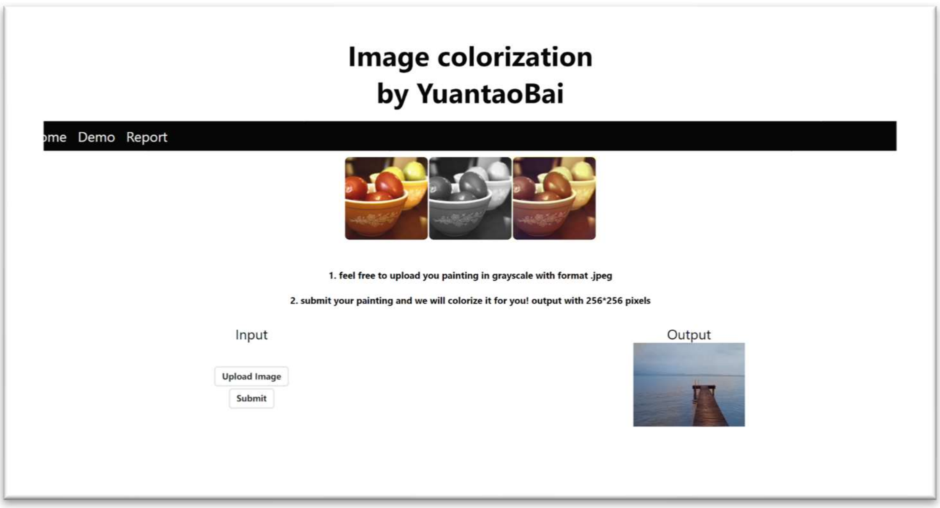
The algorithm takes a grayscale image in jpeg format of any resolution as input (no preprocessing is required, the algorithm can process the three-channel image by itself), and generates a png format image that is colored according to the grayscale image. The default resolution is 256*256

Original image grayscale image colorized image

# 8 Deployment

In this project, we deployed GAN colorizer in the AWS cloud environment, and interactively displayed the algorithm on the personal website through the API provided by it. The specific steps are as follows:

1. Export the onnx inference model from the pytorch environment and use numpy to reconstruct the color space conversion
2. Deploy the model and algorithm to AWS lambda function
3. Create API to allow external users to use lambda for forward inference
4. Deploy a website on heroku, allowing users to upload their own images and call the API to colorize the images



# 9 Reference


[1]Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." *European conference on computer vision*. Springer, Cham, 2016.

[2]Guadarrama, Sergio, et al. "Pixcolor: Pixel recursive colorization." *arXiv preprint arXiv:1705.07208* (2017).

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

[4]Nazeri, Kamyar, Eric Ng, and Mehran Ebrahimi. "Image colorization using generative adversarial networks." *International conference on articulated motion and deformable objects*. Springer, Cham, 2018.