Grayscale Image Colorization

Yanbaihui Liu
EECS 605 Project

Problem Statement

This project aims to colorize the grayscale images and deploy the live demo via aws. An intuitive way is to use L2 Loss. However, simply using the L2 norm between the real image and the generated image is not robust to the inherent ambiguity and multi-model of the coloring problem and may average out colors and finally give a greyish result as shown in Figure 1. That's why this project makes the colorization problem a multinomial classification problem and use Cross Entropy Loss.

9 Methods

9.1 Algorithms

Differ than RGB, this project actually use LAB color space, where the A-channel contains green-red components, B-channel contains the blue-yellow component, and L-channel encodes the intensity information. The L channel is the input to the model to predict the AB channel. Specifically, the ab color space is divided into 313 bins of size 10 shown in Figure 2. Suppose X denotes the input, and Y denotes the output. Set up a ConvNet to map the lightness of a given image into $\hat{Z} \in [0,1]^{H\times W\times Q}$, so for each $H \times W$ pixels, $\hat{Z}$ contains the probability that a pixel belongs to that bin. Each ground truth Y can be encoded as a 1-hot vector Z by searching for the 5-nearest quantized ab bins. Then, take the class rebalancing version cross entropy, which defined as

$$\mathcal{L}(\mathcal{Z}, \hat{\mathcal{Z}}) = -\frac{v}{|HW|}\sum_{h,w}\sum_{q} Z_{h,w,q} log(\hat{Z}_{h,w,q}) \tag{1}$$

to encourage the learning of rare colors. $\hat{Y} = H(\hat{Z})$ is used to maps the predicted distribution $\hat{Z}$ to point estimate $\hat{Y}$ in AB space. Last, then concatenate with L with predicted AB to get a possible colorful result. The network architecture is shown in Figure 3.

9.2 Architecture on Cloud

The pipeline of deployment on cloud is shown in Figure 4.

10 Experiments and Results

Considering the computational limit of this project, I train our network on 2000 images from a subset of Intel Image Classification Dataset,and test on a few separate images in the validation set.

The training and testing images are in size of 150 $\times$ 150. The batch size is set to 32 and number of epochs is set to 30. I use a self-adaptive learning rate that starts from 0.001 and decays to $\frac{1}{10}$ of the previous every time the difference of loss is less than 10^{-4}. For color rebalancing, in another word, mitigating the biased distribution typically caused by image backgrounds, the temperature is set to 0.38.

Figure 5 shows the trends of our training loss. Since we use a small dataset with only two categories, the objective function converge as soon as approximately 20 epochs. Some good sample predicted results are shown in Figure 6, but I also include the case where it doesn't work well in Figure 7. To quantitatively see the difference, I adopt Peak signal to noise ratio (PSNR) and structural index similarity (SSIM) as evaluation metrics, and the final results are the average value of three images pairs, as shwon in Table1. I also add the results generated by L2 Loss as a baseline. By looking at the table, not only the good samples, but even the bad results generated by this method have better performance than the baseline.

11 Challenges

One challenge that I faced when developing the project is that the initial model and packages are too big to be triggered by Aws Lambda. The way I figure it out is by converting the Tensorflow model to an ONNX model, so I don't need to include the pretty large Tensorflow package in the module to Lambda. An remaining challenge is the model now doesn't restore some warm-toned images well. One possible reason is, due to the limitation of resources, I only use small portion of the dataset that mostly contains glaciers and mountains, which means the model may only learn some blueish or low saturation color.

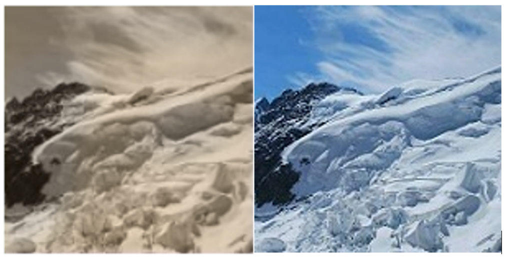

Figure 1: The Predicted Result Using Simple MSE and its True Color.

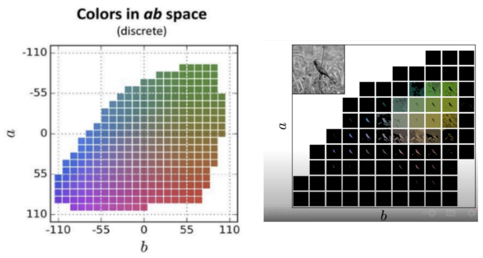

Figure 2: Left: Quantized colors in ab channel. Right: The more brightness in some regions means more probabilities the colors are belong to that bin.

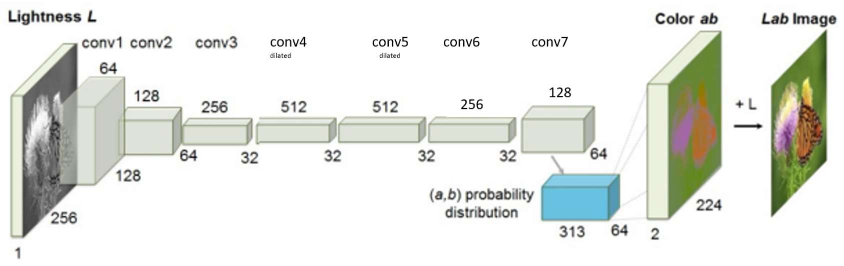

Figure 3: Network architecture.

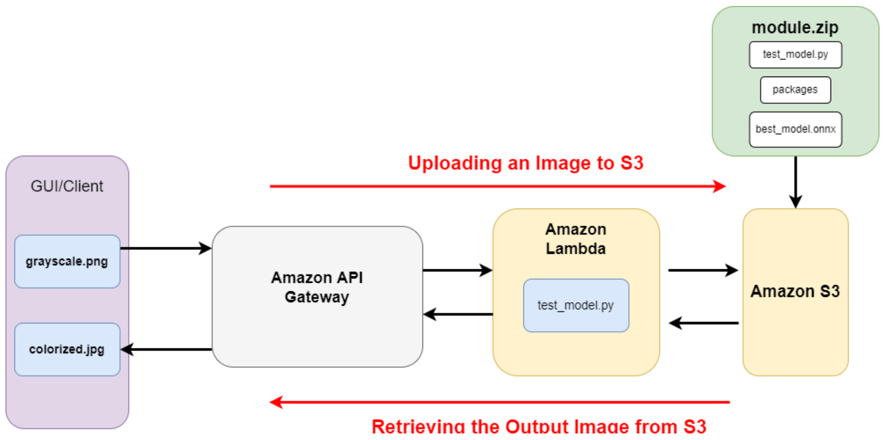

Figure 4: Pipeline Data Flow Diagram on Cloud.

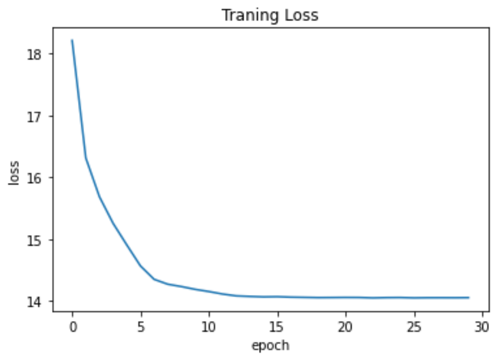

Figure 5: Training Loss.

Table 1: Comparison Metrics

	Baseline	Our Good	Our Bad
psnr	17.325	27.663	18.688
ssim	0.836	0.966	0.875

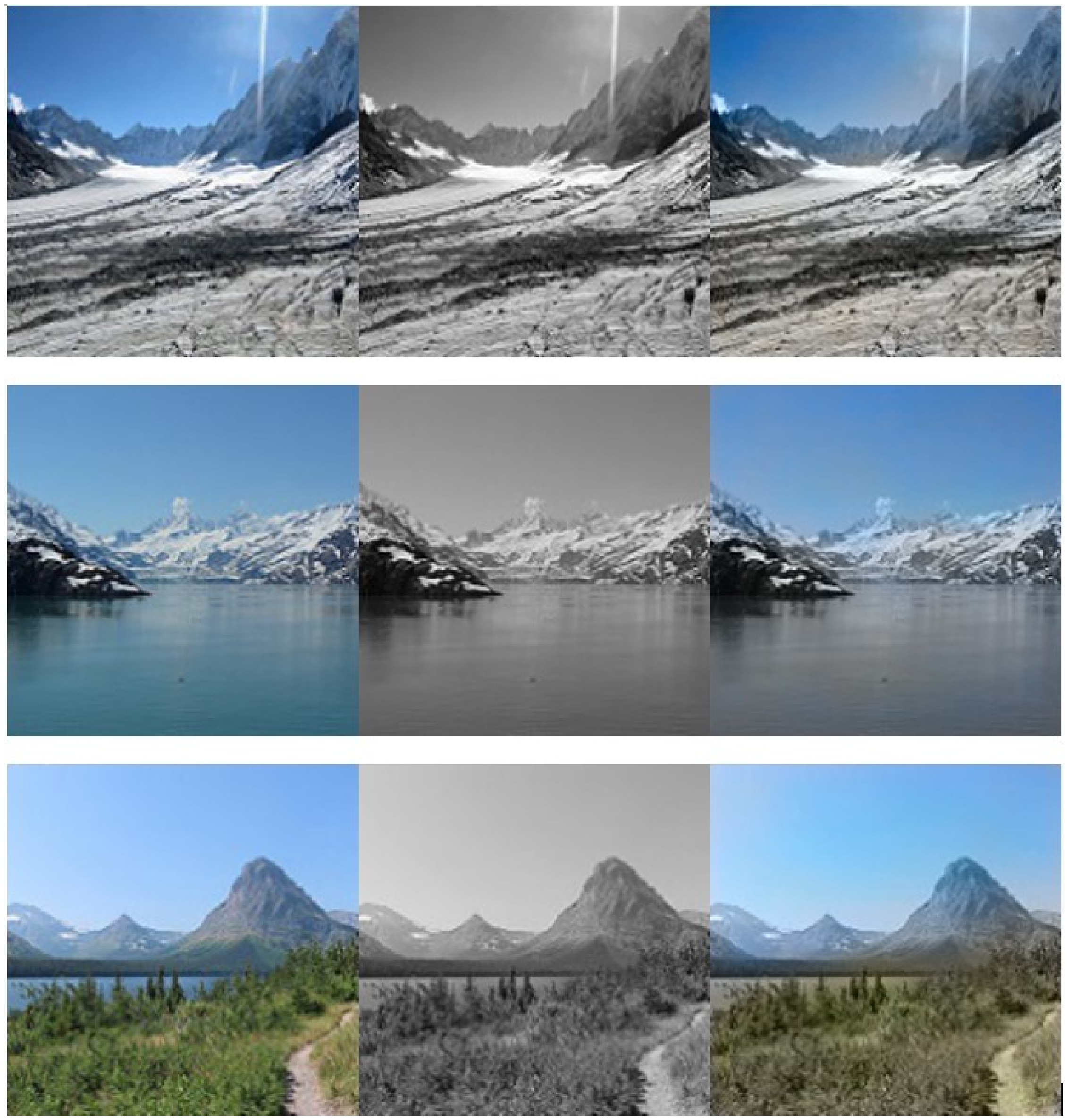

Figure 6: Sample Good Colorized Results. First column is the true color, second column is gray-scaled images, and last column is the predicted colorized results.

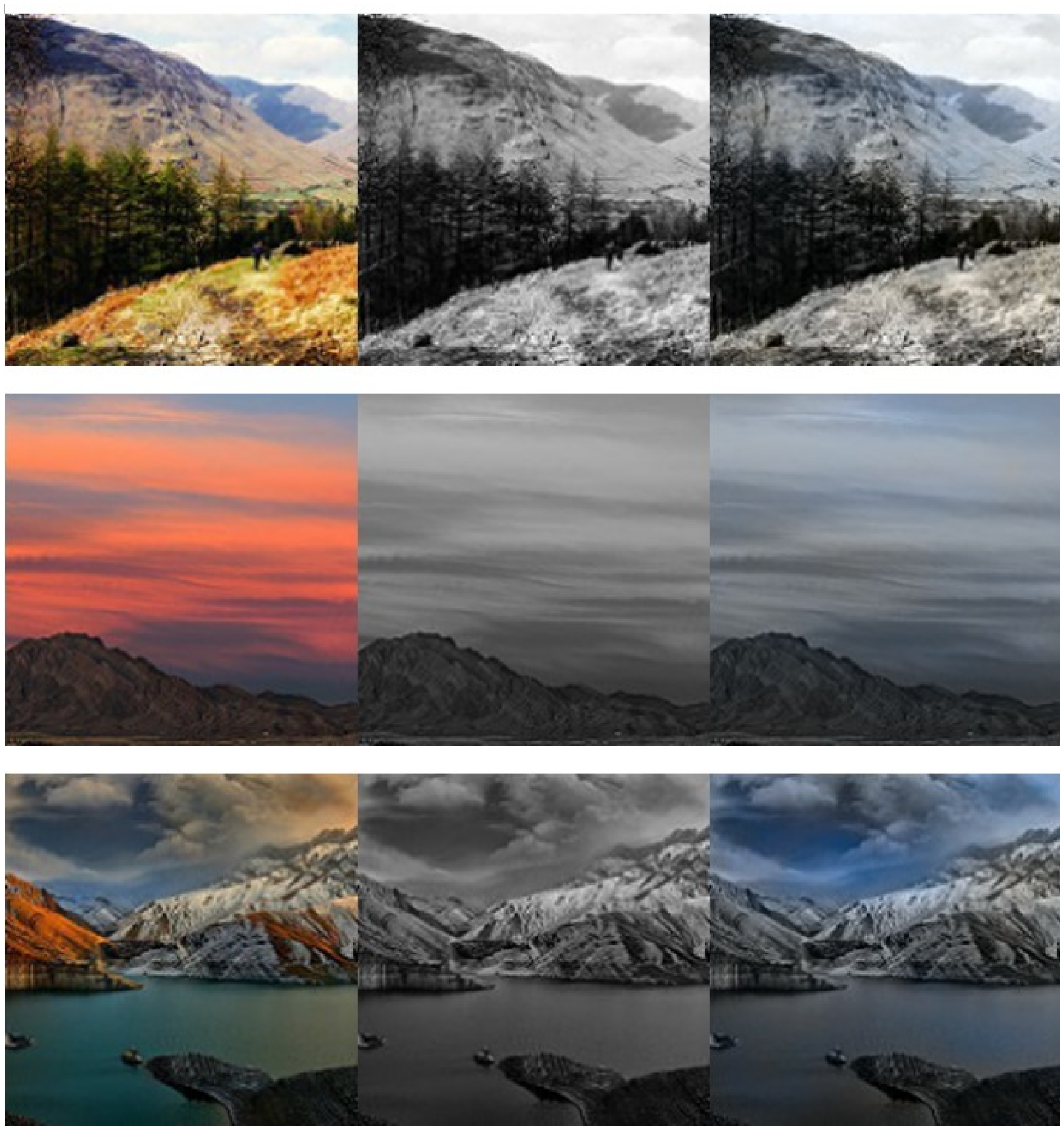

Figure 7: Bad Results. First column is the true color, second column is gray-scaled images, and last column is the predicted colorized results.