Black and White Image Colorization

Presented By

1605105 - Mashrur Ahmed Yafi

1605107 - Mehdi Hassan Akash

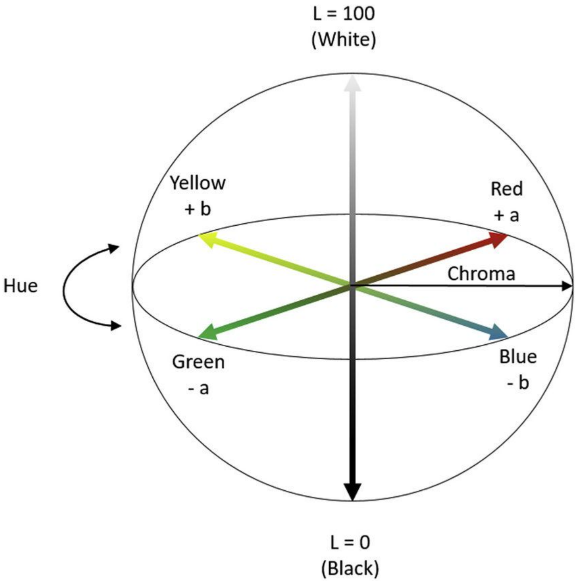
1605118 - Habibul Islam

Problem Definition

- Model takes grayscale (black and white) image as input
- Predicts colored image using the input image
- Our target is to get respectable color for output image

Method

- The network will take **L** channel as input and predict **a** and **b** channel.
- Then colored images will generated using the **L** channel and the predicted **ab** channel.



Dataset and resources

- We have used our synthetic dataset of 50 thousands images, which was made programmatically using a subset of imagenet.
- 25k of 50k were chosen randomly to train the model.
- We have taken colored images and separated L, A, B channels and kept in our dataset.
- We were motivated by the paper ‘**Colorful Image Colorization’**(2016**)** by Richard Zhang, Phillip Isola, Alexei A. Efros from University of California, Berkeley.

Challenges

- No activation functions were working good for predicting values in this range.
- If we tried to output the values between 0 and 255 using ReLU with clipping, most of the values were near 255.
- If we tried to output the values between 0 and 1 using sigmoid, most of the values were either 0 or 1.

Challenges

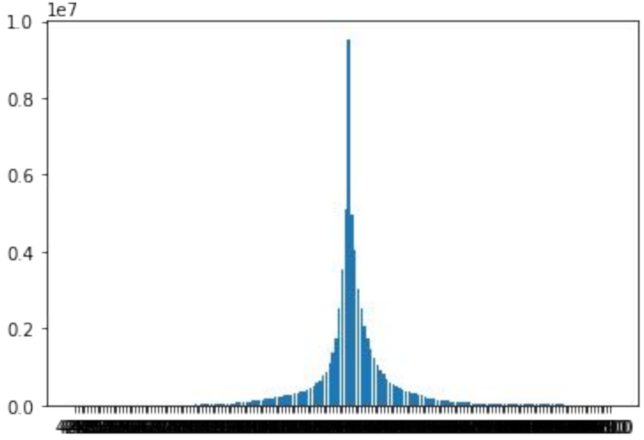
- We then put the values of **a** and **b** into **10** bins each and converted the problem into classification problem.
- Then we have used **softmax** as our activation function.
- If we could use more bins the colors would yield more smooth
- But for Resource and time shortage we used 10 bins only

Challenges

- The data were not evenly distributed. The data were centrally biased.
- The central data points (near 127) yields grayish images.
- So, we have assigned weights to the classes and used weighted categorical cross-entropy as our loss function.

$$\mathrm{L}_{cl}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$

Weighted Categorical Cross-entropy



Distribution of **a** values of **10k** images

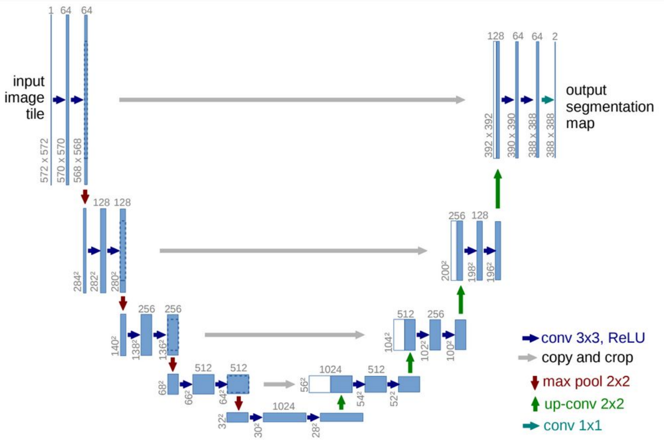
Challenges

- The channel **a** and **b** channels are not dependent
- So, we had to use to use two different models for predicting **a** and **b**
- Moreover, weights of **a** and **b** channels found from distribution were different and same model does not take different weights.

Architecture

- We have used U-Net architecture for our problem.
- We have used two separate models for predicting **a** and **b** channel.
- Each model takes 256x256 grayscale image as input and produces 256x256x10 probability distribution

  as output.



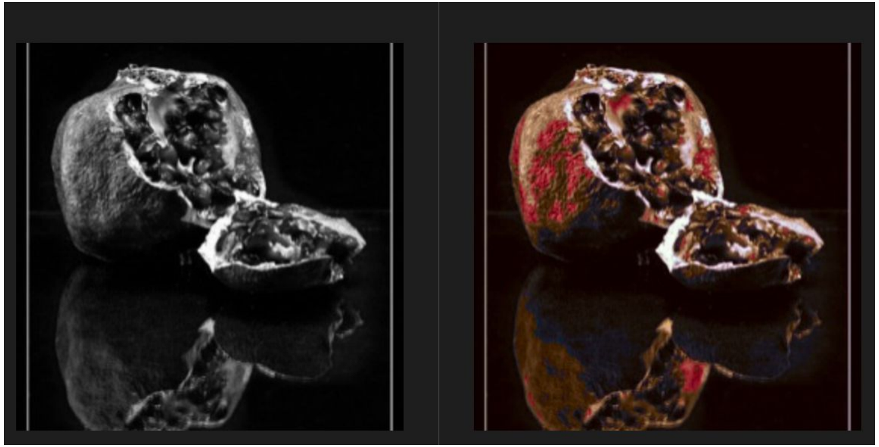
Architecture

- Instead of taking the value with highest probability, we have taken the annealed mean of the probability distribution.

$$f_T(\mathbf{z}) = \frac{\exp(\log(\mathbf{z})/T)}{\sum_q \exp(\log(\mathbf{z}_q)/T)}$$

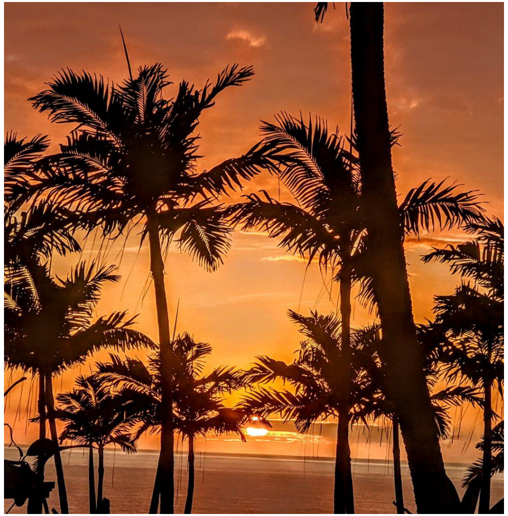
- Here **T** is a hyper parameter. If T=1, the distribution is unchanged. If T=0, the distribution becomes one hot encoded.
- We have used T=0.38 in our model.

Results



Results

Never seen image from internet



Further Analysis

- In order to color the images, our model does not simply learn the colors of each object.
- In fact, our model learns what the object is.

Original image

a channel

b channel

Future Prospects

- The output images were reddish. Weights assigned to color bins could be more balanced.
- The number of bins could be increased for better granularity.