DL Project Report - Checkpoint

Neema George - neemageo
Keshava Pranath Kakekochi - keshavap
Shashank Shankaregowda - ss675

Title

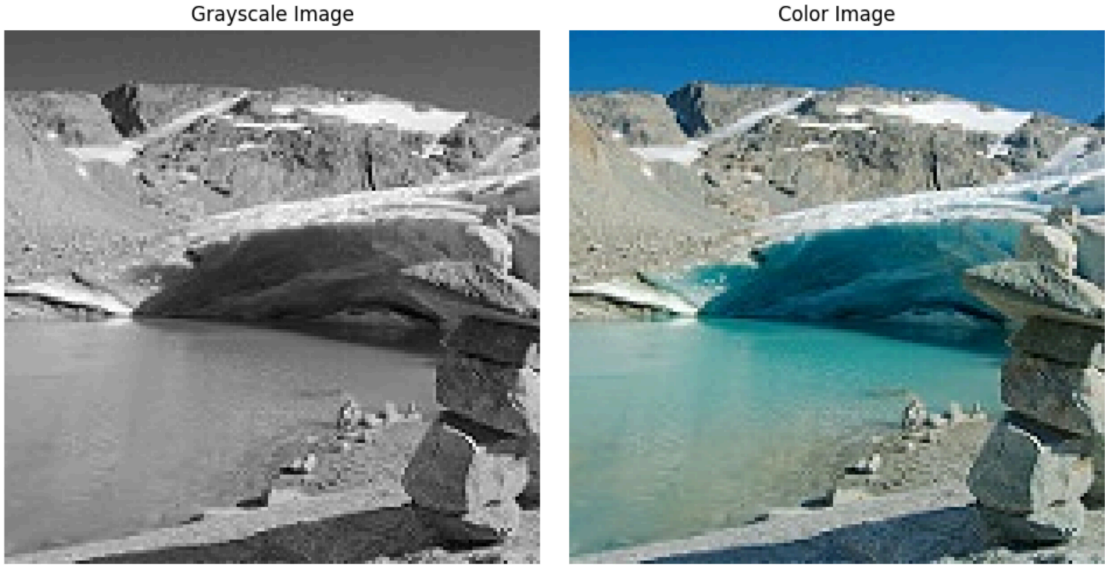
Coloring Black and White Images

Introduction

This project utilizes deep learning techniques to automate coloring black-and-white images. The proposed approach involves training using autoencoder on paired black and white and color images . The ultimate goal is to develop an accurate and realistic model that can predict the colors of an unseen black-and-white image. By doing so, not only can historical photographs be brought to life, but applications such as photo editing, and object recognition can be achieved.

Dataset

We have used the dataset from the Kaggle “Landscape Image colorization” dataset.
https://www.kaggle.com/datasets/theblackmamba31/landscape-image-colorization
The dataset consists of landscape images in both color and grayscale formats, designed for training models on colorization tasks. It features a total of 7,129 images, each in color and grayscale.
The dimensions of the images are 150 x 150 pixels. Here is an example image compared with its corresponding grayscale image.

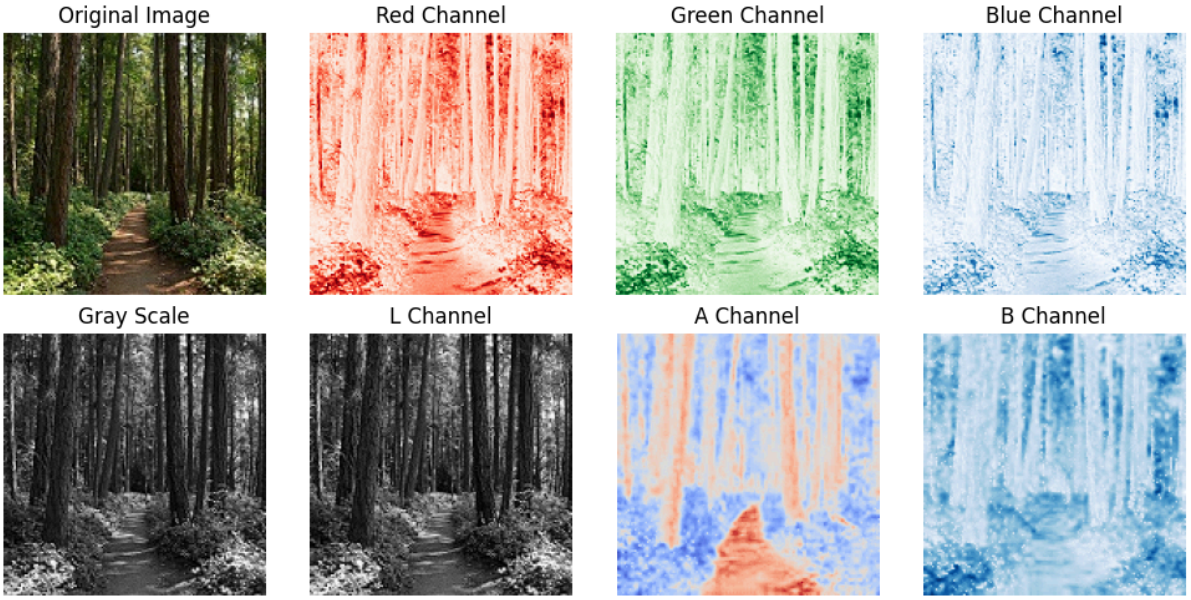


These images do not need any preprocessing as all the images are in the same size 150x150.

Baseline Model

Color images are commonly stored in the RGB format, which comprises three channels for red, green, and blue. Humans have a greater sensitivity to light intensity rather than chrominance. This simplifies the problem for neural networks from having to predict three color channels to just two. We incorporate the LAB color space into our baseline model, an autoencoder such that the neural network only has to focus on the a and b channels, which correspond to the cardinal colors perceived by human vision: red, green, blue, and yellow. The L channel, representing luminance, scales from 0 to 100, while the a and b channels span from -128 to 128.

The color image can be split into L-channel image, A-channel image and B-channel image. To understand how the RGB channels are different from LAB channels, we plot the comparison graph below. We can notice that the L-channel is the same as the original color image.



Using this, we feed the L-channel(i.e. grayscale image) into our model. The model predicts the A & B channels of the image. With this predicted value of A and B channels we generate a new color image.
The accuracy of the model is checked with the A and B channels of the original color image with the predicted A and B channels.

Training and Testing

We split the dataset into a training set, 70% of the data and testing set, 15% of the data. We then convert the coloured images from the RGB color space to the LAB color space using the `convert('LAB')` method.

Model Architecture

This model that we have used for colorization is for adding color to grayscale images. For this model we have used nn.Module to inherit some functions.

Encoder

The first convolutional layer that we have uses 3 input channels and 32 output channels with a kernel size of 3x3 with stride and padding having the value 1.
Then we have used batch normalization on the 32 output channels from the convolution layer to normalize the activations to improve the training set.
The ReLU activation function that we used to learn the complex pattern introduces non linearity to the model.

The second convolution layer that we used has 32 input channels, with 64 output channels having a kernel size of 3x3 and a stride of 2 with padding 1.
Batch normalization was then applied to the 64 output channels.
The ReLU activation function was used for this layer.

Decoder

The next part of our model will be increasing the features that we have in the model. The compressed features will be reverted to the original image size. We intend to output the target as a 2 channel image; the a and b channels in the lab color space which when combined with the original L channel, should be able to get back the colored image.

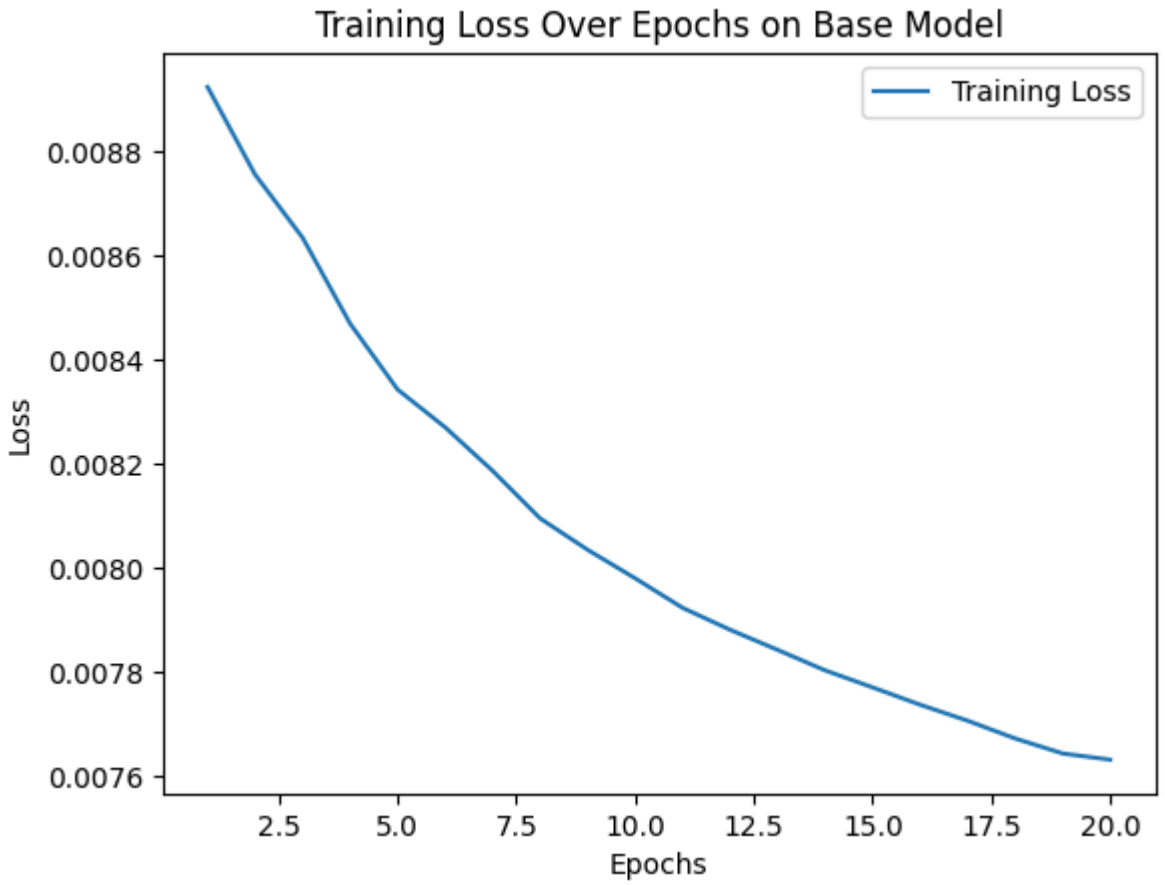
In the first convolutional transpose layer here, we have 64 input channels, 32 output channels, with a kernel size of 3x3, stride 2 and padding 1 with an output padding size of 1.We will then use batch normalization for the 32 output channels.This will be followed by the ReLU activation function.

The second convolutional transpose layer will have 32 input channels, and 2 output channels that we need for the colorization. With a kernel size of 3x3, stride 1 and padding 1, we were able to complete our model.
We were able to map a 3 channel input into a 2 channel output that is appropriate for colorizing images.

Evaluation Metrics

We couldn’t yet calculate the correct accuracy for this as the image generation is still pending. So for now we have only loss as the evaluation metric.
Training Loss:
Epoch 1/20, Loss: 0.008923576541770345
Epoch 2/20, Loss: 0.008755444143062983
Epoch 3/20, Loss: 0.008634275559765788
Epoch 4/20, Loss: 0.008469349382301936

Epoch 5/20, Loss: 0.008342839431208678
Epoch 6/20, Loss: 0.008270398892748814
Epoch 7/20, Loss: 0.008186822189973334
Epoch 8/20, Loss: 0.008095439732409058
Epoch 9/20, Loss: 0.008035153496819429
Epoch 10/20, Loss: 0.00798054146938599
Epoch 11/20, Loss: 0.00792371024353764
Epoch 12/20, Loss: 0.007881649907153005
Epoch 13/20, Loss: 0.007842977877515249
Epoch 14/20, Loss: 0.0078036892406929
Epoch 15/20, Loss: 0.007770970613003159
Epoch 16/20, Loss: 0.0077377369340795735
Epoch 17/20, Loss: 0.007707022428990175
Epoch 18/20, Loss: 0.007672967604146554
Epoch 19/20, Loss: 0.007644143182402238
Epoch 20/20, Loss: 0.007632253954234796



Test Loss:
Test Loss: 0.009055062306716162

Future Work

We now have the predicted A and B channels.
Now using these A and B channels, superimposing them with the original L channel image(grayscale image), we can create the colored image.
We will then compare the generated colored image with the original colored image and the model accuracy can be calculated.
For this we need to add the image generation into the model's training and validation loop, which we will be doing in the next phase.

References

- https://emilwallner.medium.com/colorize-b-w-photos-with-a-100-line-neural-network-53d9b4449f8d
- https://towardsdatascience.com/colorizing-black-white-images-with-u-net-and-conditional-gan-a-tutorial-81b2df111cd8
- https://anne-guilbert.medium.com/black-and-white-image-colorization-with-deep-learning-53855922cda6
- https://www.kaggle.com/datasets/theblackmamba31/landscape-image-colorization