Colorization using Deep Neural Networks

Arohan Hazarika

January 2024

1 Introduction

This was a project under Winter in Data Science conducted by Analytics Club, IIT Bombay. My project was based on colorization of gray scale images using DNNs.

2 Week 1 and 2

These weeks were mainly focused on learning Data Augmentation, Convolutional Neural Networks and finally applying them to detect Handwritten Images by training the CNNs on the **MNIST** dataset. I had used **TensorFlow** for training my CNN and I was able to achieve an accuracy upto **+99%**. The images where it couldn't recognize the written digits were mainly due to digits being confused as some other digit because of very same appearance.

3 Week 3 and 4

These two weeks included the final project where we had to implement **PyTorch** models for coloring B/W images based on a research paper and an article.

- Black and white images can be represented in grids of pixels. Each pixel has a value that corresponds to its brightness. The values span from 0–255, from black to white.
- Color images consist of three layers: a red layer, a green layer, and a blue layer. A combination of these three layers determine color and brightness of the image.
- The task is to basically build a neural network which link a grid of grayscale values to the three color grids.
- Three models namely **Alpha, Beta** and **Full Version** were to be created for colorization purposes.

- The main idea used in all the models was to convert RGB colors to the Lab color space while training, a Lab encoded image has one layer for grayscale, and has packed three color layers into two, which we will predict using the grayscale layer.

- The **Alpha** model was basically a simple implementation where the neural network had to predict the colors on the same dataset which had been used for training it.

- The second model **Beta** uses feature extraction for classifying different parts of a image and it uses the concept of Convolutional Neural Nets. For coloring, it uses hit and trial method and improves its parameters to reduce the loss by using a loss function. In this project, the loss function used is Mean-Squared Error(MSE) and its formula is

  $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

  The network starts by coloring all the objects brown. It's the color that is most similar to all other colors, thus producing the smallest error.

- In the **Full-Version** model, we use transfer learning by using a classifier called **Inception ResNet v2**. I didn't get good results using this model due to small size of data set being used as the set of testing images didn't get colored. I shall be working on it in the future and will put my efforts in improving its performance.

- The neural network architecture being used has four main components. The encoding and the feature extraction components obtain mid and high-level features, respectively, which are then merged in the fusion layer. Finally, the decoder uses these features to estimate the output.

- Github repository link - `https://github.com/ArohanH/WIDS-2K23/`

4 References

- Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2 - Federico Baldassarre, Diego Gonzalez Morin, Lucas Rodes-Guirao(2017)

- https://emilwallner.medium.com/colorize-b-w-photos-with-a-100-line-neural-network- 53d9b4449f8d