**IMAGE - COLORIZATION**

**Overview: -**

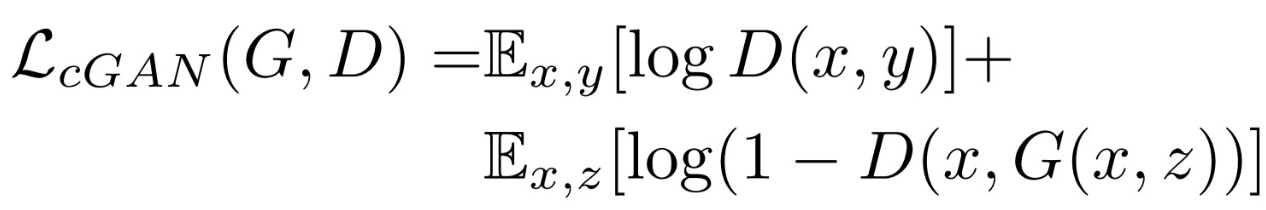
Deep Learning is an upcoming subset of machine learning which makes use of artificial neural networks. These are inspired by the human brain and its immense structure and function. Here we make use of it to colorize black and white images

**Methodology: -**

* The colorization of grayscale images can be thought of as an image-to-image translation task where we have the corresponding labels for the input grayscale image. A conditional GAN conditioned on grayscale images can be used to generate the corresponding colorized images.
* The architecture of the model consists of a conditional generator with grayscale image inputs and a random noise vector, and the output of the generator are two image channels a, b in the LAB image space to be concatenated with the L channel i.e., the grayscale input image.
* As you might know, in a GAN we have a generator and a discriminator model which learn to solve a problem together. In our setting, the generator model takes a grayscale image (1-channel image) and produces a 2-channel image, a channel for \*a and another for \*b. The discriminator takes these two produced channels and concatenates them with the input grayscale image and decides whether this new 3-channel image is fake or real. Of course, the discriminator also needs to see some real images (3-channel images again in Lab colour space) that are not produced by the generator and should learn that they are real.

So, what about the "condition" we mentioned? Well, that grayscale image which both the generator and discriminator see is the condition that we provide to both models in our GAN and expect that they take this condition into consideration.

* Let's take a look at the math. Consider x as the grayscale image, z as the input noise for the generator, and y as the 2-channel output we want from the generator (it can also represent the 2 colour channels of a real image). Also, G is the generator model and D is the discriminator. Then the loss for our conditional GAN will be:

****

**Loss function optimized: -**

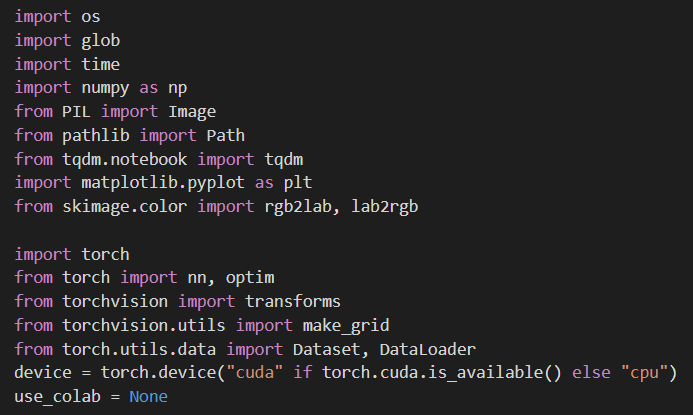
If we use L1 loss alone, the model still learns to colorize the images but it will be conservative and most of the time uses colours like "Gray" or "brown" because when it doubts which colour is the best, it takes the average and uses these colours to reduce the L1 loss as much as possible (it is similar to the blurring effect of L1 or L2 loss in super resolution task). Also, the L1 Loss is preferred over L2 loss (or mean squared error) because it reduces that effect of producing gray-ish images. So, our combined loss function will be:

****

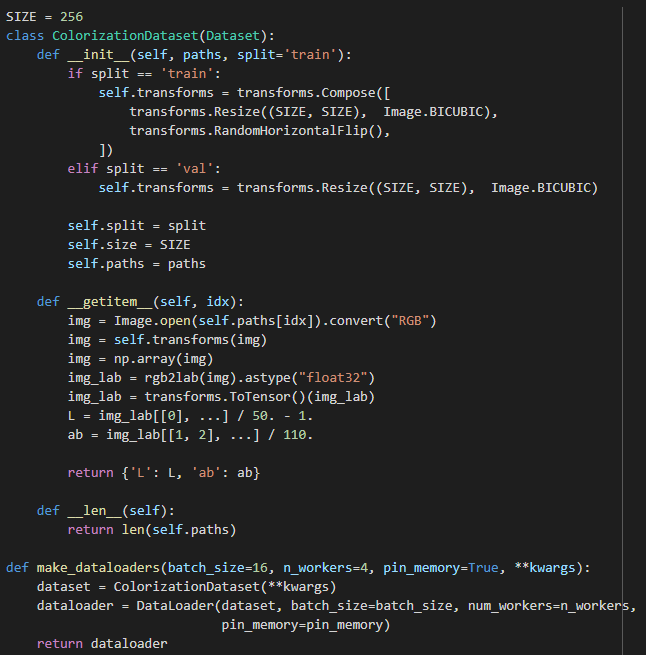
**Dataset Used: -** COCO Image dataset (10,000images: 8000 for training & rest for prediction)

**Implementation: -**

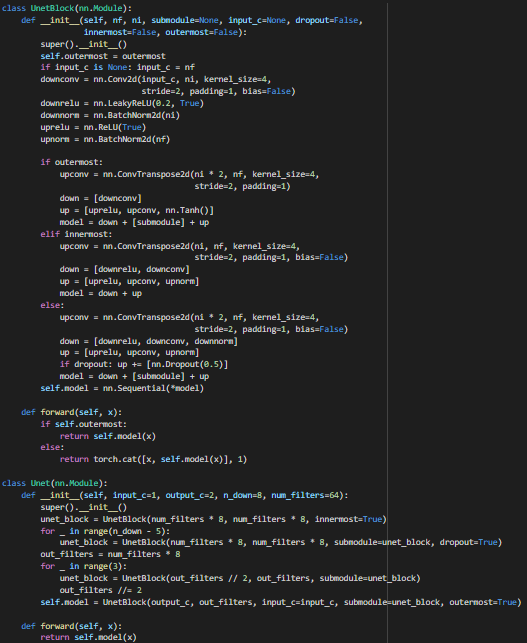
1. **Loading image paths: -**

****

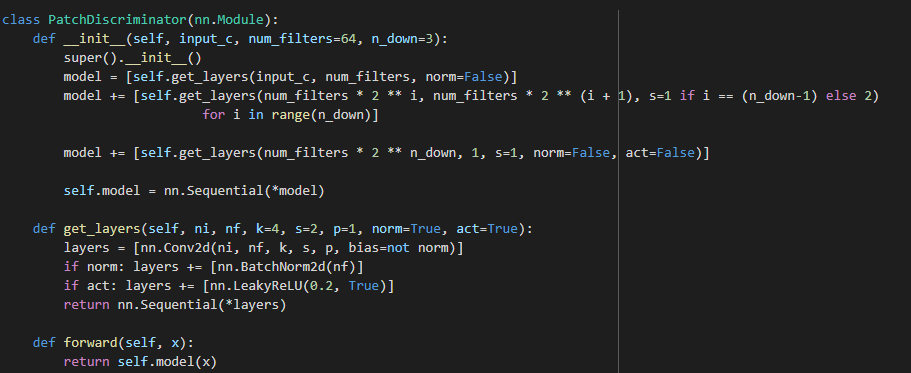
**(2) Making Datasets and Data Loaders: -**

****

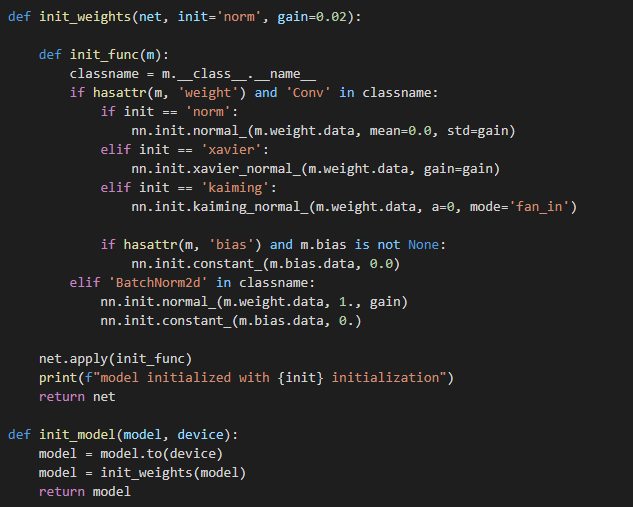
1. **U-net Generator: -**

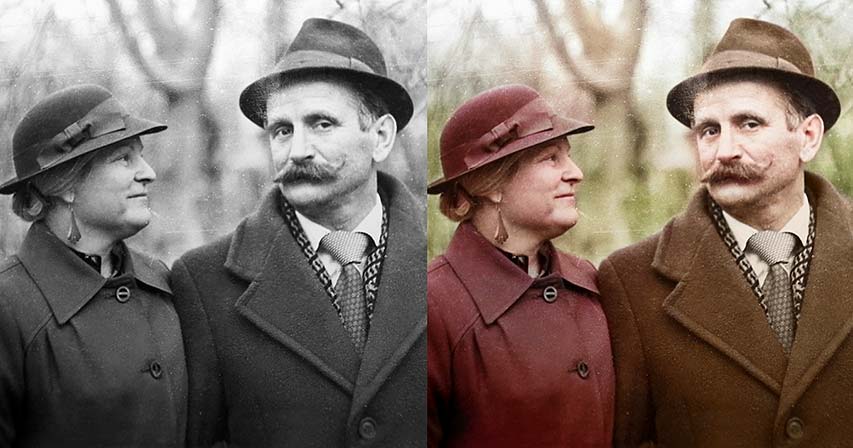
****

1. **Discriminator: -**

****

1. **Model Initialization: -**



**Results: - **

****

After 20 or so epochs:

A collage of photos

Description automatically generated with medium confidence

A person feeding a cat

Description automatically generated with low confidence 🡪 A cat drinking from a glass

Description automatically generated with low confidence

Chart

Description automatically generated

Removing Dropout Layers:

the source of the noise in the architecture of the generator proposed by authors of the paper was the dropout layers.

this conditional GAN can still work without dropout, but the outputs will be more deterministic because of the lack of that noise; Grayscale image has enough information which enables the generator to produce convincing outputs.

Quantitative Analysis: (mentioning model Loss & model accuracy)

**Graphical user interface

Description automatically generated with low confidence**

**Chart, scatter chart

Description automatically generated**

**Conclusion:**

Qualitatively, Pictures after saturation resembles the original image with or without dropout layers, after a few Epochs, obtained images are as shown with graphs attached (till nearly 200 or so epochs) & after running for nearly 900 epochs, and tuning the hyperparameters, images were quite comparable to original RGB from the dataset & Gain loss can be optimized pretraining the adversarial network. Finally, Our model was convincingly able to predict the \*a & \*b layers using L of LAB space after combing all three which gave the final RGB images, which were quite accurate based on the COCO dataset we used.

**References:**

Original paper on Image-to-Image Translation (using U-Net generator & PatchGAN discriminator)

by Berkeley AI Research (BAIR) Laboratory, UC Berkeley

<https://arxiv.org/pdf/1611.07004.pdf>

For GANs (Introduction, GANs, C-GANs, Algorithm, etc.)

<https://jonathan-hui.medium.com/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09>

<https://jonathan-hui.medium.com/gan-cgan-infogan-using-labels-to-improve-gan-8ba4de5f9c3d>

By:

Saransh (20115125), EE

Shivam Kumar Singhwal (20115134), EE