Image colorization

Venkata Bhanu Teja Pallakonda (bhanu@tamu.edu)

I have tried with default autoencoder architecture and UNET architecture with various loss functions (MSE, SSIM, TVLOSS, Pretrained VGG feature loss, Weighted of previous 4 losses). Also, instead of predicting 3 RGB channels with an input gray image. The image is converted to lab space (grayscale and channel a, b). Now we can reduce our regression task by one channel with is a lot. So, all the methods I tried I used input as gray scale image and predicted two a, b channels. And finally combine gray image with predicted a, b channels to get final color.

Method 1:

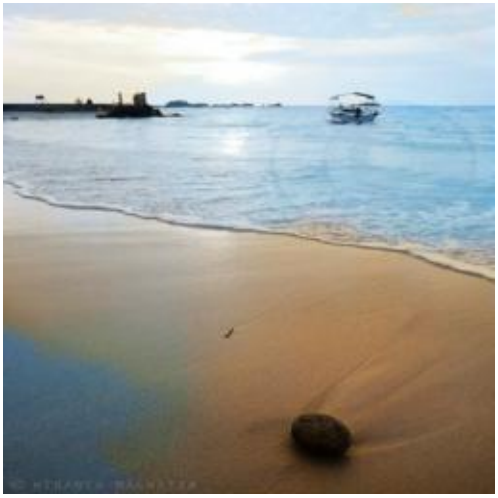
Default net (Resents and up sampling) with MSE LOSS

After method was that MSE isn't a good criterion of the colorization task. I have tried with SSIM loss and VGG feature losses. Also, my intuition is that this problem can be slightly related to semantic segmentation task so that if we can learn the colors of the segments, we could get the over all good scenic colorization. So, I used UNET as UNETS seems to perform decently good on semantic segmentation tasks.

Method 2: With UNETS and various loss approaches.

a) MSE loss

b) SSIM loss: SSIM loos seems to preform lot better because we are looking on overall generality of the image in 11*11 window. Whereas for MSE loss we are hard imposing penalty over each pixel and asking out model to predict. Which we don't require, and we want to maintain overall generality of the image. (Because, given an gray scale image there could be various color images).



c) Weighted VGG Feature loss:

For this loss method. I took a pretrained VGG model and passed out initial original image and predicted color image and took the features of first 8 layers and computed loss by comparing these features and model seems to be performing very good when compared to earlier cases. Furthermore, I also considered other looses (MSE, TP) and weighted these 3 loses as my final criterion. I spend most of time on this model as there were few components that I had to rewrite. Exploring more weighted losses with this could yield even more better results.

Few prediction outputs :