Task 1 - Image Colorization

Objectives:

1. Instead of creating a new model, use the existing model and analyze and implement different loss functions.

-> Objective completed. I have used the provided UNet model and trained the model using 2 different loss functions i.e., **L1 Loss** and **Mean squared error**. Comparison between the two training curves can be easily seen from the given **fig.**

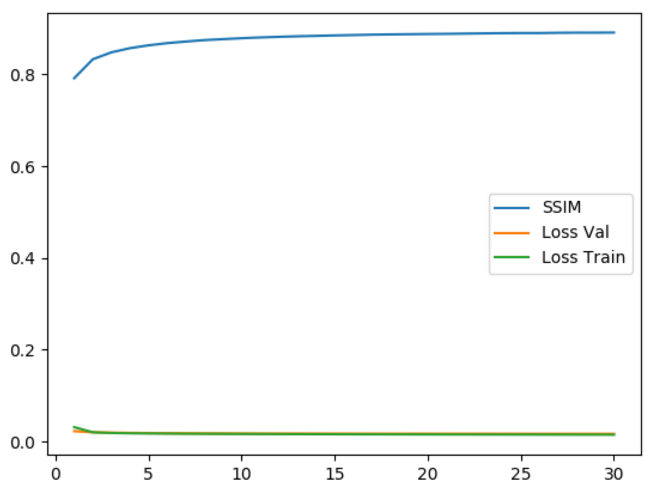

Fig. Training curve for MSE

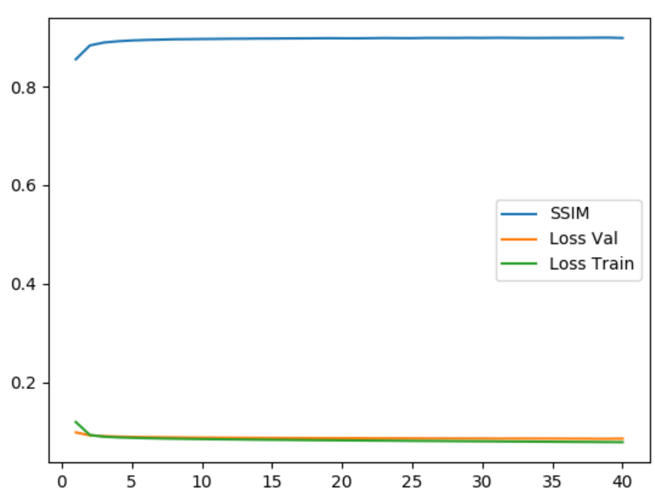

Fig. Training curve for L1

2. Try different activation functions, pre-processing and post-processing steps, and analyze their effects.

   -> I didn't toggle the activation function as UNet with **ReLU as the activation** function performs pretty well. We can try using **Leaky ReLU** to avoid the problem of *vanishing gradient* which is faced during using ReLU (although it is not necessary that we will face that problem here).

   -> Do not have many ideas about what kind of pre-processing and post-processing techniques I should try.

3. Without changing the base architecture, try to add different modules (e.g., attention, etc.)
   and understand their effect on the performance.

   -> Taking reference from *Attention U-Net: Learning Where to Look for the Pancreas* paper, I made the inference that using an attention module helps the model in learning the relative position of objects and reduces computation cost by ignoring looking in unnecessary areas for a particular object.

   -> But I don't think it will be of any use here because here every region in the image is of nearly equal importance.

4. Look into different metrics to evaluate your model performance for different settings.

   -> I have explored various metrics for this task, for example- **MSE**, **PSNR** (peak signal to noise ratio), and **SSIM** (structural similarity index).

   -> MSE just gives an idea about how much different the values of each pixel are on average.

   -> But PSNR is an interesting metric as The PSNR block computes the *peak signal-to-noise ratio*, in decibels, between two images. This ratio is used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image. The higher the PSNR, the more similar the images are.
   Here I have considered the image as the original image and predicted the colored image as an image containing noise. It was an interesting metric to learn I didn’t know about it earlier.

   -> PSNR was used earlier but now SSIM is preferred as It is the measure of similarity between two images and it is based on how humans observe the similarity between two images. SSIM is a perception-based model that considers image degradation as a *perceived change in structural information*, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. The difference with other techniques such as MSE or PSNR is that these approaches estimate *absolute errors*. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. Luminance masking is a phenomenon whereby image distortions (in this context) tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is a significant activity or "texture" in the image.

Conclusion: SSIM is the most suitable metric for this task among all the above metrics. The value of SSIM is close to 1 when the images are more similar and close to zero when the images are very different from each other.

Learning Outcomes:

-> Got an even better understanding of UNet.
-> Learned about Nested UNet and attention guided UNet.
-> Different metrics like PSNR and SSIM.
-> Learned more about the composition of images.
-> Read about some methods based on GANs (Generative adversarial networks) for the same task.

A proposed solution to the problem:

As the information contained in each pixel is not completely independent of its surrounding. In an image pixels close to each other have similar values so we can say that the pixels close to each other are related. In the current model, the problem is that its prediction is only affected by the value of a pixel in the grayscale image, that's why there are some regions in the predicted color images where the color suddenly washout and even the colored image looks grayscale in that region (for example the marked region in the given figure). However, if we can find a way to relate the information of pixels locally to each other then we can overcome this shortcoming of this model and the predicted images will look more natural.

References:

1. **Attention U-Net: Learning Where to Look for the Pancreas**

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, Daniel Rueckert

2. PNSR (https://www.mathworks.com/help/vision/ref/psnr.html#:~:text=Vision%20Toolbox%20%2F%20Statistics-,Description,the%20compressed%2C%20or%20reconstructed%20image.)
3. SSIM (https://en.wikipedia.org/wiki/Structural_similarity)
4. GAN based solution (https://towardsdatascience.com/colorizing-black-white-images-with-u-net-and-conditional-gan-a-tutorial-81b2df111cd8)
5.