Colorization Of Gray-Scale Images Using U-net Architecture

L.Danon * T.Ladijinsky *

June 2020

Abstract

In this paper, we present our approach in making a neural network which colorizes black and white(gray-scale) images from a data-set of scene pictures which was converted from RGB to gray-scale in the data collection process. We do so by utilizing a U-Net architecture, which has shown great results for tasks of image segmentation and image colorization, and achieve mean squared error of 331.0627 after 9 epochs of training on an NVIDIA Tesla K80 GPU which proves that U-net architecture is a very suitable architecture for image colorization and specifically for the case of scene colorization. We have reduced the time and resources needed to color images by showing a lightweight architecture which produces considerable results without human intervention in the process.

1 Introduction

Every one of us has seen a black and white photo in some stage of their life, no matter one's age, and wondered how the situation looked in color. They are in our family albums, history textbooks, old beloved TV-shows, movies and countless other places. Coloring old gray-scale images photos will give us better understanding and help everyone learn better their history. It can be used for reconstruction of photos or videos and 'modernize' black and white films. Notwithstanding the prevalence of black and white photos in today's world, the problem of colorizing black and white images is a painstakingly labor-intensive process requiring a photoshop professional to sit on a computer for long hours picking colors, hues, correct positions, shades etc. This method is imperfect becuase of the reliance on the whims of the worker. Therefore arises the need for further continuous research of ways to automate the process of colorizing images in order to save time and money. Automatic image colorization is an ill-posed problem which turns out to be very hard to solve due to the fact that colorizing an image corresponds to finding a mapping from n dimensional vector of gray-scale pixel data to 3n dimensional RGB pixel data. This means that for

*Holon Institute Of Technology, Dept. of Mathematics

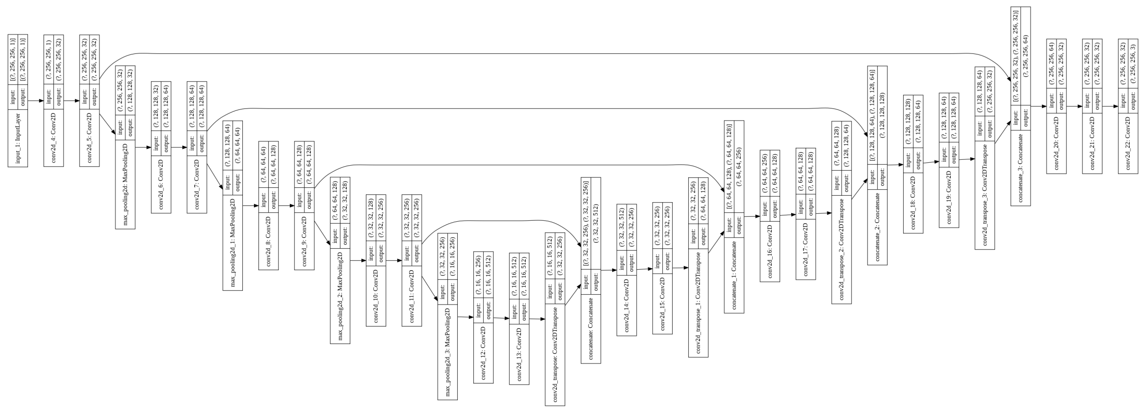

Figure 1: Our U-net architecture

the truth of the matter is that there might not be enough information in the image data to determine a suitable coloring of it. In these paper we will show our attempt into colorizing black and white images of scenes from all around the world using a U-net architecture.

2 Related Work

As said in the introduction there are numerous applications of automatic image colorization which explains the surge of research in the field most famous of which being DeOldify which uses generative adversarial network architecture (generator discriminator) to colorize images thus removing the need to determine a loss function on pixel values. DeOldify also used a U-net in their architecture as it has shown good results for the task needed because of its skip connections which let it accumulate information from both worlds being higher level features extracted by the depth of the network and low level features showing corners and very basic shapes (finer detail). The paper [1] is much closer to our research. Like them we have used a U-net architecture in order to colorize images but unlike them we chose to not discretize the colors to allow for a wider spectrum of colors in our colorization. Also, in [3] they have shown that connecting the hidden layers of a bottleneck neural network to each other also known as a U-net helps in attaining good results as it aids in conserving localization information from the image after the bottleneck which would be mostly useless for problems such as classification but for problems like image segmentation and colorization which make much use of the location information it measurably improves their results. This insight helped us in deciding the network architecture in our paper as it was shown that this architecture can be trained quickly and achieve good results nonetheless. In close proximity to the work done in [1], in [4] they tackled the problem of image colorization as a classification task by binning the colors and then used a classification loss function, re-balanced to rare classes so that also colors that appear less in the data may appear in the colorization of the images. They, unlike us, also used a very big network based on the VGG-16 network with added layers and dilated convolutions and trained it on the Image-Net dataset which is known to be a very big dataset with an entire yearly competition based around it benchmarking the state of the art in image classification task. We did not opt for such big network as we had less data and less computing power then they had. Our solution is also an end to end solution and does not involve post-processing like in [2] which uses multiple post-processing methods in order to improve the results of the colorization.

Figure 2: Test Set Image Colorized using our U-net

3 Main research

3.1 Data

A dataset from Kaggle called "Scene Classification", which was originally intended for classification task, was used. The images were converted from RGB to gray-scale images such that the normalised gray-scale images were given to the model as input and the non-normalized RGB images were given to the model as output.

3.2 Architecture

As you can see in Figure [1] our U-Net is built using an encoder decoder architecture with skip connections where the encoder uses Max-Pooling layers to reduce the height and width of the images while the number of channels grows and our decoder uses ConvTranspose layers to make the height and width higher and channels smaller (as typical for U-Net architecture). Our architecture is a symmetric architecture utilizing the fact our images are scaled to 256 by 256 which is a power of two in order to make the transformations intuitive for the reader and network. ReLU activation functions were used for the convolution layers of the model including the output layer such that an image would have to be clipped in range before transformed to an image. We opted for ReLU in output layer instead of sigmoid activation as sigmoid would have forced high values in the output to be enormous in size for very white or very black colors which are quite common in nature.

3.3 Training

For training, we have used an Adam optimizer which minimizes MSE as our loss function on the output both being the standard for many Deep Learning tasks (MSE loss specifically for regression). The model was trained for 9 epochs

Figure 3: Test Set Image Colorized using our U-net

over the entire training data which consists of 80% of the entire data-set and the test data being the remaining 20% with using Tensorflow library in Python, with prefetch optimisations for acceleration of the training process. The model was trained in google colab which operates as a virtual machine with NVIDIA Tesla K80 GPU which vastly sped up our NN training process.

3.4 Miss-Colorizations

As in any Deep Learning task there are cases where the colorization does not resemble the actual photo or has some problem in it.We have encountered two basic phenomenons.

- Spotting of colors where you have spots of either the gray-scale image or another color on an otherwise correctly colored image (seen as example in Figure 3 where one can see spots of blue in the water and sky).
- Miss-Colorization of the object where the object is colored with a color not matching the original image which happens mainly where the color is a rare color not present much in the data set such as red.

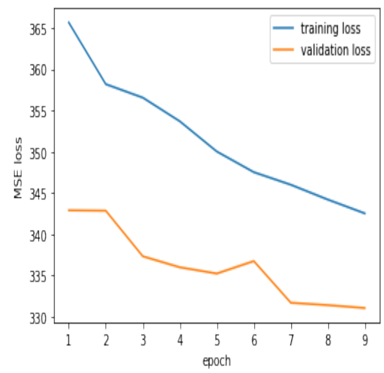

Figure 4: The training process

4 Results

As you can see in Figure [4] in the main network we achieved MSE loss of 331.0627 in the validation set between RGB non-normalized pixels on the test data. We have successfully created an end to end solution capable of coloring images of scenes to different degrees depending on the image and the frequency of occurrence of the objects in the image. Our previous attempts at the colorization images field were done on a different data-set of images of dogs (Stanford dog data-set) but opted to change the data-set to have more color-full images to see if the network can identify patterns in the scenes it was presented with (which it clearly did).

5 Discussion

The automatic image colorization community is full of different architectures and methods to achieve the best results possible. In this paper we have tried to make a good end to end solution that can be trained on limited data and computational resources and picked the one which worked best. As of today we did not improve known models and improvement is must.

References

[1] Vincent Billaut, Matthieu de Rochemonteix, and Marc Thibault. "ColorUNet: A convolutional classification approach to colorization". In: *CoRR* abs/1811.03120 (2018). arXiv: `1811.03120`. URL: `http://arxiv.org/abs/1811.03120`.

[2] *ICCV '15: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. USA: IEEE Computer Society, 2015. ISBN: 9781467383912.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: `1505.04597`. URL: `http://arxiv.org/abs/1505.04597`.

[4] Richard Zhang, Phillip Isola, and Alexei A. Efros. "Colorful Image Colorization". In: *CoRR* abs/1603.08511 (2016). arXiv: `1603.08511`. URL: `http://arxiv.org/abs/1603.08511`.