Image Colorization Project

Alessandro Ardenghi, Rocco Cristiano Giampetruzzi, Rolf Minardi

May 2024

1 Introduction

In this project, we aim at solving an Image Colorization task. More specifically, our objective is to develop a robust and general model that is able to take as input historical grayscale images, and output a colorized version of it which a human would find credible. Obiously, this task can be classified as self-supervised learning, since given an image, it can be split into luminescence, AB channels, and the former can be used as label to predict the latter.
Let us remark that we are not seeking to build a model which can guess the true, real-life color of a grayscale object in a picture, as that would likely be impossible, but rather we aim at building a model which will return plausible colors on an image, with the final objective of producing predictions good enough that a human cannot distinguish them from actual RGB images.
To solve this task, we will use a U-Net like architecture, composed of a pretrained encoder (ResNet50) which will perform downsampling, and a learnt encoder that performs upsampling (using *pix2pix* layers).
Let us also remark that, because of space constraints, we will only sum up our analysis in this report, while all the details can be found on the Notebooks.

2 Data

We initially decided to train our models on the Microsoft COCO dataset, which features thousands of images of objects, landscapes, animals and people in various formats. We selected 40k images on which to run the training and the validation steps, and to test the models we used historical images found online, as well some 700 images of the COCO dataset which we did not use for training and validating. After having analyzed the results obtained by the models trained on this dataset, we noticed a constant incapacity of our model to precisely color people, and therefore we decided to train some new models specifically on a dataset made of pictures of people (this dataset is made of the images that can be found at: *http://chenlab.ece.cornell.edu/people/Andy/ImagesOfGroups.html* (section Datafiles)).

Figure 1: Sample Images from COCO

3 Architecture

As mentioned before, the Architecture consists of a pretrained encoder (from ResNet50) and a decoder (using pix2pix layers). The trainings were composed of two steps: first, we froze the ResNet50 layers and only trained the upsampling layers and the final layers. Then we unfroze the ResNet50 layers and performed a finetuning on the whole model. Since the ResNet50 was pretrained on RGB images, we had to feed it as input 3-channel images, therefore we concatenated together 3 equal lumiescence layers, and obtained the black and white image in RGB format. We present and evaluate 4 models:

- Model trained and finetuned on 40k images of the COCO dataset, using 224x224 images and MSE Loss. (**Model1**)
- Model trained (but not finetuned) on 40k images of the COCO dataset, using 224x224 images and MSE Loss. (**Model2**)
- Model trained and finetuned on 1k images of the People dataset, using 224x224 images and MSE Loss. (**Model3**)
- Model trained (but not finetuned) on 1k images of the People dataset, using 224x224 images using weighted Cross Entropy Loss. (**Model4**)

Note that you can find the various learnt weights in the linked OneDrive Folder.
Instead of relying on a numerical loss value, which carries little meaning in a task like image colorization, we evaluate our model based on how realistic the reconstructed images appear to the human eye.

4 Regression with MSE

Our first approach to the task was to model the problem as a regression in which we try to predict the values of the 2 channels A,B given the Luminescence as input. We train and finetune the U-net architecture described in section 3 on 40k images taken from the COCO dataset and we use the Mean Square Error Loss. We test both the not finetuned and the finetuned models on a test set extracted from the COCO we get the following predictions (more images in the Appendix):

Figure 2: Predictions of the not finetuned model on COCO dataset

Figure 3: Predictions of the finetuned model on COCO dataset

As we can see from the above results, the model which was trained on 40k images but not finetuned (Figure 2) generally colors images with brighter colors, whereas the finetuned model (Figure 3) tends to color images with desaturated colors. On the other hand, the finetuned model is much more precise in coloring images inside their contours, and rarely has random splashed of color on the image like the not finetuned model has.

Moreover, we notice is that both the models generate predictions which are almost good, but the predicted colors are desaturated. This effect is probably due to the type of loss that we are using, indeed, colors are not evenly represented in training images, with green and light blue being very frequent (in grass and skies), whereas bright colors like red are quite rare, so the model correctly colors skies and grass, but struggles with people, shirts and other objects. Furthermore, the MSE favors mild colors, which have short "distance" to all other colors, so desaturated colors like gray and light brown. Since the models' main problem is the colorization of people and the desaturation of the colors, we try to build a new model on a dataset of pictures of people (**Model3**), as well as a new model with a different type of loss (**Model4**).

We thus trained and finetuned again the model on a smaller dataset of 1k pictures of people and what we get is much more reliable results on predicting people, and surprisingly, even though it was trained on portraits of people, the model also learnt very well to recognize and color landscapes, which are often in the background of pictures. In Figure 4 we show some results of this model.

In conclusion, it performs much better than the others in this case and most of the predictions have bright colors and precise countours, along with great background coloring.

Figure 4: Predictions of the finetuned model on People dataset. The first image is taken from COCO while the other two are taken from People dataset.

5 Classification

In this section we proceed with the second approach with the aim of solving the problems of desaturated colors and that images exhibit a limited color gamut, utilizing only a narrow range of colors from the available palette. To solve this issue, we proceeded with a literature review, and understood that this is quite a common problem. Zhang et al. (2016), proposed to use a weighted crossentropy loss, instead of MSE. In their paper "Colorful Image Colorization", this approach led to better results. Thus, we decided to reframe the task as a classification problem, where given the values of the luminance channel of the picture, we quantize the AB space, and then try to correctly classify each pixel, assigning it to the correct quantized value, and then reconstruct the image by mapping the picture to RGB for visualization. We tried to proceed analogously to the paper, and here we briefly discuss the implementation of this new loss.

We build a 25x25 grid of quantized bins for the AB channels, where each grid tile has size 10x10, and if a pixel has AB channel values $[x, y]$, we map the pixel to the bin $[y/10, x/10]$. The aim is to populate this grid by counting the number of occurrences of each bin, and to then extract an empirical probability distribution on the weights from this.
Therefore, our first step is to traverse the dataset and obtain the empirical probability distribution on the bins. We then use this distribution on the quantized bins to obtain a matrix of weights which is inversely proportional to the probability of the bin, in order to account for color unbalance in the dataset and give more weight to unlikely colors (see "Colorful Image Colorization", Zhang et al., 2016 for more details). The idea of this weights is to penalize colors that occurr often, like gray, light blue and light brown, and to favor rare colors, like bright red...
We observed that some combinations of A and B channel are extremely unlikely, therefore we remove such bins from our prediction, and restrict the labels to the bins which have nonzero probability in the dataset. This allows us to reduce the number of labels from 25x25 (625) to only 243.

Here is the Loss function:

$$L_{cl}(Z^*, Z) = -\sum_{h,w} v({Z^*}_{h,w}) \log(Z_{h,w,{Z^*}_{h,w}})$$

where h and w describe the pixel position, $Z^*_{h,w}$ corresponds to the quantized ground truth label (i.e., the bin corresponding to the true ab values of an image in pixel h,w), $Z_{h,w}$ our predicted distribution (as usual for cross entropy), and v takes care of the reweighting.
More details about the reweighting can be found at page 6 of Zhang et al.

We finally trained the U-net like architecture on the 1k People dataset, without finetuning the full model because it was too expensive. Indeed, as explained before, now we are solving a classification problem so we have much more output channels (240-260 channels) to predict. As explained before, the network returns a probability distribution over the possible values colors in each pixel, and for sampling from this distribution we can use different temperatures. Here we show the predictions given by the optimal temperature in Figure 5 (to see more results also with other temperatures refer to the notebooks).

Although the predictions are good, we expected more vivid colors in the images. From our understanding and analysis this is due to the fact that we had the possibility of training it with a restricted dataset, while on the paper mentioned above they used around 1 million images. Moreover by changing temperature, the results did not show a drastic change as we would expect. In the end, the result are still good, but do not properly solve the problem of the previous network.

Figure 5: Predictions of the model trained using our loss on People dataset. The temperature used to sample is 0.05.

6 Conclusion

After the training of all these model, by using these different techniques, we are now ready to use the best among those to reconstruct black and white historical images. In figure 6 and Figure 7 , we show some of our results with the models trained on COCO, and you could find the rest in the appendix and in the notebooks.

Figure 6: Predictions of the non finetuned model on COCO dataset (MODEL 2)

Figure 7: Predictions of the finetuned model on COCO dataset (MODEL 1)

In the Figure 9 and 8, we show the results using the Model3, which appears to be the model producing the best coloring of the images. Since it appears to work so well, we tested it also on some old gray scale pictures of our grandparents and, indeed, it produces very good predictions (Figure 8). Lastly, about Model 4, the predictions on the historical images are not very satisfactory, so we do not include them here, but for the sake of completeness they are also included in the notebooks.