Colorization of Grayscale Images

Category: Computer Vision

Bardia Beigi
Stanford University
bardia@stanford.edu

Vamsi Chitters
Stanford University
vamsikc@stanford.edu

Fabian Frank
Stanford University
fabfrank@stanford.edu

Abstract

Image colorization is a challenging task and a topic of ongoing research in the area of Computer Vision. This paper presents an approach for colorizing grayscale images in a way that makes it challenging to discern generated images from ground-truth images. Building upon methods from existing literature, we propose a CNN-based architecture for this task. Furthermore, we include a detailed account of the incremental changes that were made to the model throughout the development process, ranging from regression to classification models. The best model which relies on AB-plane discretization, color rebalancing and label smoothing achieves close to 70 % on user surveys results.

1 Introduction

Colorizing images is an interesting image-to-image translation problem that enables historical photographs (and those captured on grayscale sensors) to be seen with the true colors they were captured in. The objective of this research effort is to hallucinate colors by using deep neural networks such that the output image seems natural to the human eye.

(a) Original Grayscale Picture

(b) Colorized Picture

Figure 1: An example pair of input and desired output

The main challenge is to reconstruct lost information in terms of color values as close as possible to a natural, colorized version of the input.

2 Related Work

2.1 Semi-automated Colorization

Until recently, colorization of grayscale images was a semi-automated task which involved human assistance and input. [9] proposed to group neighboring pixels with similar intensity into the same color through optimization. [5, 10] improved upon this by incorporating color bleeding and color continuity.

2.2 Convolutional Neural Networks

[3] was one of the first to implement a fully automated approach with CNNs, but only achieved mixed results with rather unsaturated outputs due to the averaging effect of the L2 Loss. This problem has been tackled using both regression [4] and classification models [1]. The breakthrough came in 2015 with a fully convolutional network [2] trained on the `CIFAR-10` dataset. [11] combined ConvNets with prior belief on prior color probabilities to improve the results substantially.

2.3 Generative Adversarial Networks

Recently, GANs have become increasingly popular in learning not only the input-output mapping, but also the loss. With the rise of Conditional GANs, [6] was able to achieve more natural-looking outputs based on various image-translation problem settings, including colorization of grayscale images. [12] took this one step further and was able to achieve favorable results even with unpaired input and output images.

3 Dataset

We utilize the `CIFAR-10` [7] dataset, which comprises of 60,000 32×32 images (45,000 images for training, 10,000 images for evaluating the performance of the models, as well as 5,000 unseen images). The training set we construct is comprised of tuples $\{i_g, i_c\}$, wherein i_c is a color image and i_g is the corresponding grayscale image. Naturally, it follows that the test set is comprised of only grayscale images to be colorized by the model. Initially, we attempted to train the models on the `ImageNet` dataset, but due to the large distribution of images, training is 10-20 times more computationally expensive than when training on `CIFAR-10`, which is why we decided to work with `CIFAR-10`.

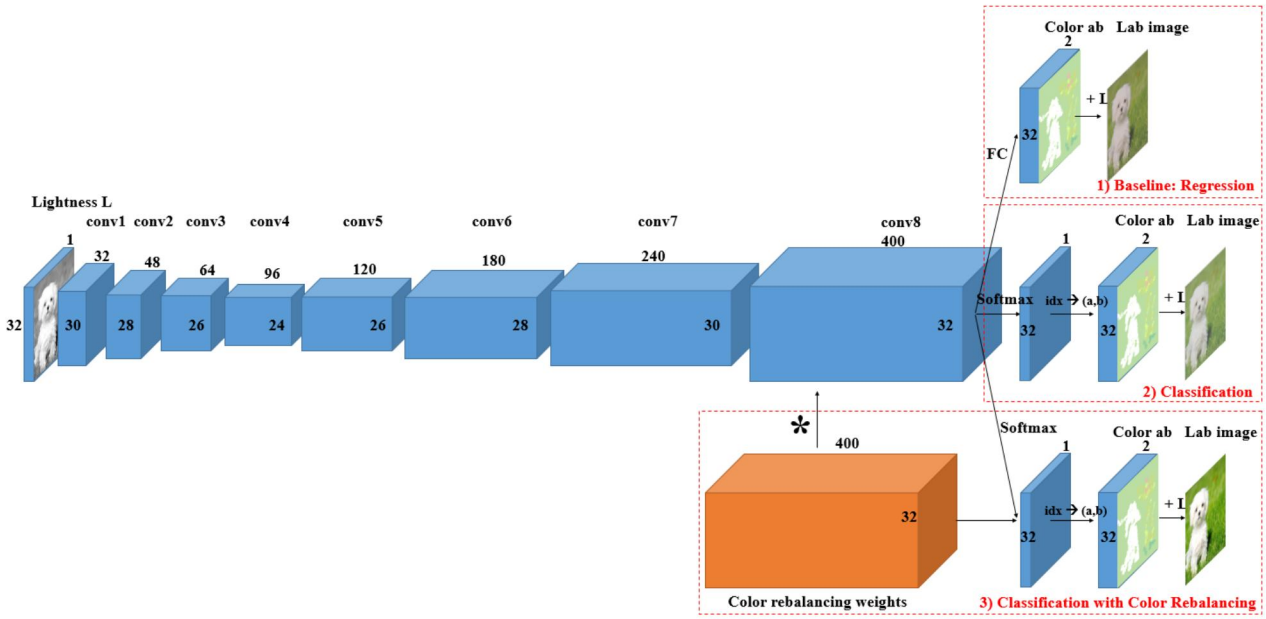

Figure 2: Model architecture. The input layer is a 32×32 grayscale image. Through four convolutions with 3×3 filter size, the image dimensions are reduced to 24×24, before they are deconvolved back to 32×32. In the third dimension, we apply an increasing number of filters (32, 48, 64, 96, 120, 180, 240, 400) branching off at the end to serve the purposes of our three models.

3.1 AB Plane Discretization

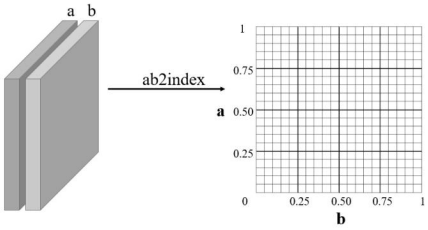

Figure 3: A 20×20 grid with cell size of 0.05 to convert the AB channels of a pixel into one index (0 to 399).

It is more intuitive to perform the colorization task in the LAB colorspace, wherein the L channel determines the intensity of each pixel (input to the model) and the A and B channels together determine the color of the corresponding pixel (output of the model). Each of the A and B channels ranges in possible values from 0 to 255.

Since the output of each pixel will be a point in the discrete AB color plane, colorization can naturally be categorized as a classification task. However, the number of possible classes $255 \times 255 = 2^{16}$ is too large. As a result, the AB plane is discretized into $20 \times 20 = 400$ bins (Figure 3) to establish a more reasonable number of classes. In the process, the A and B channels are normalized to represent values in [0,1] range. The conversion to bins follows the function $(a, b) \rightarrow$ idx where $\texttt{idx} = [0, 400)$:

$$\texttt{idx} = \lfloor \texttt{a} * \texttt{num\_row} \rfloor * \texttt{num\_row} + \lfloor \texttt{b} * \texttt{num\_column} \rfloor$$

Through this process, we index a pixel based on its AB channels and map it to a cell on the grid. To convert the index back to the AB channels, the following conversion function is used idx $\rightarrow (a, b)$:

$$\texttt{a} = \lfloor \frac{\texttt{idx}}{\texttt{num\_row}} \rfloor \quad , \quad \texttt{b} = \texttt{idx} \ \textit{mod} \ \texttt{num\_column}$$

3.2 Label Smoothing

For our task, the accuracy of predicted colors is not as important as generating plausible colors in the output image. As a result, as part of pre-processing, we add the immediate neighbors of the label color in the AB plane (a window of 3×3) as acceptable colors without affecting the loss of the model. This is achieved by creating a 3×3 kernel of (almost) uniform values and convolving it with the singleton color label in the AB plane per pixel as depicted in Figure 4.

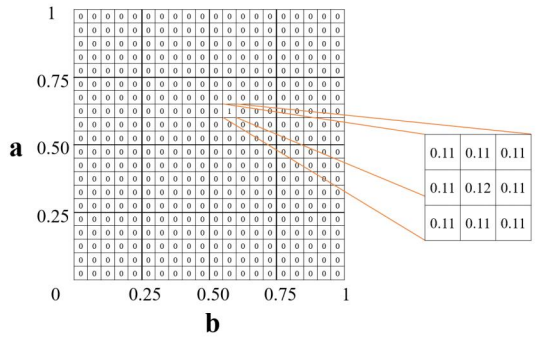

Figure 4: Label smoothing achieved by convolving a kernel with the singleton label color in the AB plane.

4 Approaches & Models

For the purpose of this project, we focus on a CNN-based approach to tackle the colorization problem. CNNs [8] have recently become the state-of-the-art technique for various image related tasks, including classification, style transfer and image generation. It therefore seems like a natural extension to try using it as the approach for this task. We experiment with several CNN model architectures, and attempt to approach the task from different angles in three main models. All of them use a convolutional neural network as the underlying architecture, which is comprised of 4 convolutional layers with 32, 48, 64, and 96 filters, and 4 deconvolutional layers with 120, 180, 240, and 400 filters (Figure 2). All of the filters have a 3×3 size, strides of 1 and no padding. The combination of convolutional and devolutional layers is necessary to have the output image be the same size as the input image. One final design choice we employ is to have 400 filters in the last layer to match the number of classes, such that the last layer would act as logits for the classification task.

Every convolutional (including deconvolutional) layer is accompanied by `ReLU` activation and batch normalization. The models are trained with the Adam Optimizer with a learning rate of 10^{-3} and a mini-batch size of 32. The aforementioned hyperparameters were chosen after observing the accuracy results over a few epochs.

The same underlying ConvNet branches off in the end to serve as the basis for our three different strategies to solve the task as described below.

4.1 Regression

The colorization task can be treated as a regression problem. We use this approach as our baseline anticipating modest performance. In a regression model, the predicted values are compared to the label values, and the objective function is usually an L2 loss between the two. In our problem, we predict two values (A and B channels) per pixel of an image. Therefore, we minimize the loss between the predicted $(\hat{y}_a, \hat{y}_b)$ and the label (y_a, y_b):

$$L_2(y_a, y_b, \hat{y}_a, \hat{y}_b) = \frac{1}{2} \sum_{h,w} ||y_{h,w_a} - \hat{y}_{h,w_a}||_2^2 + ||y_{h,w_b} - \hat{y}_{h,w_b}||_2^2$$

A fully connected layer is added to the end of the ConvNet to output the model predictions of (a,b) pairs per pixel. It is important to note that the regression model predicts (a,b) in the unnormalized [0,255] range and not with respect to bins.

4.2 Classification

Classification is the main focus of this report. In the classification interpretation of this task, the (a, b) bin for every pixel is predicted out of 400 such bins that span the AB plane. The classes are labeled [0, 400) for convenience with class 0 at the top left and class 399 at the bottom right of the AB plane. Let's denote the predicted color bin for a given pixel as $\hat{Z}_{h,w,q}$ and label color bin as $Z_{h,w,q}$ where q shows the class. The cross-entropy objective function that we employ for this multi-class classification task is as follows:

$$CE(Z_{h,w,q}, \hat{Z}_{h,w,q}) = -\sum_q Z_{h,w,q} \log(\hat{Z}_{h,w,q})$$

Finally, softmax is applied to the last layer of the ConvNet to generate bin probability predictions per pixel. The highest bin probabilities are then converted back to the AB plane values per pixel to form the final colorized image.

4.3 Classification with Color Rebalancing

Color rebalancing improves the pure classification model by favoring more real and vibrant colors. This is done by finding how likely it is for each (a,b) pair to appear as a color based on the entire dataset. Figure 5 shows the overall color distribution in `CIFAR-10` over the AB plane. As seen, the colors with mid-range a and b values are far more popular than the corners of the plane. Those popular colors are the unsaturated ones coming from an abundance of unsaturated backgrounds such as clouds, ground, dirt, and soil.

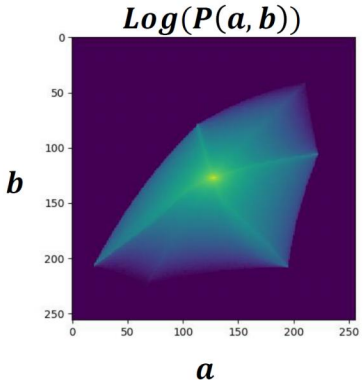

Figure 5: Log of color distribution probability in the AB plane over `CIFAR-10`

To account for the color rarity problem, once probability p is computed from Figure 5, weights $v(Z_{h,w})$ are generated per bin according to the equations below where $\lambda = 0.5$ is a tuned hyperparameter and $Q = 400$ is the selected number of bins.

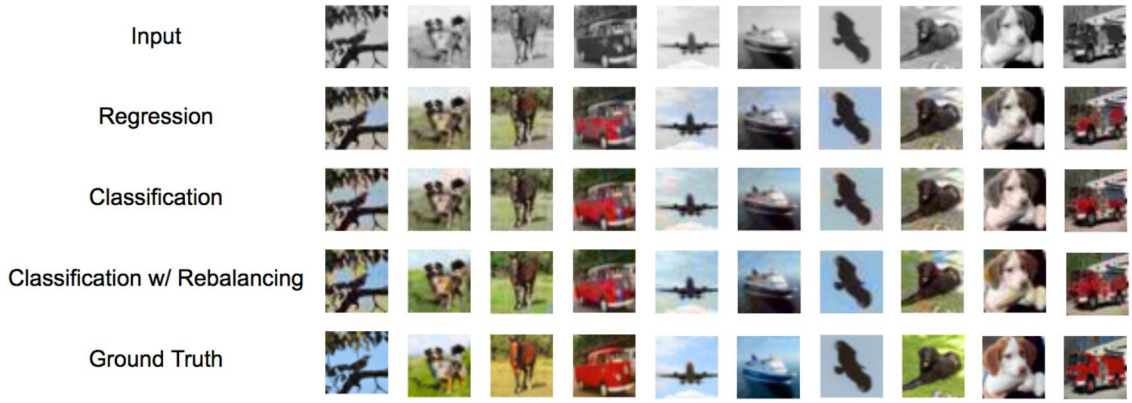

Figure 6: A comparison of the input examples, the outputs of the models, and the ground-truth images.

$$v(Z_{h,w}) = w_{q^*},\ where\ q^* = \underset{q}{\operatorname{argmax}}\, Z_{h,w,q}$$

$$w \propto ((1-\lambda)p + \frac{\lambda}{Q})^{-1},\ \mathbb{E}[w] = \sum_q p_q w_q = 1$$

$v(Z_{h,w})$ is inversely proportional to p, meaning high weights are associated with more rare colors and low weights are assigned to more popular unsaturated colors.

$$CE_{reb}(Z, \hat{Z}) = -\sum_{h,w} v(Z_{h,w}) \sum_q Z_{h,w,q} \log(\hat{Z}_{h,w,q})$$

These weights are directly multiplied with the previously discussed cross-entropy loss per pixel, optimizing a balance between bin accuracy and color saturation.

4.4 Other Models

A variety of ConvNet models comprising of different architecture, number of layers, number of filters, and filter sizes were trained over the course of this project. The aforementioned model architecture showcases the model that performed the best for the given task.

5 Results

5.1 Evaluation Metrics

Evaluating the quality of colorized images is subjective to some extent. In this regard, we evaluate the generated images both quantitatively and qualitatively.

Quantitative metrics:

We measure the accuracy for regression as the percentage of correctly predicted color values in the AB plane (1/65536 random chance):

$$Acc = \frac{1}{N^2} \sum_{(i,j)}^{(N,N)} 1\{A_{p_{(i,j)}} = A_{a_{(i,j)}} \wedge B_{p_{(i,j)}} = B_{a_{(i,j)}}\}$$

We measure the accuracy for classification as the percentage of correctly predicted color bins in the AB plane (1/400 random chance):

$$Acc = \frac{1}{N^2} \sum_{(i,j)}^{(N,N)} 1\{bin(A_{p_{(i,j)}}, B_{p_{(i,j)}}) = bin(A_{a_{(i,j)}}, B_{a_{(i,j)}})\}$$

Qualitative metrics:

We presented peers with a survey consisting of 50% synthesized images and 50% ground-truth images, asking them to categorize the images as either synthesized or real. The number of times a human is fooled into thinking a synthesized image is real is the important aspect to capture here:

$$Acc_{qual} = \frac{\text{\# of synthesized pictures deemed real}}{\text{\# of synthesized pictures shown}}$$

5.2 Analysis

	Quant.		Qual.
Model	**Train**	**Val**	**Acc**
Regression	0.01 %	0.01 %	34.6 %
Classification	**23.0 %**	**22.6 %**	57.7 %
With Color Rebalancing	19.2 %	17.7 %	**69.2 %**

Table 1: Models Results (Train: 45,000, Val: 10,000)

Figure 7 shows the loss and accuracy during the process of training the classification model over 75 epochs.

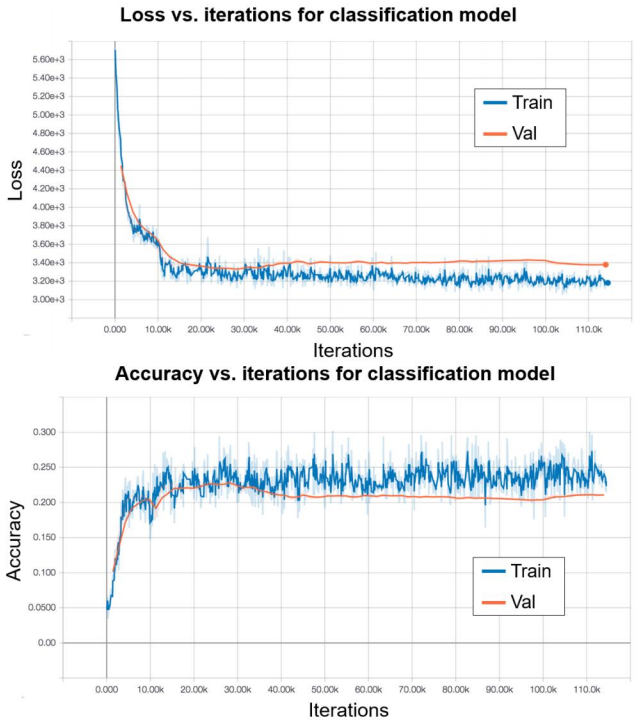

Figure 7: Training process for classification model over 75 epochs

In terms of the quantitative results, as expected, the accuracy results for regression are much worse than those for classification (and the color-rebalanced variant). This is because in the case of regression, the model needs to determine the exact a and b values for the example to be considered correct (65K+ choices), while in the case of classification, the problem is rescoped to classifying out of 400 possible bins instead, significantly improving the chances of a correct prediction. In the color rebalancing variant, the model is incentivized to make more exciting color choices, thus accuracy no longer holds up. In general, however, it is evident that getting high accuracy results for the colorization task is fundamentally difficult and it may therefore be prudent to also focus on the qualitative results.

From a qualitative standpoint (Figure 6), it is clear that all the models in general are able to capture the color distribution of the ground-truth image effectively. However, the regression model, which tends to predict in the unsaturated region of the AB plane to minimize the L2 loss, performs worse than the classification model. In comparison, color rebalancing proved very effective in generating more vibrant and realistic images because it inherently accounts for the skew toward mid-range (a,b) values in natural images. This is particularly evident from the results of the survey, which indicated that less colorful images were more often considered 'synthesized'. The state-of-art model in [11] achieves a 33% score in fooling participants while presenting ground-truth and output images side-by-side, a variant of our study that makes us confident our results are close in quality.

Ultimately, our model performed really well and in fact in some cases (see: horse and eagle), the rendered images look more 'real' than the ground-truth image. It effectively learned that the grass should be colored green, clouds should be white, etc. In terms of scope for improvement, the model performs poorly for instance on frogs, maybe due to a lack of contrast in the training images and a diverse color palette.

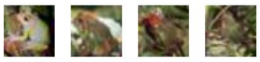

Figure 8: Examples of bad colorization results

6 Challenges

Some noteworthy challenges we encountered during the model development process are highlighted here:

1. The distribution of (a,b) values is skewed towards the mid-range, due to many of the `CIFAR-10` images containing backgrounds such as clouds, ground and walls. We had to ensure that the model was able to incorporate more rare/uncommon colors in its synthesized image.

2. The training set we currently utilize consists of small images (32×32). Although we optimized for faster training cycles, utilizing larger resolution images could have potentially helped the model learn more effectively.

3. Developing a model to train well on the `ImageNet` dataset proved to be difficult due to the large class distribution.

7 Conclusion and Future Work

The colorization task was reapproached from a classification perspective, where we predict colors in terms of bins that span the AB plane. Using a ConvNet and improving rendered colors through a color rebalancing technique achieved the best results fooling participants into believing 69.2% of the presented synthesized images are real.

In addition to utilizing transfer learning to improve the current ConvNet architecture, as well as incorporating prior knowledge [11], our future work will include implementing a Conditional GAN-based approach [6] to generate realistic colorized images to fool the discriminator.

8 Appendix

8.1 Contributions

During almost all of our coding and writeup sessions we sat together in one conference room to ensure everyone is contributing an equal amount to the project. We often split the work into several sub-problems that were tackled individually (divide and conquer, have different people use AWS in parallel to try different models) and later merge our findings.

8.2 Code

Please find out code on the following Bitbucket repository link : https://bitbucket.org/bardia/cs230-colorization-repo

References

[1] G. Charpiat, M. Hofmann, and B. Schölkopf. Automatic image colorization via multimodal predictions. In *European conference on computer vision*, pages 126–139. Springer, 2008.

[2] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[3] R. Dahl. Automatic colorization, 2016.

[4] A. Deshpande, J. Rock, and D. Forsyth. Learning large-scale automatic image colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 567–575, 2015.

[5] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 351–354. ACM, 2005.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[7] A. Krizhevsky, V. Nair, and G. Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM Transactions on Graphics (ToG)*, volume 23, pages 689–694. ACM, 2004.

[10] Y. Qu, T.-T. Wong, and P.-A. Heng. Manga colorization. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1214–1220. ACM, 2006.

[11] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[12] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.