Automatic Image Colorization

Oskar Eriksson
oseriks@kth.se

Gustave Rousselet
grou@kth.se

Abstract

In this paper we present a model for automatic colorization of grayscale images. The model works in the CIELAB (Lab) color space, which is discretized into 313 "color bins". The grayscale image is treated as the lightness channel, and the colorization is done using a convolutional neural network (CNN) which produces a categorical distribution over the color bins for each pixel. This distribution is then converted to a colorization using a decoding strategy. The model is trained using an augmented categorical cross entropy loss which promotes colorful and penalizes desaturated results. Two models were trained, one on a generic dataset containing a wide variety of images, and one on a dataset containing only images of dogs. Two experiments were performed. First, a questionnaire which asked human observers to compare colorizations of dogs using the two models. This experiment showed that the model trained only on dogs performed this task better, suggesting that it might be beneficial to use a specialized dataset for specialized tasks, rather than aiming to train a more generic colorizer. The second experiment involved using an image classifier (VGG-16-BN) pre-trained on the ImageNet dataset, and comparing its classification performance on different colorizations of the ImageNet validation set. These colorizations were the ground truth colorization, a random colorization and colorizations made by our two models. This experiment showed that the model was able to produce colorizations which outperformed a random colorization and came close to the ground truth performance, indicating that the colorization provides valuable information to the classifier. The conclusion was drawn that this approach to the colorization problem performs well on multiple colorization tasks.

1 Introduction

In this section, the colorization problem is introduced and an overview of our work is presented.

1.1 Colorization

Photography which mirrors the contemporary definition has existed since 1826 when the first black and white photograph was taken by Joseph Nicéphore Niépce. Until around the 1940s, black and white photography was the de facto standard and the color counterparts that did exist were of lower quality and usually did not capture the true colors of a scene. Because of this, there is an enormous amount of photography and videography from this time in black and white. In order for these photos and videos to capture the true view of a scene, they would need to be colorized, which can be done manually by a human. However, manually colorizing black and white photographs is a laborious task which requires significant skill and artistry. This manual colorization essentially requires that the colorizer detects objects and shapes which are then known to be associated with a certain color. This is a subjective process, and indeed objects which are of the same shape can have a broad spectrum of colors. For example, an apple can be green, yellow, or red, depending on its strain. The colorizer would have to be very familiar with the strain in order to choose an appropriate color from a grayscale image.

Project report for DD2424

In recent years, deep learning has proven to be a powerful tool in image processing tasks, ranging from image classification, object detection, and segmentation. There is a clear indication that CNNs are able to perform image processing tasks at near human levels of performance. Since the task of colorization is based on object detection and shape recognition, automatic colorization models could be adjacent to current image processing models. A model could be trained to perceive that the grass should be green, while the sky and the ocean should be blue. However, the problem is underconstrained in the sense that many objects can be colorized in multiple different ways as previously exemplified using the apples.

1.2 Contribution

In this project, we implemented the approach to this problem proposed by Zhang et al. (2016), where a grayscale image is provided as input and the output is a feasible colorization for the image, see Figure 1. We strive to generate a colorization which is supposed to look good to a human observer, and which could also benefit classification of grayscale images by a classifier pre-trained on color images.

The paper presents related work in image colorization, thereafter the data and methods used are presented. The methods section presents how the image colorization model is built and trained. The experiments section outlines two experiments used to evaluate the model, one using a human subjective assesment of the colorization, and one using a pre-trained image classifier to evaluate a colorized dataset compared to a ground truth and a random colorization. The conclusion section summarizes the results and concludes the paper.

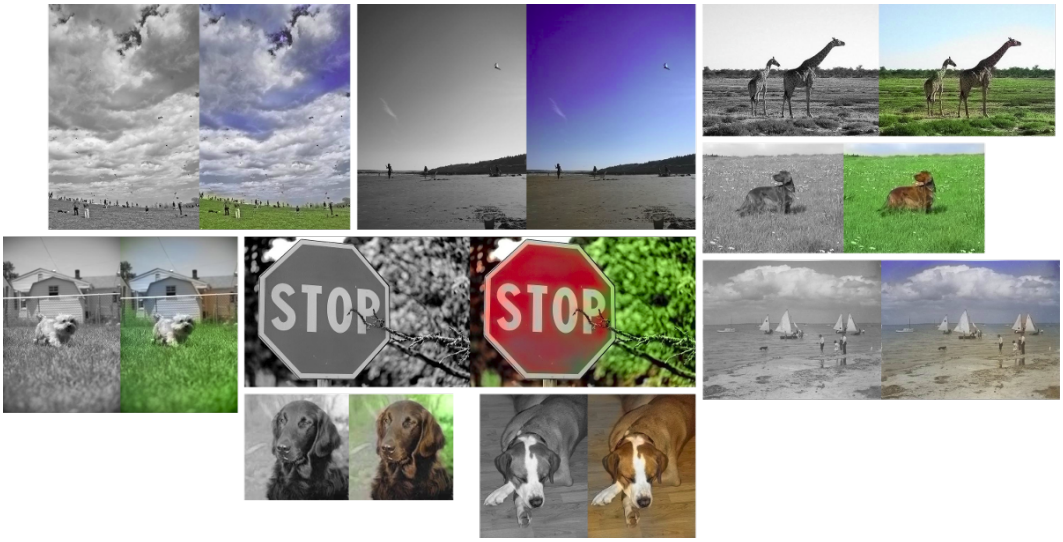

Figure 1: A selection of colorized images and their grayscale counterparts where our model performs well. The code is available at `https://github.com/oskaerik/colorization`.

2 Related Work

The main inspiration for this project is the paper *Colorful Image Colorization* by Zhang et al. (2016), who propose an approach to take a grayscale image and generate a possible colorization. Their approach is based on using a CNN to generate a distribution of colors for each pixel of the image, and then using a decoding function to yield the final colorization. They propose a loss function and a decoding function which promotes colorful results, hence the title of the paper.

Zhang et al. (2016) use three metrics to evaluate the performance of their model: perceptual realism, which basically measures how often a human would be fooled to believe that the colorized image was the ground truth image, and vice-versa; semantic interpretability, which measures how well a pre-trained classifier performs on a colorized set of images compared to their ground truth counterparts; and finally raw accuracy which measures pixel-by-pixel how close the colorizations are to their ground

truth counterparts. They conclude that their approach yields results which are hard to distinguish from actual color photographs.

3 Data

We trained two models on two different datasets, to be able to compare the two. The first dataset is specialized on dog images, called *Stanford Dogs* by Khosla et al. (2011), which contains about 20,000 images of dogs of 120 different species. The other dataset is *Microsoft COCO* by Lin et al. (2014), which is a more general dataset containing 200,000 images of common objects from 80 different categories. We also used the *ImageNet* validation set by Deng et al. (2009) to evaluate the models, which contains 50,000 images of generic images from 1,000 different categories.

We chose these datasets since we wanted to see if we could get better colorization of dogs by training exclusively on dogs, or if it would be better to train on objects of different types. We also wanted to see whether training only on dogs would be useful to colorizing other objects as well.

Naturally, images lacking color channels were removed before training the models. The images were resized using bilinear interpolation to fit our method, and the grayscale input was normalized so that each pixel has a value between -1 and 1.

4 Methods

The method that we use is heavily inspired by Zhang et al. (2016), and we use their notation to a large extent. The problem is to take a grayscale image of $H \times W$ pixels and find a possible colorization of the image. We are working in the Lab color space, where a color image consists of three channels: the lightness channel *L*, the green-red channel *a* and the blue-yellow channel *b*. We denote the *L* channel $\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$ and the *ab* channels $\mathbf{Y} \in \mathbb{R}^{H \times W \times 2}$.

One can imagine that there is an underlying mapping $\mathbf{X} \mapsto \mathbf{Y}$. We try to approximate this mapping, i.e. we try to find a mapping $\mathcal{F}$ such that

$$\mathcal{F} : \underbrace{\mathbb{R}^{H \times W \times 1}}_{\text{true } L \text{ channel } \mathbf{X}} \longrightarrow \underbrace{\mathbb{R}^{H \times W \times 2}}_{\text{approximated } ab \text{ channels } \widehat{\mathbf{Y}}} . \tag{1}$$

After applying $\mathcal{F}$, we stack $\mathbf{X}$ and $\widehat{\mathbf{Y}}$ to get the final colorized image.

Zhang et al. (2016) treats the mapping $\mathcal{F}$ as a composition of mappings: $\mathcal{F}(\mathbf{X}) = (\mathcal{G} \circ \mathcal{H})(\mathbf{X}) = \widehat{\mathbf{Y}}$, where $\mathcal{G}$ is a CNN and $\mathcal{H}$ is a decoding function. This is the approach we use as well, and these steps are described in this section.

4.1 CNN

Zhang et al. (2016) proposes a way to treat the problem as multiclass classification. The *ab* color space is quantized into a grid where each bin has size 10×10, which yields $Q = 313$ valid color bins. The goal is to find a categorical distribution for each pixel of the input image which gives a probability for the pixel belonging to each of these Q classes. However, the CNN outputs $H' \times W'$ pixels which is smaller than the original image. This is solved by using bilinear interpolation in the decoding stage, i.e. resizing the resulting color channels. That is, we train the CNN to find a mapping

$$\mathcal{G} : \underbrace{\mathbb{R}^{H \times W \times 1}}_{\text{true } L \text{ channel } \mathbf{X}} \longrightarrow \underbrace{[0,1]^{H' \times W' \times Q}}_{\text{categorical distribution } \widehat{\mathbf{Z}}} . \tag{2}$$

We need to compare this categorical distribution $\widehat{\mathbf{Z}}$ to some "ground truth" distribution. To do this, Zhang et al. (2016) proposes a soft-encoding scheme $\mathbf{Y} \mapsto \mathbf{Z}$, which we use as well. For each pixel in $\mathbf{Y}$, we find the 5-nearest neighboring bins, then we create the categorical distribution $\mathbf{Z}$ by measuring the distance of these bins from the true *ab* values using an RBF kernel with $\sigma = 5$ and normalize.

We used the same CNN architecure as Zhang et al. (2016), which is shown in Table 1. We also train the CNN the same way as Zhang et al. (2016): using the data $\mathcal{D} = \{(\mathbf{X}, \mathbf{Z})\}$, with weighted categorical cross entropy loss

$$L(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\sum_{h=1}^{H}\sum_{w=1}^{W} v(\mathbf{Z}_{h,w}) \sum_{q=1}^{Q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q}), \tag{3}$$

where $v(\mathbf{Z}_{h,w})$ is a weighting factor used to promote rare colors and prevent desaturated results. Desaturated colors are much more common than saturated colors in a majority of photographs, so weighting is necessary to be able to produce colorizations which not only consists of desaturated colors. This is equivalent to the common approach of class rebalancing when having a skewed dataset.

To calculate the weighting factor, we use the same approach as Zhang et al. (2016). First we find the empirical distribution the bins $\mathbf{p} \in [0, 1]^Q$ by simply reading in the entire dataset and dividing the occurrences of each bin with the total number of pixels. We smooth this distribution using a Gaussian filter with $\sigma = 5$, which yields the smoothed distribution $\widetilde{\mathbf{p}}$. Finally, we add $\widetilde{\mathbf{p}}$ and a uniform distribution $\frac{1}{Q}$ and take the inverse, we call this $\mathbf{w}$. Finally, we normalize so that $\mathbb{E}[\mathbf{w}] = \sum_{q=1}^{Q} \widetilde{\mathbf{p}}_q \mathbf{w}_q = 1$. Then for a given pixel, we find the closest *ab* bin q^* and set $v(\mathbf{Z}_{h,w}) = \mathbf{w}_{q^*}$.

Training the CNN was done similarly to Zhang et al. (2016), using the Adam optimizer with learning rate $= 3 * 10^{-5}$, $\beta = (0.9, 0.99)$ and weight decay $= 10^{-3}$. We trained two models on two different datasets, one being the COCO dataset and one being the Stanford Dogs dataset. We trained until the training loss slowed down to $\approx 1\%$ improvement per epoch, which resulted in 10 epochs for the COCO dataset and 30 epochs for the Stanford Dogs dataset. We had to limit training because of limited resources and due to the time constraints of the project, otherwise we would have trained more and also experimented with adjusting the learning rate as training progressed.

T	**X**	**C**	**K**	**S**	**P**	**D**
Input	224×224	1	-	-	-	-
Convolution	224×224	64	3	1	1	1
Convolution	112×112	64	3	2	1	1
BatchNorm	112×112	64	-	-	-	-
Convolution	112×112	128	3	1	1	1
Convolution	56×56	128	3	2	1	1
BatchNorm	56×56	128	-	-	-	-
Convolution	56×56	256	3	1	1	1
Convolution	56×56	256	3	1	1	1
Convolution	28×28	256	3	2	1	1
BatchNorm	28×28	256	-	-	-	-
Convolution	28×28	512	3	1	1	1
Convolution	28×28	512	3	1	1	1
Convolution	28×28	512	3	1	1	1
BatchNorm	28×28	512	-	-	-	-
Convolution	28×28	512	3	1	2	2
Convolution	28×28	512	3	1	2	2
Convolution	28×28	512	3	1	2	2
BatchNorm	28×28	512	-	-	-	-
Convolution	28×28	512	3	1	2	2
Convolution	28×28	512	3	1	2	2
Convolution	28×28	512	3	1	2	2
BatchNorm	28×28	512	-	-	-	-
Convolution	28×28	256	3	1	1	1
Convolution	28×28	256	3	1	1	1
Convolution	28×28	256	3	1	1	1
BatchNorm	28×28	256	-	-	-	-
Deconvolution	56×56	128	4	2	1	1
Convolution	56×56	128	3	1	1	1
Convolution	56×56	128	3	1	1	1
Convolution	56×56	313	1	1	0	1

Table 1: The CNN architecture. **T** is the layer type, **X** is the size of the output, **C** is the number of output channels, **K** is the kernel size, **S** is the stride, **P** is the padding and **D** is the dilation. We used the same dimensions for the images as Zhang et al. (2016), which is $H = W = 224$ and $H' = W' = 56$.

4.2 Decoding

The CNN outputs a categorical distribution $\widehat{\mathbf{Z}}$, but we want to find a colorization $\widehat{\mathbf{Y}}$. We use the same approach as Zhang et al. (2016), where we define a decoding function $\mathcal{H}$ such that

$$\mathcal{H}: \underbrace{[0,1]^{H' \times W' \times Q}}_{\text{categorical distribution } \widehat{\mathbf{Z}}} \longrightarrow \underbrace{\mathbb{R}^{H \times W \times 2}}_{\text{approximated } ab \text{ channels } \widehat{\mathbf{Y}}} . \tag{4}$$

The decoding function $\mathcal{H}$ has three steps. First, we apply the annealed-mean operation proposed by Zhang et al. (2016) to $\widehat{\mathbf{Z}}$, which makes the the distribution more concentrated around the bins with the largest probabilities. Annealed-mean is defined as applying $\mathrm{softmax}(\widehat{\mathbf{Z}}_{w,h}/T)$ to each pixel w, h, where T is a temperature parameter. Zhang et al. (2016) found that $T = 0.38$ is a good value, so that is what we used as well. Second, we take the expected value for each pixel, i.e. we take the sum of each bin weighted by its probability. Finally, we use bilinear interpolation to resize the image back to the size of the lightness channel *L*.

Zhang et al. (2016) discusses the purpose of the annealed-mean operation. Taking the expected value for each pixel without applying annealed-mean results in spatially consistent but desaturated results. Taking the mode for each pixel results in colorful images but spatially inconsistent results. The purpose of annealed-mean is to get a trade-off between these two approaches, which results in colorful and spatially consistent results.

5 Experiments

We trained two models: one on the COCO dataset and one on the Stanford Dogs dataset. We then performed two major experiments with these two models. In the first experiment, we asked people which model they thought were the best for colorizing dog images. In the second experiment, we colorized the ImageNet validation set and the Stanford Dogs test set, and evaluated the performance of a pre-trained classifier on the colorized images. These experiments and their outcomes are presented in this section.

5.1 Questionnaire

We sampled 30 random images from the Stanford Dogs images which had not been used in training, and colorized these images with each model respectively. We created an online forced-choice questionnaire where we presented each image pair and asked the participant which colorization was the most appealing. We asked the participants to not look at the image pair for more than five seconds, since we wanted them to go with their intuition and not overanalyze the images.

In total, 73 participants answered the questionnaire. In 21 out of the 30 images, there was a majority who preferred the model trained exclusively on dog images, and in the other 9 cases the majority preferred the model trained on the more generic dataset. In total, 68% of the votes went toward the model trained on dogs. The results for each image are visualized in Figure 2

The image where most people preferred the COCO model can be seen in Figure 3. What is interesting in this image is that it is shot indoors, and there is a human present. In the Stanford Dogs dataset, most pictures are shot outside and with only a dog. There is reason to believe that the colorizer trained on the COCO dataset is better at generating visually appealing colorizations for other objects than dogs, since it has seen a wider variety of objects during training.

In Figure 4, the image were most people preferred the model trained on Stanford Dogs is shown. In this case, it seems as if the model trained on COCO treats the dog as vegetation, while the Stanford Dogs model produced a more realistic colorization.

Finally, in Figure 5 the image which had the most even distribution of votes between the two models is shown.

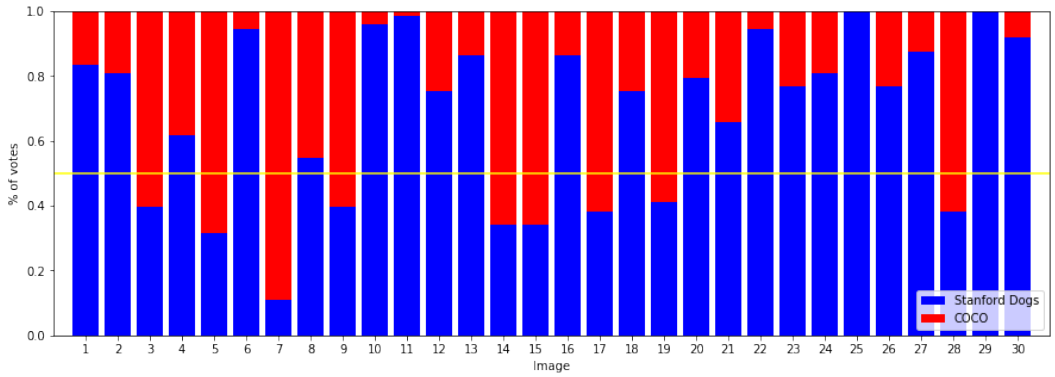

Figure 2: The distribution of votes for each of the 30 images in the questionnaire.

Figure 3: The image where most people prefered the model trained on COCO.

Figure 4: The image where most people prefered the model trained on Stanford Dogs.

Figure 5: The image which had the most even distribution of votes.

5.2 Pre-trained classifier

The questionnaire from the previous experiment provides qualitative results insight into the performance of our colorization from a subjective human perspective. In order to asses the colorization from a quantitative perspective, we used a pre-trained classifier and compared its classification results on our colorizations vs. a ground truth and random colorization dataset. The reasoning behind this experiment is that if our colorizations leads to performance gains above that of a random colorization, it indicates that our colorization provides valuable information to the classifier. Furthermore, if our colorization comes close to the ground truth dataset performance, this indicates that our colorization is providing highly valuable (and by extension more realistic) information. This experiment is very similar the semantic interpretability experiments performed by Zhang et al. (2016), where they used an off-the-shelf classifier to evaluate their colorizations.

The classifier used was VGG-16 with batch normalization (VGG-16-BN) by Simonyan & Zisserman (2014). This classifier was chosen to match the classifier used in the original colorization paper by Zhang et al. (2016), and also because it is a well known classifier for the ImageNet dataset by Deng et al. (2009). It uses relatively standard image classification techniques and achieves a high performance on the ImageNet dataset.

The results are presented in Table 2. The ground truth image and a random colorization were included as baselines. For the random colorization, a random color bin was picked for each pixel. Top-1 class accuracy was used on the ImageNet and Stanford Dogs validation datasets.

Colorizer / Dataset	ImageNet	Stanford Dogs
COCO	57.04%	56.85%
Stanford Dogs	58.25%	60.00%
Random	39.87%	41.07%
Ground truth	73.36%	77.85%

Table 2: Accuracy achieved by the pre-trained VGG-16-BN classifier on colorizations and ground truth.

For both datasets and colorization models, the results clearly indicate that the colorization models are able to provide valuable information to the classifier due to the resulting validation accuracies being $\approx 20\%$ higher than a random colorization. Furthermore, the validation accuracies are $\approx 15\%$ lower than the ground truth images, meaning that the colorizations come close to a true colorization. This is under the assumption that the classifier's "perception" of the image somewhat matches that of true human perception.

Perhaps the COCO colorizer would need more training since the dataset is much more diverse. The Stanford Dogs colorizer seems to go a "safer" route with objects that it has not seen, basically yielding almost grayscale images, see Figure 6. The COCO colorizer seems to be more "daring" in its colorizations even though they might turn out wrong, see Figure 7. Therefore, one explanation to the Stanford Dogs model colorizations yielding higher accuracies on the ImageNet dataset could be that the classifer is better at classifying grayish image compared to wrongly colorized images.

Figure 6: The Stanford Dogs colorizer taking a safe route.

Figure 7: The COCO colorizer being more daring.

6 Conclusion

From the results, it is clear that the colorization models provide information valuable to an image classifier. The results from Table 2 clearly show that both the colorization models outperform a random colorization, and more so indicate that the performance on the colorizations are close to the performance on the ground truth, giving us reason to believe that our colorizers yield realistic colorizations. Our results in Figure 2 suggests that it could be beneficial to train the model on the type of images that it will be used for, instead of going for a more generic model. However, we do not know how the models would have performed given more training, and this is something that would need to be investigated further. However, it does seem that for images which are not typical for the dataset, as e.g. Figure 3 where the image is taken indoors and also features a human, it could be good to train the model on a more generic dataset.

References

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009), ImageNet: A Large-Scale Hierarchical Image Database, *in* 'CVPR09'.

Khosla, A., Jayadevaprakash, N., Yao, B. & Fei-Fei, L. (2011), Novel Dataset for Fine-Grained Image Categorization, *in* 'First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition', Colorado Springs, CO.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014), Microsoft COCO: Common Objects in Context, *in* 'Computer Vision – ECCV 2014', Springer International Publishing, Cham, pp. 740–755.

Simonyan, K. & Zisserman, A. (2014), 'Very Deep Convolutional Networks for Large-Scale Image Recognition'.
URL: *https://arxiv.org/abs/1409.1556*

Zhang, R., Isola, P. & Efros, A. A. (2016), 'Colorful Image Colorization'.
URL: *http://arxiv.org/abs/1603.08511*