Colorizing Grayscale Images

Emil Gustavsson emigust@kth.se **Gabriel Ellebrink** elleb@kth.se

Gabriel Rosenberg grosenb@kth.se **Rasmus Rudling** rrudling@kth.se

Abstract

The goal of this paper is to study how a grayscale image can be colorized using a convolutional neural network (CNN). The method presented in the paper *Colorful Image Colorization*, written by R. Zhang et al., was used as reference [1]. The data used to train the model presented in this paper is confined to portrait images of people's faces. The dataset used in this study is thus more narrow than what is used in the original paper, which is a wide variety of image categories. Two different loss functions are implemented with results showing that a multinomial cross-entropy loss function produces much better results than a euclidian loss function. Although it has been trained for far fewer iterations, the model presented in this paper colorizes portrait images more realistically in 58% of the cases when compared to the model presented by R. Zhang et al. in [1] in a qualitative study.

Keywords: Colorization, CNN, Computer Vision.

1 Introduction

Colorization of grayscale images is a technique that can be used in many applications. As described in [2], a colorized image can be useful within medical fields to easier interpret magnetic resonance imaging (MRI). Another area of application is for entertainment purposes. As an example, R. Zhang et al. used their trained model to colorize grayscale legacy images [1]. The goal of their model is not to produce ground truth colors. Instead, they aim to output a plausible colored picture that does not look artificially made. To get a deeper knowledge of how to colorize grayscale pictures we have tried to replicate the work of R. Zhang et al. A CNN was implemented that, given the lightness channel L, predicts the color descriptive a- and b-channels in the *CIELAB* colorspace. To make training easier, the image dataset was narrowed down to only include pictures of human faces. Two versions of the model were trained, one with Euclidean loss and the other with Multinomial cross-entropy loss. The result of the two models clearly tells that a Multinomial cross-entropy loss function is better suited for our model. Through a qualitative study, it was shown that the images that were colorized by the model presented in this paper were more convincing than the same portrait images colorized by the original model in 58% of the cases.

2 Previous work

Deep learning methods have proven to be a powerful tool in computer vision, and are used in many papers that explore the process of colorizing images. As an example, Q. Zhang et al. uses a CNN to automate animation colorization [3]. Their goal is to create a color mapping from a reference image to subsequent image frames that together result in a colored animation. For instance, they could have one colored image and 30 subsequent grayscale images. Their goal was to color the 30 subsequent images with the help of the first reference image. This could be compared with exemplar-based colorization, where one approach is presented by M. He et al. in their paper *Deep Exemplar-based*

Colorization [4]. To colorize a grayscale image, they give the model a reference image that influences the colors of the black and white image.

S. Khodadade et al. explored object recoloring in their paper *Automatic object recoloring using adversarial learning* [5]. They provide a method to combine natural language processing (NLP) with a generative adversarial network (GAN) to allow a user to input an object they want to recolor and to what color [5].

3 Data

The dataset used in this paper is called *Labeled Faces in the Wild* and is provided by the Computer Vision Laboratory at the University of Massachusetts [6]. Originally, it serves as a public benchmark for face verification. In this paper, the data processing resources were limited. Therefore, this dataset of people was chosen with the intention that a more niche dataset would ease the learning for our model.

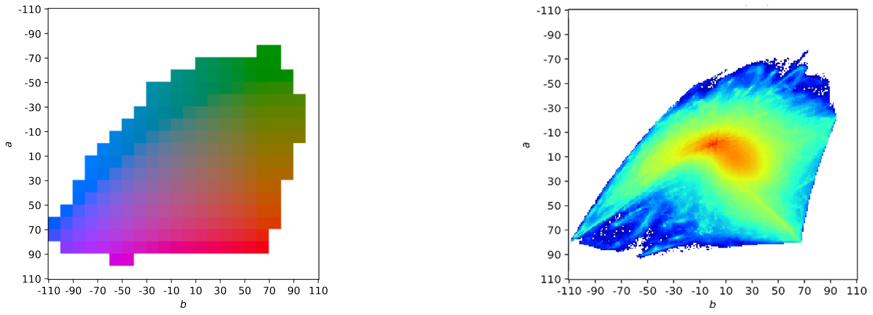
The dataset contains 13 233 images where each image has 250 × 250 pixels, 96 dpi in both horizontal and vertical resolution with a bit depth of 24. The implemented CNN takes in images that are 256 × 256 pixels. Therefore, to make the images compatible they are up-scaled to 256 × 256 pixels.

4 Methods

In order to convert a grayscale image to a colorized RGB image, the *CIELAB* color space is utilized. *CIELAB* has three channels; *L*, *a*, and *b*. The grayscale image provided as input to the network represents the *L*-channel. The goal is to predict the *a*- and *b*-channels in order to compose an artificially colorized image. In mathematical terms, given an input *L*-channel $\mathbf{X} \in \mathbb{R}^{H\times W\times 1}$, we want to find a mapping $\widehat{\mathbf{Y}} \in \mathcal{F}(\mathbf{X})$, where $\widehat{\mathbf{Y}} \in \mathbb{R}^{H\times W\times 2}$ and H, W are the dimensions of the image.

4.1 Simplification of the color space

In order for the model to learn efficiently with a limited dataset, the number of colors (labels the dataset) was reduced. This was done by analyzing the ab-values of the image dataset and then discretizing the colors into bins of size 10x10. Figure 2(b) shows a heatmap over all existing colors in the dataset and their rarity. Each bin in Figure 2(a) corresponds to 100 of these possible colors. In total, we get $\mathbf{Q} = 247$ which is the number of bins used to represent the colors of the image dataset.



(a)

(b)

Figure 1: (a) $RGB(a, b\,|L = 50)$. Discretized *ab* color space with a grid size of 10 created from 13,233 images. $Q = 247$. (b) Probability distribution of *ab* values, shown in log scale.

Before the final prediction $\widehat{\mathbf{Y}}$, the model learns a mapping $\widehat{\mathbf{Z}} \in [0, 1]^{H\times W\times Q}$ where $\widehat{\mathbf{Z}}[h][w][q]$ describes the probability that the q^{th} possible discretized color represents the correct color in the pixel.

4.2 Ground truth color encoding

During training, the model needs to be able to compare the color probability distribution $\widehat{\mathbf{Z}}$ with the ground truth colors $\mathbf{Y}$. As proposed by R. Zhang et al. in [1], a comparison can be made by defining a function $\mathbf{Z} = \mathcal{H}_{gt}^{-1}(\mathbf{Y})$, using a soft-encoding scheme. Initially, they mention that each ground-truth value $\mathbf{Y}_{h,w}$ could be encoded as a 1-hot vector $\widehat{\mathbf{Z}}_{\mathbf{h},\mathbf{w}}$ instead of soft-encoding. While using a 1-hot vector is trivial and an easy-to-understand solution, they found that using soft-encoding yielded a network that could learn faster. The soft-encoding used by R. Zhang et al. and in this paper finds the 5-nearest neighbors to $\mathbf{Y}_{h,w}$ in the discretized *ab* output space. It is then weighted using a Gaussian kernel with $\sigma = 5$ proportionally to their distance from the ground truth *ab* color.

4.3 Multinomial cross entropy loss

R. Zhang et al. emphasized that the type of loss function is of great importance to achieve a vibrant image result. Two previous colorization networks mentioned in the paper have used the Euclidean loss function between predicted and ground truth colors, and according to R. Zhang et al., this loss function will favor the mean of the possible colors which yields a desaturated image. Instead, a multinomial cross-entropy loss function is used:

$$L(Z, \widehat{Z}) = -\sum_{w,h} v(Z_{w,h}) \sum_{q} Z_{w,h,q} log(\widehat{Z}_{w,h,q}) \tag{1}$$

where $v(Z_{h,w})$ is a weighting term that is used to rebalance the loss based on how rare a specific *ab* value is. The function is defined as follows:

$$v(Z_{w,h}) \propto \left((1-\lambda)\tilde{\mathbf{p}} + \frac{\lambda}{Q}\right)^{-1} \tag{2}$$

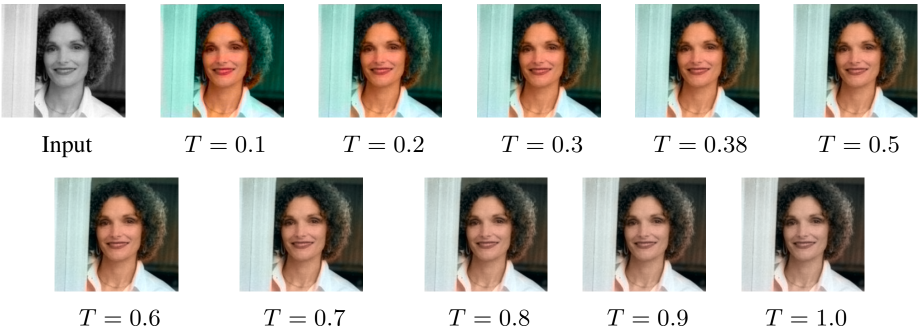
We denote $\mathbf{p}$ as a discretized version of the probability distribution of *ab* values visualized in Figure 2(b). That is $\mathbf{p} \in [0, 1]^Q$. As R. Zhang et al. suggests in [1], $\mathbf{p}$ is smoothed with a Gaussian kernel with $\sigma = 5$ and is now denoted as $\tilde{\mathbf{p}}$. Then, $\tilde{\mathbf{p}}$ is mixed with a uniform distribution with weight $\lambda = 0.5$. As a last step, the distribution is normalized with the term $\frac{\lambda}{Q}$ such that the expected value of the weighting factor $v(Z_{h,w})$ is 1.

4.4 From color probabilities to final color prediction

After obtaining the final $\widehat{\mathbf{Z}}$ for a specific image, the prediction of quantified color values is transformed to ab-space. The final prediction of the image ab-values is denoted as $\widehat{\mathbf{Y}}$. One way to obtain $\widehat{\mathbf{Y}}$ from $\widehat{\mathbf{Z}}$ is to take the mode of the predicted distribution for each pixel. Another option is to take the mean of the prediction. Due to issues with both these methods, R. Zhang et al. instead introduces a third option referred to as the annealed-mean of the $\widehat{\mathbf{Z}}$ distribution:

$$H(Z_{h,w}) = E[f(Z_{h,w})] \qquad f(z) = \frac{exp(log(z)/T)}{\sum_q exp(log(z)/T)} \tag{3}$$

This function is very similar to the softmax function but with the difference that it can create a clearer distinction between small and large values depending on the T parameter. A value close to 1.0 results in a mean of the distribution and with T close to 0 the method instead resembles the mode of the distribution. This means that T can be adjusted to get something in between. R. Zhang et al. set T to 0.38. However, $T = 0.6$ seemed to give the best result for the presented model on the dataset in this paper.



Input $T = 0.1$ $T = 0.2$ $T = 0.3$ $T = 0.38$ $T = 0.5$

$T = 0.6$ $T = 0.7$ $T = 0.8$ $T = 0.9$ $T = 1.0$

Figure 2: How the temperature parameter T in the annealed-mean affects the result. In the original paper [1] $T = 0.38$ is used, in this paper $T = 0.6$.

4.5 CNN architecture

In Figure 3 the architecture of our CNN is shown and is meant to replicate the network of R. Zhang et al.

Layer	C	S	D	BN	L
conv1_1	64	1	1	-	-
conv1_2	64	2	1	✓	-
conv2_1	128	1	1	-	-
conv2_2	128	2	1	✓	-
conv3_1	256	1	1	-	-
conv3_2	256	1	1	-	-
conv3_3	256	2	1	✓	-
conv4_1	512	1	1	-	-
conv4_2	512	1	1	-	-
conv4_3	512	1	1	✓	-
conv5_1	512	1	2	-	-
conv5_2	512	1	2	-	-
conv5_3	512	1	2	✓	-
conv6_1	512	1	2	-	-
conv6_2	512	1	2	-	-
conv6_3	512	1	2	✓	-
conv7_1	512	1	1	-	-
conv7_2	512	1	1	-	-
conv7_3	512	1	1	✓	-
conv8_1	256	0.5	1	-	-
conv8_2	256	1	1	-	-
conv8_3	256	1	1	-	✓

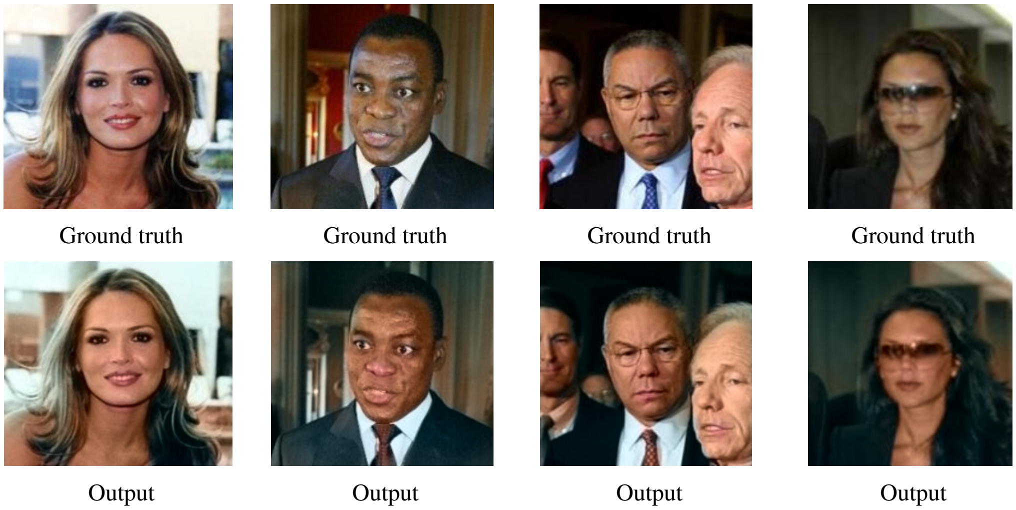
Figure 3: CNN architecture. **C** = number of channels of output, **S** = computation stride, **D** = kernel dilation, **BN** = whether BatchNorm layer was used after layer, **L** whether a 1x1 conv and cross-entropy loss layer was imposed.

4.6 Training

Due to problems using Google Cloud Platform for training we had to limit the batch size and the number of iterations. We trained our network with batches of size 1 for 2 iterations compared to 40 and 450 000 respectively, as in the original paper. The same optimizer settings were applied; an ADAM solver with $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay $= 10^{-3}$ and learning rate $= 3 \times 10^{-5}$.

5 Experiments

In Figure 4 some examples of pictures colorized by our network compared to the ground truth are displayed. In many cases, our networks do not create a fully realistic colorization which was expected due to the small amount of training. However, even though the network was trained with few iterations, it is still capable of distinguishing faces from the background and color them with relatively realistic tones. The model seemingly have difficulties with coloring details in the background. This could be an effect from that many backgrounds in the training set have desaturated colors, and that the model might be trained with too few iterations to be able to distinguish details.


Ground truth Ground truth Ground truth Ground truth

Output Output Output Output

Figure 4: Comparison between the original images and the outputs from the presented trained model using $T = 0.6$.

5.1 Replace the loss function with Euclidean loss, L_2 function

In [1], R. Zhang et al. discuss why it is a better idea to use a rebalanced multinomial cross-entropy loss function (1) instead of a Euclidean loss, L_2 function (4). If there is a distinct set of ab colors the optimal solution to the euclidean loss will be the mean of the set. This would result in ab values close to zero, which would make the model favor grayish and desaturated colors. The model is trained on images of human faces that in many cases have desaturated colors, which makes it interesting to see if a model trained with a euclidean loss function would perform as well as the one with a rebalanced multinomial cross-entropy loss function.

$$L_2(\widehat{Y}, Y) = \frac{1}{2}\sum_{q} \|Y_{w,h} - \widehat{Y}_{w,h}\|_2^2 \tag{4}$$

In Figure 5 the effects of using different loss functions during training are presented. The result shows that using a L_2 loss function will favor the same desaturated mean value of the ab channels over the whole picture, and fail to distinguish the face and other details in the picture.

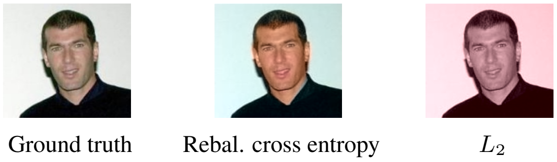
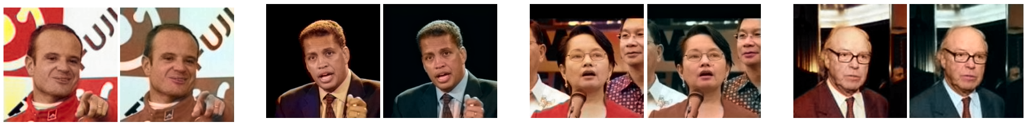

Ground truth Rebal. cross entropy L_2

Figure 5: Comparison between the ground truth and the result from using different loss functions during training.

5.2 Comparison to the original paper

The model presented by R. Zhang et al. was trained on a wide variety of image subjects and thus also seems to perform well on most images. When testing our model with images of other subjects than faces it is clear that our network performs worse than the original. This is not a surprise since we only train on images of a single subject and also uses a considerably smaller dataset with far fewer iterations during training. Yet, our performance on portrait images is surprisingly good compared to the same images colorized by the original network. Figure 6 shows a comparison of the two different networks on random images from our dataset *Labeled Faces in the Wild* [6].



1(a) 1(b) 2(a) 2(b) 3(a) 3(b) 4(a) 4(b)

Figure 6: Comparison between the outputs from the model presented by R. Zhang et al. in [1] (a) and the model presented in this paper (b).

5.2.1 Qualitative evaluation

In order to compare which network generates the most realistic images, we performed a qualitative study, where participants got to decide which one of two colorized images looked the most realistic. During the study, the participants were shown 50 images colorized by the two different models. Their task was to select the image they thought looked the most realistic without knowing anything about how the images were generated.

Participant	**This paper**	**R. Zhang et al. in [1]**
1	27	23
2	23	27
3	29	21
4	31	19
5	33	17
6	31	19
In total	174 (58%)	126 (42%)

Figure 7: Results from qualitative evaluation. The second column refers to the model used in this paper and the third column refers to the model used by R. Zhang et al. in [1].

The results, shown in Figure 7, tell that our model clearly yields more realistic colorization according to the participants. After the test, many participants said that the images from the original model were unrealistically vibrant in unexpected areas of the image (without knowing which image belonged to what model), while the images generated by our model often were less saturated.

6 Conclusion

This paper has highlighted different aspects of artificial image colorization. The presented CNN model was trained on 13 233 images from the *Labeled Faces in the Wild* dataset which consists of facial images of celebrities in various environments. While the model was only trained for two iterations on the dataset, it still seems to output convincing results.

A qualitative evaluation was held to assess the results from the presented model in comparison with the original paper's model. The test images used were from the *Labeled Faces in the Wild* dataset, i.e the dataset that the presented model was trained on. The original paper's model was trained on the ImageNet which is more diverse than *Labeled Faces in the Wild*, consisting not only of faces [6][7]. Six people participated in the qualitative evaluation where they were shown two versions of 50 images, where one was produced from the presented model and one from the original paper's model. They were asked which of the two artificially colored images was most realistic. The result was that the participants preferred the images produced by this paper's model by 58% on average. While we realize that six participants are not enough to draw definitive conclusions, we believe that the presented model produced satisfying results.

For future studies, we think it would be interesting to train our model with more iterations. Considering that the presented model was only trained for two iterations in comparison to the approximate 450 000 training iterations of the original paper's model, we believe that our presented solution has great potential and room for improvement.

References

[1] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, pages 649–666, Cham, 2016. Springer International Publishing.

[2] Jiancheng An, Koffi Gagnon Kpeyiton, and Qingnan Shi. Grayscale images colorization with convolutional neural networks. *Soft Computing*, 24:4751–4758, April 2020.

[3] Qian Zhang, Bo Wang, Wei Wen, Hai Li, and Junhui Liu. Line art correlation matching feature transfer network for automatic animation colorization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3872–3881, January 2021.

[4] Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, and Lu Yuan. Deep exemplar-based colorization, 2018.

[5] Siavash Khodadadeh, Saeid Motiian, Zhe Lin, Ladislau Boloni, and Shabnam Ghadar. Automatic object recoloring using adversarial learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1488–1496, January 2021.

[6] Computer Vision Laboratory at the University of Massachusetts. Labeled faces in the wild.

[7] Princeton University Stanford Vision Lab, Stanford University. Imagenet.