Image Colorization with DenseNet

Bugrahan Akbulut
Department of Computer Engineering
Hacettepe University, Ankara, TURKEY
akbulutbugrahan[at]gmail.com

Gokay Atay
Department of Computer Engineering
Hacettepe University, Ankara, TURKEY
gokayatay96[at]gmail.com

Abstract

Recent works have shown us convolutional neural networks are the proper solution for many visual tasks, including image colorization. In this paper, we try to explain our model, evaluation metrics, methodology, experimental results, set-up, and future works. For colorization problem it is really hard to calculate color channels without any user interaction. To solve this problem, we researched a lot of solution and in this study, we propose using different CNN architecture for the model to calculate color channels for a given gray image.

1. Introduction

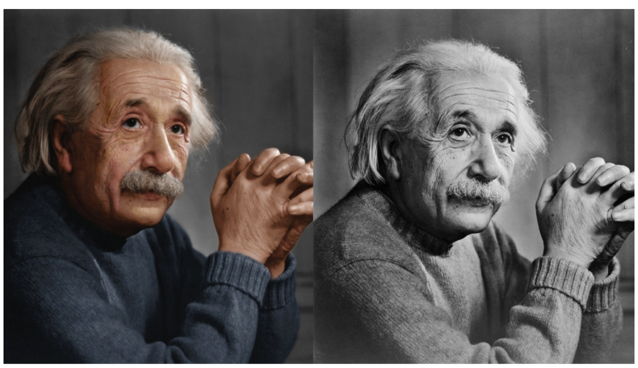


Figure 1: Colorized and Grayscale Image of Einstein

Historical pictures, black and white movies etc. are both consists tones of gray images. Asking the question, can we re-visualize these images as colorized images, forms the study of image colorization. Image colorization assigns colors to a gray-scale image. Colorization of gray-scale image without any user interaction is nearly impossible. It requires some preliminary information about image such as background color, objects in that image, texture information, edges, lines etc. For image colorization problem, there have been three different methods that can be considered in the past years. These three methods are scribble-based colorization, example-based colorization and learning-based colorization. We will go into detail about this methods and the works done in the next section. Simply, pixel intensities of a image depends on different characteristics of image such as edges, lines, structures etc. If we apply learning to these, we can predict the colorized image approximately.
In the most of scene images, sky is blue or grass is green, not always but most of the time. Without user interaction we can not gather these kind of preliminary information unless we dont use deep learning. Today, for the most visual tasks deep learning models work almost faultless, even for some cases they might work better than humans and this motivates us to explore its potential application in our context. Since recent works have shown us deep learning architectures are the proper solution for many visual tasks, by training our DenseNet-like model with some subset of ImageNet dataset we aim to overcome the image colorization problem.

2. Related Works

For image colorization problem, there have been three different methods that can be considered in the past years. These three methods are scribble-based colorization, example-based colorization and learning-based colorization. In recent years several studies have been conducted in this field and also there has been corresponding progress. Our paper is closely related to learning-based image colorization method. In this section, we review some other works, methods which are related to image colorization problem.

Scribble-Based Colorization

One of the earliest approaches for image colorization problem is scribble-based colorization. We think that scribble-based colorization is the approach that requires the most human work. In scribble-based colorization, artists paints small colored lines on the grayscale images and then these images are used to colorize grayscale image. Close neighbor pixels are colored with the same color. One of the works are conducted by Levin, A., et al. [6]. Like we defined, in this work, they used some images with color scribbles generated by artists and they colored pixels with similar intensities and close pixels with the same color. Since this method needs a powerful edge detection to split different image color regions, Yi-Chin Huang et al. [13] proposed an extension method of scribble-based colorization. To prevent blurry colors in the edge locations, they implemented a powerful edge detection algorithm and then applied the colorization.

Example-Based Colorization

Example-based colorization is also used widely in different studies. The main idea of this approach is, using a reference image, algorithm colorizes pixels with same color which has similar characteristics with the reference image. A simple work is proposed by Tomihisa Welsh et al [5] in 2002. They simply use a similar image to the grayscale image and matches the color with the grayscale image (e.g human skin color from reference image to human skin in target image). Guillaume Charpiat et al. [10] uses a probability distribution of possible colors instead of using the most probable color as earlier works. They do not consider only a single reference image, but use multiple reference images to colorize the target image. Some of the more recent works such as [11], [12], [4] are also used example-based colorization. Given a reference string with the target grayscale image, a method is developed to search the internet for similar reference images to colorize target image. Another method uses superpixel level representation and compares superpixels of the reference image and target image to apply colorization and so on.

Learning-Based Colorization

As we mentioned, our method is also related to the learning-based colorization. Learning-based colorization uses a large dataset to train the learning model (usually CNN), and learns the color info of the given input images from the train dataset. Some of the studies on learning-based colorization [8], [1], [14], [9] gave nice colorization results in the recent years. Most of these approaches uses a deep Convolutional Neural Network (CNN) architecture to extract features and then solve the classification/regression problem. While examining these studies, we have encountered architectures such as VGG, ResNet like deep architectures. Some of the studies implement their own probability distribution layer instead of a fully-connected layer and so on. Most recent work we have encountered and can be considered as state-of-the-art method is work of Mingming He et al. [7] where they both use example-based colorization and learning-based colorization together.

3. The Approach

Our first base line was Colorful Image Colorization [1], but according to some issues on framework we had to switch our interest on another research which is Automatic Image Colorization [2]. We were planning to switch backbone of Colorful Image Colorization's model with ResNet style network. They have developed and implemented VGG-like model to achieve colorization problem. Each convolution block is followed by activation function, ReLU (Rectified Linear Unit) and batch normalization. But their CNN(convolution neural network) does not contain any pooling layer and fully connected layer [1].

But the Automatic Image Colorization research [2] showed us that the ResNet style network is already implemented. In Automatic Image Colorization, they used ResNet-18 and MRF(Markov Random Field) to implement colorization operation.

They used ResNet-18 for their model's backbone. Colorization problem uses gray-scale inputs for forward propagation, they say this backbone increases feature learning skills of their model and provides some simplicities for training. So after reading the paper of Automatic Image Colorization[2] we decide to create our model using DenseNet. Our model contains two different parts like Automatic Image Colorization. We create first part of our model using DenseNet121 [3].

And for second part we used deconvolution layers to get output as a regression.

Recent years shows us convolutional neural networks can solve many visual task perfectly. Although twenty years have passed since their discovery they became popular recently according to the some improvements on computer hardwares and big data.

Day by day researches came up with new models and they simply propose deeper model followed by smart changes to achieve more complex problems, but getting deeper convolution neural network creates another problem: vanishing. Since gradient moves through many many layers it loses it's effects when it reaches the beginning of the convolution neural network.

In traditional convolution neural networks each layer follows by another layer after applying operations such as filtering, pooling, batch normalization, activation function etc. In Dense Convolutional Network(DenseNet) it concatenates features map with same height and width.

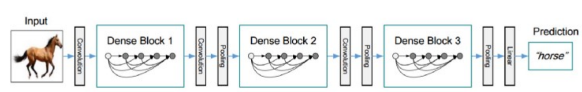


Figure 2. A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature map sizes via convolution and pooling.

Figure 2: DenseNet

Concatenating each feature map with matching sizes, keeps maximum amount of information which is transferring between layers. Also this dense connectivity pattern needs less number of parameters than traditional neural networks since there is no need to relearn redundant feature-maps.

We directly copied DenseNet121's first 52 layers

and added into our model. We used pretrained model's weigths(on ImageNet) and bottom layers of DenseNet because using pretrained model provides an advantage: We do not have to reinvent the wheel. Since bottom of deep models encodes low level information such as edges, lines, textures by copying these layers and their weights' we are reducing the needed train time.
In second part we applied deconvolution operation (convolution blocks and upsampling to get increased resolution for features). Output shape is (224, 224, 2) for each image and it contains a and b color channels corresponding images.

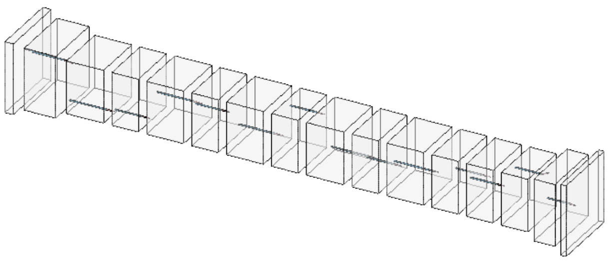


Figure 3: Visualization of Our Model Architecture

We look colorization problem as a regression problem. Because it is simply predicting color channel values by using gray-scale version of same image.

4. Experimental Set-Up

4.1. Loss Function

We approach to the problem as a regression problem. There are output color values to predict by using gray-scale version of the image.
We mapped lightness values to ab values. We choose $CIE\ lab$ color space to perform training. Since CIE lab is a perceptually uniform space, this means that, in any color pairs difference in CIE lab color is proportional with Euclidean Distance. Also using $CIE\ lab$ instead of RGB reduces number of parameters to predict (our model maps gray to a and b channels instead of r, g and b channels).

We decided use mean square error for this regression problem just like Automatic Image Colorization [2]. But in the nature of colorization problem some plausible result can be considered as a lower losses.

4.2. Dataset

Currently, there are 14,197,122 images, 21841 synsets indexed on ImageNet. Since training the model on complete ImageNet dataset will be painful in terms of time, we have decided to train/test our model with a small subset of cat and dog images from ImageNet dataset. We have created a text file that contains URLs of some cat and dog images. We have coded a script which iterates through each image URL, and downloads the image to generate our dataset.

Figure 4: Ground Truth Image - Colorized Output with High Loss Value

In domain of colorization problem there are some previous researches [1], [2] related with this topic which used ImageNet and MIT Places365 datasets mostly for training phase. ImageNet has 14,197,122 images and Places365 has nearly 10 million images. When we compare these datasets with ours, we have obviously limited dataset. To provide our model more training samples we used some data augmentation techniques such as random noise, random rotation, horizontal and vertical flip to fill the dataset with more input images.
We had approximately 6k images before we applied data augmentation. These images contain both different kind of cats and dogs images. After we have applied random noise, random rotation, horizontal and vertical flip to our images we got different 24k images to use as an input.

4.3. Train

We trained our model on Google Colaboratory with our dataset. We iterated model around forty epochs. Due to lack of computational power, we could not play with the hyper-parameters that much.

When we observed the graph in Figure 5, we thought that

Figure 5: Example Images from the Dataset

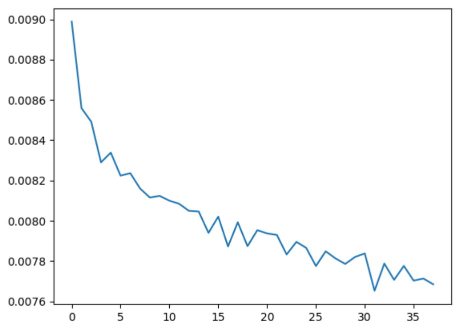


Figure 6: Our model's loss graph in every 100 batches

our model started to learn but since loss did not converge, we can not say training was enough. Actually there are nearly twenty epochs, we trained our model again after we get this graph but due to the some issues on Google Colab's notebooks we could not retrieve final losses.

5. Experimental Results

5.1. Evaluation Metrics

5.1.1 L2 Distance

To measure our models' power in a accuracy manner, we compare predicted color pixels' ab color channels with the ground-truths. After gather the results we measured between distances results and ground truths by L2 distance.

$$d = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}$$

5.1.2 Pixel by Pixel

Similarly to L2 distance we thought we can define similarity in a similar way. For each pixel if the distance between them is less than the threshold value we consider them as correct predicted pixels, else we consider them as wrongly predicted pixels. For correctly predicted pixels we are not modifying total differences between two images. But if these points encodes different ab values we are increasing total difference between two image by one.

(a) Predicted Result with L2 Distance = 221.59

(b) Original Image

Figure 7: Best result in our evaluation set according to L2 dist.

(a) Predicted Result with L2 Distance = 370.62

(b) Original Image

Figure 8: Second best result in our evaluation set according to L2 dist.

(a) Predicted Result with L2 Distance = 790.31

(b) Original Image

Figure 9: Third best result in our evaluation set according to L2 dist.

(a) Predicted Result with Pixel by Pixel Distance = 2

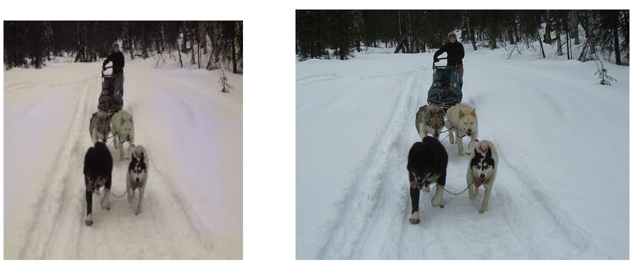
(b) Original Image

Figure 10: Best result in our evaluation set

(a) Predicted Result with Pixel by Pixel Distance = 4

(b) Original Image

Figure 11: Second best result in our evaluation set



(a) Predicted Result with Pixel by Pixel Distance = 23

(b) Original Image

Figure 12: Third best result in our evaluation set

5.1.3 Semantic Interpretability (VGG Classification)

As previous work[1] we thought that was good idea to feed a deep classifier with fake colorized images. We thought VGG16 will be fine for this task [1]. Since VGG16 is trained with real colored images, if classifier works fine with our fake colorized images that means our predicted colorized images are not that bad else they might be colorized not good enough. We used top one accuracy to calculate accuracies.

Accuracy with Ground-Truths	Accuracy with Gray Scales	Accuracy with Colorized Images
%61	%49	%35

When we investigate the results, we can clearly say that our colorization is not good as much as we aimed. But main reasons of the bad quality are the lack of training dataset and lack of computational power.

6. Conclusion and Future Works

We worked on very limited dataset that comes with its' consequences like, if we train with large iterations on our dataset, our model may overfit to our dataset or if we train with small iteration count and use small learning rates, this time our model may suffer from underfitting. Even if we ran train with forty epochs on our dataset we can observe some sign of overfitting. When we investigae background in our evaluation results, our model thinks that ground material is greenish most of the time. And sometimes we can observe that our model overfit for green color.

(a) Ground Truth Image

(b) Colorized Image

Figure 13

To overcome these problems we did not use too deep model. We tried to use small amount of the learn-able parameters since we did not have enough data to learn correct weights. We can clearly see that, this is not a quiet solution. As future works we can list some useful tasks for us:

Data set expanding : we can not prevent overfitting problem by using simpler model but we have strong faith in our model, it can perform quite good colorization if it is trained on larger dataset.

Using deeper model : Since we did not have quiet large dataset, we tried to use as simpler model as possible. In future we should try deeper model with larger train dataset.

Converting regression problem to a classification : In previous works[1][2] they tried to solve this problem from a different angle. They treat this problem as a classification problem. They divide color space CIE ab into different number of pins and tried to estimate every pixels corresponding color channel class. In our solution, we tried to estimate every pixel's exact color channel values. Converting this problem from regression to classification might increase the plausibility and accuracy.

Image colorization is one of the core problems in the computer vision. It contains predicting color channel values of gray pixels. By this project we have shown that, Densely Connected Convolutional Networks can be used as a solution of this problem. We got not that brilliant results but we have strong believe in our model. We think our model can get a proper solutions on larger dataset with enough training. Even though it gives some results which motivates us for future works.

References

[1] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. ECCV, 2016

[2] L. Melas-Kyriazi, G. Han. Combining deep convolution neural networks with Markov Random Fields for image colorization

[3] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In CVPR, 2017.

[4] He, M., Liao, J., Yuan, L. and Sander, P.V. *Neural color transfer between images.* arXiv preprint arXiv:1710.00756., 2017.

[5] Welsh, T., Ashikhmin, M. and Mueller, K. *Transferring color to greyscale images.* ACM Transactions on Graphics (TOG), 2002.

[6] Levin, A., Lischinski, D. and Weiss, Y. *Colorization using optimization.* ACM Transactions on Graphics (TOG), 2004.

[7] He, M., Chen, D., Liao, J., Sander, P.V. and Yuan, L. *Deep exemplar-based colorization.* ACM Transactions on Graphics (TOG), 2018.

[8] Larsson, G., Maire, M., Shakhnarovich, G. *Learning representations for automatic colorization.* European Conference on Computer Vision, 2016.

[9] Cheng, Z., Yang, Q., Sheng, B. *Deep colorization.* Proceedings of the IEEE International Conference on Computer Vision, 2015.

[10] Charpiat, G., Hofmann, M., Scholkopf, B. *Automatic image colorization via multimodal predictions.* Computer VisionECCV, 2008.

[11] Chia, A.Y.S., Zhuo, S., Gupta, R.K., Tai, Y.W., Cho, S.Y., Tan, P., Lin, S. *Semantic colorization with internet images.* ACM Transactions on Graphics (TOG), 2011.

[12] Gupta, R.K., Chia, A.Y.S., Rajan, D., Ng, E.S., Zhiyong, H. *Image colorization using similar images.* Proceedings of the 20th ACM international conference on Multimedia, ACM, 2012.

[13] Huang, Y.C., Tung, Y.S., Chen, J.C., Wang, S.W., Wu, J.L. *An adaptive edge detection based colorization algorithm and its applications.* ACM International Conference on Multimedia, 2005.

[14] Iizuka, S., Simo-Serra, E., Ishikawa, H. *Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification.* ACM Transactions on Graphics (Proc. of SIGGRAPH 2016), 2016.