ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)

Available Online at www.ijetcr.org

Volume 8; Issue 2; March-April 2020; Page No. 01-05

IMAGE COLORIZATION

Alok Dixit

IT Department (AKTU), Student of B.Tech (4thyear), Pranveer Singh Institute of Technology, Kanpur, India

E-mail: alok.dixit.8953141221@gmail.com

Conflicts of interest: Nil

Corresponding author: Alok Dixit

Abstract

We are going to develop an automatic image colorization project. Here we work on the deep neural network. In this project we train our model and by training we predict the pixels color. This intermediate output help to colorize the grayscale image to color image. There are various methods to colorize the image but, in this project, we are using Tensor Flow to change the grayscale image into color image. This uses self-supervised learning feature and train thousands of the images and finally predict the color of the image. The predicted color is totally depending on the training of images. We leverage an existing large-scale scene database to train our model to learn the global priors and classify the objects of image to be able to map color to it. We demonstrate our method extensively on many different types of images, including black-and-white photography from over a hundred years ago, and show realistic colorizations.

INTRODUCTION

In this project we introduce the problem of grey image colorization which consider one of the most problems face the user because it requires a hard effort to produce a natural output close to the ground truth and a high computation from machine. There was many tries to apply colorization process over images but all tries didn't be as the user expect as it requires a deep interaction from the user to help in colorizing process which still a problem to solve. To change the Grayscale images into color image is a very simple task. Predict the color of image by human being is very simple like color of moon is white, tree's leaf is green, and sky is blue. This colorization gives multiple applications (e.g. images and videos [1], artist assistance [2,3]. We recently join some recent efforts on automatic colorization [4,5,6]

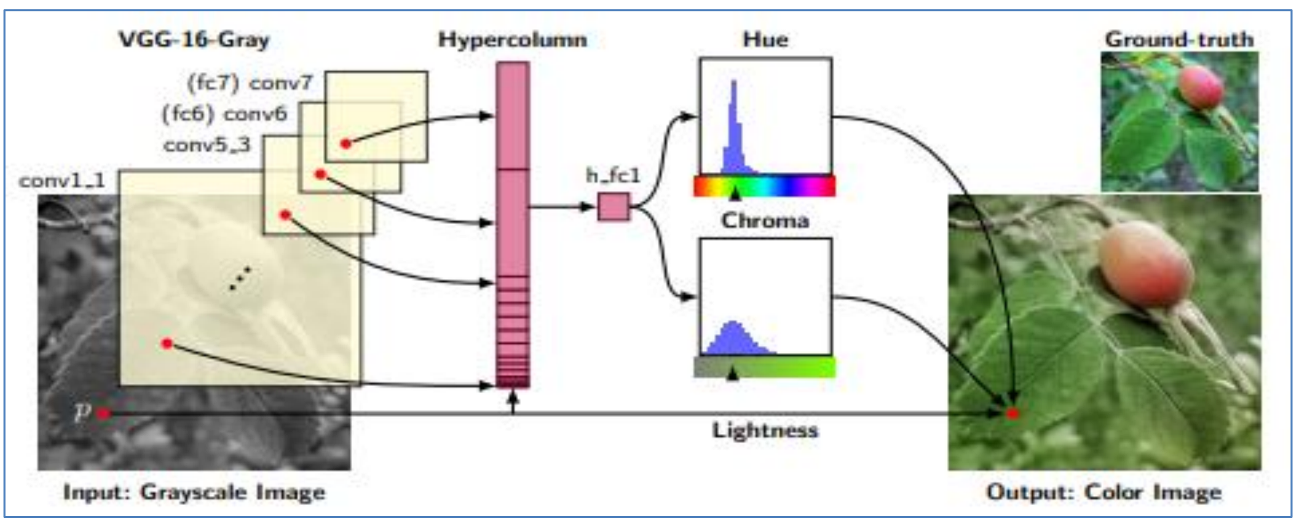

Figure 1:

The above image [7] is showing that we make the convolutional layer up to 7 and train it. After that it generates Hyper Column and it gives two things i.e. Hue and lightness. Output color image cannot be same as original (truth) image due to change in Hue and lightness.

Literature Review

A. Affined Work

Our work is totally based on automatically colorizing images. Our systems rely on many image like Net-like Dahl's, there are trained layers that are completely connected with each other that are already exist in the system. We first introduce the statistics of our dataset. We then show three main applications: realistic image manipulation, generative image transformation, and generating a photo from scratch using our brush tools. Finally, we evaluate our image reconstruction methods and perform a human perception study to understand the realism of generated results. Please refer to the supplementary material for more results and comparisons. In previous work we have to inject various methods to ensure that many colorization induced by a channel are globally complete. One method to conserve a relative field that can be slow. There is one method for the conserve a CNN with multiple output. It can evaluate a different colorization of an image. For the selection the best image it can be additionally trained the network. The whole process and mixture is only conserving for the image compression to some extent colorization. And another approach is to capture dependencies amongst outputs via a low dimensional latent space for a conserve a conditional variation auto encoder. It will use VAE's decoder for the changement in color image. Unfortunately, this results and method was produces by sepia toned. For the capture of dependence on the input image, proposes for the conserve of a mixture density network to learn a mapping from a gray input image to a distribution over the latent codes. Hence, a successful automatic color predictor should be general enough to incorporate various sources of information in a global manner. In order to cope with the multi-modality, we discretize the color space and investigate multi-class machine learning methods.

System Analysis and Design

We are processing the Grayscale image through deep neural network and take process it by 8 layers. You can take 9 or 12 layers [8] so that the output is more precise, but this will increase the complexity during the test image.

When we provide a Grayscale image first it references to trained images that we have taken and then transferring into color image. This is done only if references images are found and it is difficult for unique images. To colorize the image the system must be interpret the image i.e. leaf, moon, sun, cars, faces etc.

It is not mandatory that it colorize the image same as the original image like the color of clothes. It depends on the image style and the references images. We are creating our project that instead of single color of clothes it predicts a color histogram. The whole framework, including the global and local priors as well as the colorization model, is trained in an end-to-end fashion.

Active Function

In this we create objective function, we have given an input image *X* and our work is to learn a mapping *Y=f(X)* where $Y \in I^{H*W}$ here H and W are the dimensions of the image.

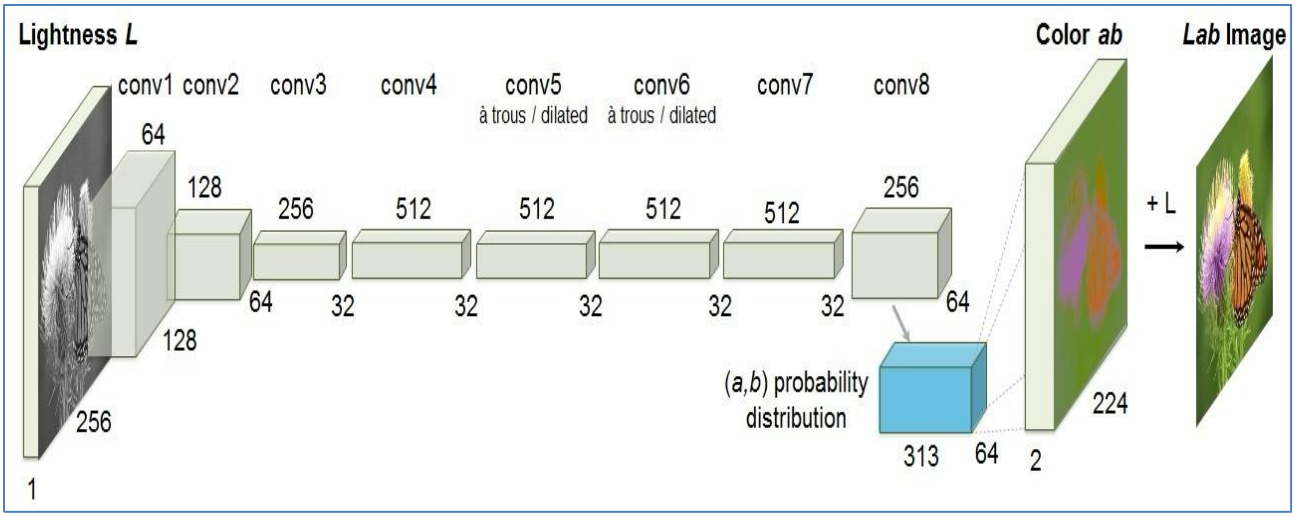

Figure 2:

The above image [4] showed that we have a simple Grayscale image and we trained the image using 8 convolutional neural networks and then we have color ab probability distribution and change into Lab image.

Remember, we cannot predict the same color as original image has. Some images must be predefined color for example- The color of sky is blue, the color of tree is green, the color of apple is red (it is possible the color of apple may be green, yellow or red but we take only red color).

In colorization problem we frame as learning a function *f: X → Y*. We have a Grayscale image that is x belongs to *X. f* predicts the color of the image and the general size of the grayscale image is S*S. In our project it reads images of pixel which have a dimension 264*264 and it have 3 channels red, green, and blue in the RGB color space. Black and White image is an input for our model and it produces U and V channels as an output. The images are converted into CIELUV color space. We have now three channels one which is an input for the model and other two channels are produced by the model. Then combine all the three channels to form CIELUV representation of the predicted image.

In this project we used ReLU (Rectified Linear Unit) activation function. Since it is most widely used activation function in neural network. We are using ReLU in our project because we want to decrease the non-linearity in our image.

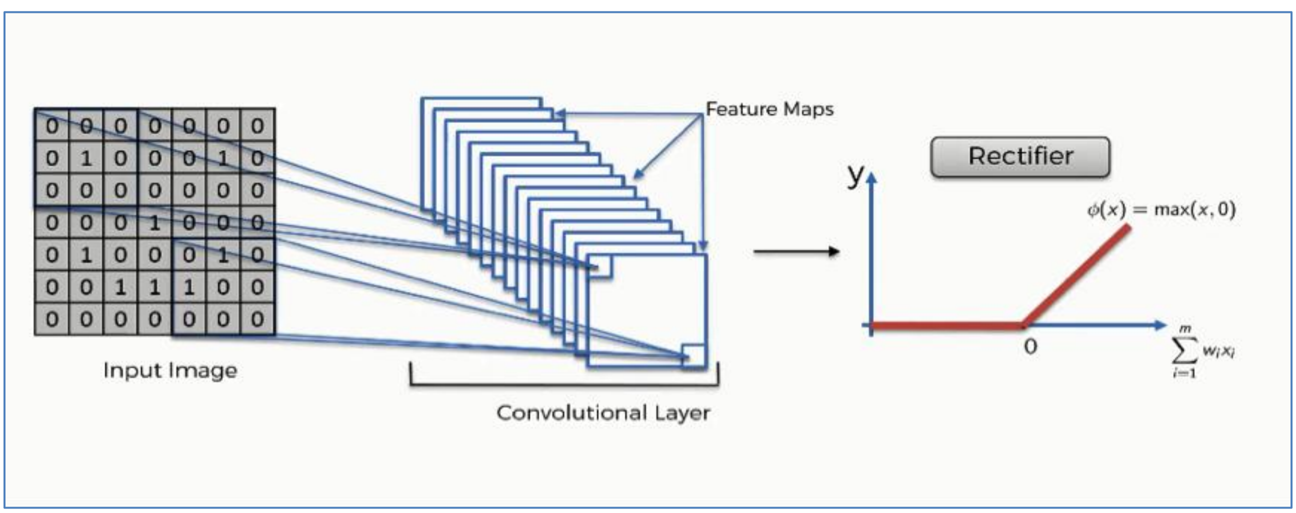

Figure 3:

Mathematically it can be represented as: -

$$f(x)=max\ (0, x)$$

This function accelerates the training of data. It is simpler than many other convolutional activation functions.

We will create training data by converting color images to grayscale according to L=(R+G+B)/3.

Evaluation Matrices

The closeness of the generated image to the actual image as the sum of the L2 norms of the difference of the generated image pixels and actual image pixels in the U and V channels [9]**:**

$$L_{reg.} = \left\|U_g - U_a\right\|_2^2 + \left\|V_g - V_a\right\|_2^2$$

For classification we measure the closeness of the generated image and the actual image by percent of binned pixels value that compare with the generated image and the actual image [10].

$$Acc._{v.} = \frac{1}{N^2} \sum_{(i,j)}^{(N,N)} 1\{bin(U_g) = bin(U_a)\}$$

Finally, we find the percent deviation in color saturation between the generated image pixels and in the actual image:

$$\text{Satur. diff.} = \frac{\sum_{(i,j)}^{(N,N)} s_{g_{i,j}} - \sum_{(i,j)}^{(N,N)} s_{a_{i,j}}}{\sum_{(i,j)}^{(N,N)} s_{a_{i,j}}}$$

In this project we are using “Colorization Turing Test” (CTT) to test the output image, whether the output image is different from original image if “yes” then “how much it is different”. In this test we take real participants as human and asked them to find the original and output image. It is difficult for them to find the output image.

In addition to serving as a perceptual metric, this analysis demonstrates a practical conserve for our algorithm without any additional training or fine-tuning; we can improve performance on gray scale image classification, simply by colorizing images with our algorithm and passing them to an off-the-shelf classifier.

Result and Review

Figure.(4) There are two objects of quail and assortment along with the black-and white input images that model are cultivate that was trained on the dataset.

Our model can process images of any size, it is most efficient when the input images are 128x128 pixels, as the shared low-level features layers can share one output between mid and global level features. When the changement of the input image size resolution, while the low-level feature weights are shared then the image size of rescaled is 128 × 128 must be conserved for the global features network that requires rescale image through model processing in both the original image. So, in Test Model function where we put our testing model networks we conserved 2 low level networks and both conserves the shared low-level feature weights but one of them takes the original grey scale image and feed the mid-level network with its output layers while the other network conserves a rescaled image of size 128x128 and feed the global level network.

References

1. Tsaftaris, S.A., Casadio, Andral, J.L., Katsaggelos, A.K.: A novel visualization tool for art history and conservation: Automated colorization of black and white archival photographs of works of art. Studies in Conservation 59(3) (2014)
2. Qu, Y., Wong, T.T., Heng, P.A.: Manga colorization. ACM Transactions on Graphics (TOG) 25(3) (2006).
3. S`ykora, D., Buri´anek, J., Z´ara, J.: Unsupervised colorization of black-and-white ˇ cartoons. In: International symposium on Non-photorealistic animation and rendering (2004).
4. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: ICCV (2015).
5. Deshpande, A., Rock, J., Forsyth, D.: Learning large-scale automatic image colorization. In: ICCV (2015).
6. Aditya Deshpande, Jason Rock and David Forsyth. Learning Large-Scale Automatic Image Colorization**.** In *ICCV*, Dec 2015.
7. Gustav Larsson, Michael Maire , and Gregory Shakhnarovich March 22 2016.
8. Automatic Image Colorization with Simultaneous Classification**.** In *SIGGRAPH*, 2016.
9. 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.
10. Image Colorization with Deep Convolutional Neural Networks jeff Hwang and You Zhou.