Journal of Image Processing and Artificial Intelligence
e-ISSN: 2581-3803
Volume 04, Issue 02
4th National Conference on Advancements in Information Technology

An Approach to Convert Grayscale Images to Color

[1]Urvashi Kar, [2]Prajval B, [3]Shaikh Arif Mohd Mondal, [4]Chandan D S, [5]Dr. T H Sreenivas
[5]Professor
Department of Computer Science and Engineering
Vidyavardhaka College of Engineering, Mysuru
Email: [1]karurvashi@gmail.com, [2]prajbhas@gmail.com, [3]moondal.arif@gmail.com, [4]iamchandandongre@gmail.com, [5]drthsreenivas@vvce.ac.in

Abstract

This paper investigates the coloring problems that occur while transforming a grayscale image into a color image. Earlier works on colorization techniques involve approaches to choose colors from a palette of RGB and transferring them on to the gray image. In other methods it either requires a lot of color scribbling on the black and white image or a huge dataset of color reference for the image in particular to the era it belongs to. We use convolutional neural networks along with a feature extractor and the Inception-ResNet-v2 pre-trained classifier model for higher efficiency in coloring. Our neural network is combined with the classifier that increases the performance of similar images. We train our neural network on images from Unsplash, an image collection website, that are available as a public dataset.

***Index Terms**: Colorization, Inception Reset, Keras, Neural Networks, Tensor Flow.*

INTRODUCTION

Colors in an image, make the visual appearance of it more appealing. Every object that is visible to the human eye is perceived by its shape, size, texture and the colors on it. A colorless image, that is a grayscale image would be less comprehensible for information gathering and less appealing. This reason calls for the colorization of grayscale images. Historical images, old documentaries, scientific illustrations and surveillance feeds could be colored and made more beneficial and insightful.

Colorization of an image is the process of coloring every pixel in a gray image. Colorization is an expensive process and takes lot of time. Traditional coloring methods include artists using photo editing tools, researching about the history of the image to gain knowledge of the styles of colors existing and used in the time of the image. This process could easily take up to a month or sometimes more. But with recent developments in machine learning and computing capacities, the methods of colorization are changing.

A gray image has only one dimension, its luminance, while a colored image has three dimensions which pertain to the RGB color scale. Since a single gray pixel with its luminance can be related to different colors, there is no right solution to this problem. At such stage, human intervention plays an important role in making decisions about the result being correct. The proposed methodology intends to make this process more automatic with a lower need of human interactions.

In this paper we are using deep neural networks to build a model that takes a grayscale image as an input and colorizes it. We train our network using a dataset of images from Unsplash. We are using CNN with the activation functions Tanh and ReLU (non-linear) to train the model. We use Inception-ResNet-v2 [1], which is a pre-trained network on colored images. The Inception model is used to extract features in an image before it is colorized.

RELATED WORKS

Di Blasi et al., proposed a method of transferring color to a grayscale image from a base image based on the luminance information of the black & white pixels [2]. This is a semi-automatic method that minimizes the amount of human effort required in colorization; it uses a large collection of 'color words' to achieve the final image. Ref. [3] is another method that transfers color to grayscale images.

Hwang et al., proposed two approaches to image colorization, one using a regression learning model and another VGG16 model to classify the image after regression to recolor images based on the subject matter [4]. They use a 224 x 224 size image with RGB channels to train the model and convert the final output of the grayscale image to a CIELUV image. They found that adding a classification model was a better choice in coloring the images compared to just using the regression model.

Cheng et al., investigated a fully automatic method of coloring grayscale images using a patch matching technique with a low-level patch feature, mid-level DAISY features, and a high-level semantic feature with Chrominance refinement to achieve fast high-quality color conversion [5].

Organization

The organization of the paper is in multiple sections. Related works section, emphasizes on the other works that have been done in colorization of images. Core logic section is a brief idea of how our proposed idea works in the most basic sense. Methodology section includes the in-detail methodology of our algorithm and is split into two subsections. Results section presents the results obtained by us in the form of images obtained on executing the model. The last section concludes our experiments and discusses the future scope of colorization methodology.

CORE LOGIC

Every image can be represented as an array containing rows and columns, and each individual cell in this array is called a pixel, so an image is essentially made out of pixels. A black and white image consists of pixels and each pixel has values that span from 0-255.

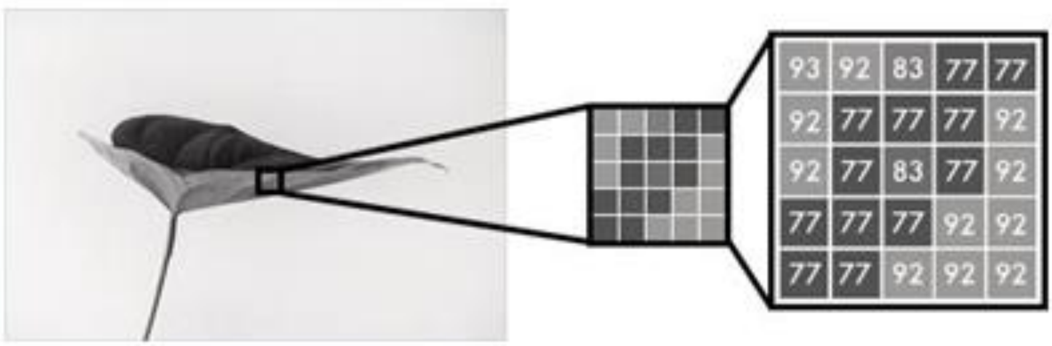

***Fig 1:** An image of a leaf in grayscale with a tiny section enlarged to depict the pixels with their corresponding value in the range of 0-255*

A colored image in RGB scale is made of three separate layers. The red layer, green layer and the blue layer. These layers define the color intensity as well as its brightness.

Journal of Image Processing and Artificial Intelligence
e-ISSN: 2581-3803
Volume 04, Issue 02
4th National Conference on Advancements in Information Technology

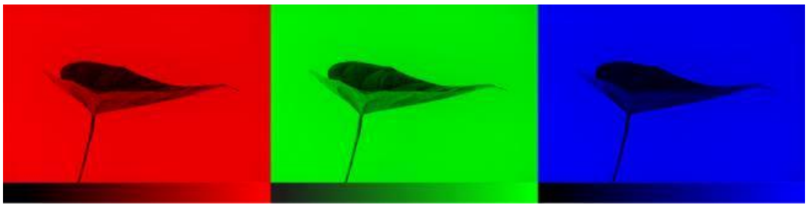

Fig 2: *Three layers of colors in the leaf image in rgb*

Each layers on combining forms an image that seems realistic to the human eye. The three layers of colors can also be called the three channels. These colored layers also have a value in the range of 0-255 for each pixel. The combination of the values of these three layers gives us a resultant color. The value 0, implying no color in the layer. When all the three channels in a pixel have the value 0, the pixel gives a black color. And if the three channels give a value of 255, the pixel shows a white color.

The core logic of our network is to find a relationship between the pixel values of the grayscale image to the corresponding pixel values of the colored image, in the three layers. Such relationships are best defined using neural networks.

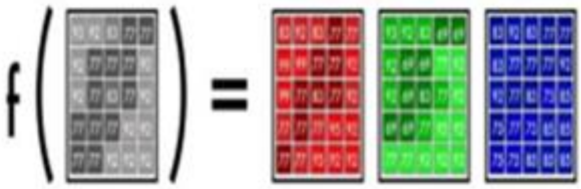

Fig 3: *Grayscale Input, Gives An Rgb Output After Being Passed Through F(), The Neural Network*

METHODOLOGY

We tried two versions of approaches to colorize the grayscale images, and they are explained in the following sections.

Initial version

We train our initial version of network on a single input image, which is colored and then test it on the same image. The colored input image, is converted to grayscale internally and then tested to give us the output image.

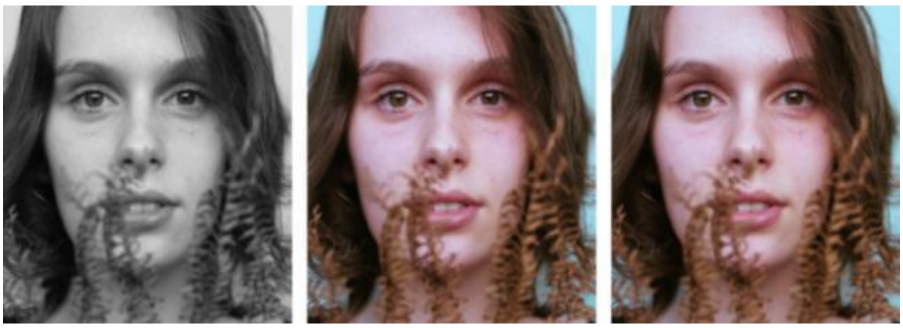

Fig. 4: *The image on the right is our colored input image and the middle image is the output of the first version neural network. The left image is an internal conversion of the input image to grayscale.*

MAT JOURNALS
Journal of Image Processing and Artificial Intelligence
e-ISSN: 2581-3803
Volume 04, Issue 02
4th National Conference on Advancements in Information Technology

Before we begin training the network, we convert the RGB color space to Lab color space. L stands for the amount of light or brightness of the image, a stands for the colors green and red, and b stands for blue and yellow that gives us one grayscale image layer and two colored layers.

Fig 5: *The three images are the three layers in the Lab scale.*

Our prediction involves taking the grayscale as input and predicting the two colored layers ab in Lab. For conversion of one layer to two, we use Convolutional Neural Network (CNN), which act like filters. Hundreds of these filters are stacked together to generate the a and b layer.

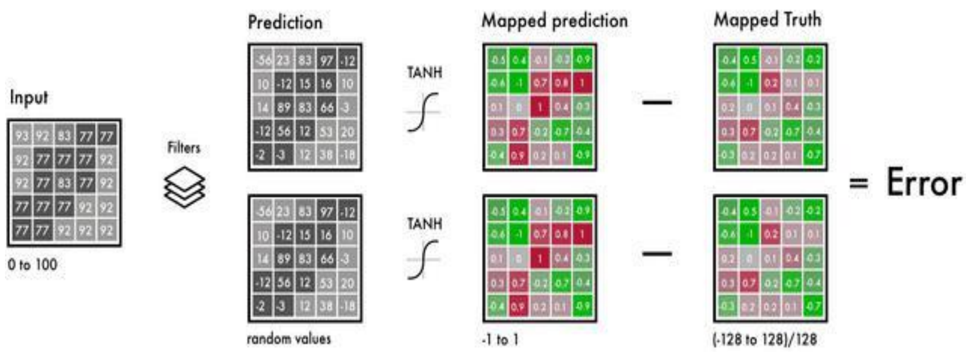

Fig 6: *From left, the input, the filters and the mapped prediction compared to original to produce error.*

A Tanh activation function is used to have our interval ranged between -1 and 1. After generating the Lab color space image, we convert it to RGB and produce our final output image.

The drawback of this version is that we can't test the trained network with any new unknown image, it would only give us bad results. Thus, leading to the next version.

Final version

In this version, we train and test our network with a dataset of 10,000 images taken from Unsplash. This public dataset is available on Floydhub. The images are processed through a feature extractor, where the image is filtered repeatedly. Each filter generates a new image. After some steps, we have filtered images that look something like.

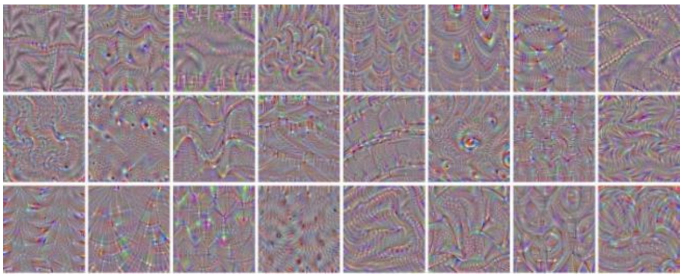

Fig 7: *Intermediate images of the filtering process*

Using just the feature extractor and training the images, the result images show a majority of brown texture and color. But this is overcome with the use of the pre-trained model of Inception-ResNet-v2 in TensorFlow.

To include the classifier, we introduce an encoder in parallel to it. After this, the image is extracted from the classification layer and merged the output from the decoder.

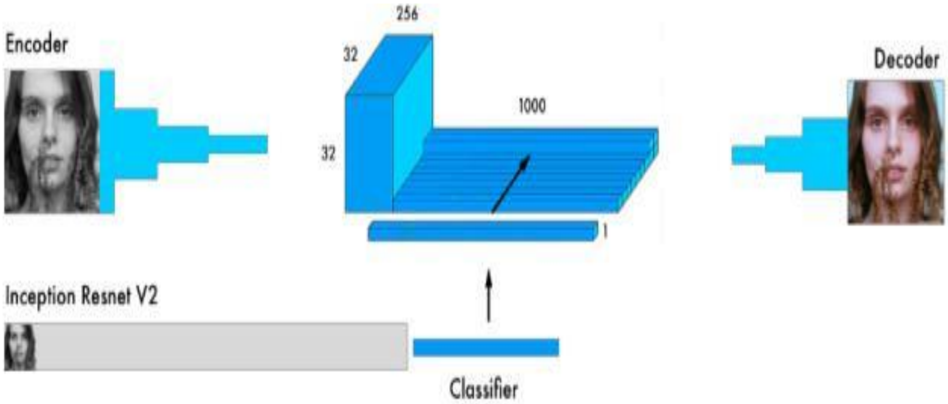

Fig 8: *The encoder in parallel to classifier, inception resnet-v2, and the decoder.*

RESULTS

The output images shown below are a result of having trained the network on 20 images alone (due to time and performance constraints). The images are better than the initial version, yet they need to improve. The error correction is done using Mean Square error functions. Further training on all the 9,500 images from the dataset will definitely improve the results.

Fig 9: *Result images of the trained model*

CONCLUSION AND FUTURE SCOPE

This paper emphasizes on using CNN with a classifier Inception-ResNet-v2. In particular, our approach is able to successfully color images such as the desert, the sea, forests or anything that has multiple shades of the same color or multiple colors. However, the performance in coloring smaller details is still to be improved by using a larger dataset to train.

The future prospect would be to apply our colorization techniques to video sequences which could be used to re-master old documentaries, movies and other footage.

REFERENCES

1. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. “Rethinking the inception architecture for computer vision” *arXiv preprint arXiv:1512.00567*, 2015.
2. D. Blasi, R. Recupero, “Fast colorization of gray images,” Eurographics Italian Chapter, 2003.
3. T. Welsh, M. Ashikhmin, and K. Mueller. “Transferring Color to Greyscale Images,” *in Proceedings of ACM SIGGRAPH*, San Antonio, 2002, pp 227-280.
4. J. Hwang and Y. Zhou, “Image colorization with deep convolutional neural networks,” Stanford University, Tech. Rep.,2016.
5. Z. Cheng, Q. Yang, B. Sheng, “Deep colorization,” in *Proceedings of the International Conference on Computer Vision*, 2015.