# Recolorisation of Images using Deep Neural Networks

**Madhavi Mane[1], Shivam[2], Khushboo Kumari[3], Tanmay Sharma[4]**

[1]*Assistant Professor, Dept. of Computer Engineering, Bharati Vidyapeeth (DU) College of Engineering Pune, Maharashtra, India*

[2,3,4]*Student, Dept. of Computer Engineering, Bharati Vidyapeeth (DU) College of Engineering Pune, Maharashtra, India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract** - *Recoloring greyscale images of historic events and old movies has always been a topic of interest and a challenge requiring intense manual labor alongside historic and artistic expertise for accurate reproduction. We propose a refined approach using deep convolutional and adversarial neural net frameworks for achieving the task of recolourization and substantially reducing the manual work force in bringing out aesthetically pleasing re-colorized images from a vast data set of natural as well as digitally crafted images. We dive through convolutional neural networks as the underlying foundation for understanding how the machine manipulates images and reduce error in output prediction. We also aim at producing exceptional outputs with reduced rounds and generate higher resolution colorized images.*

***Key Words***: **Recolouring, Neural network, Convolutional, Grayscale image, Deep learning,**

## 1. INTRODUCTION

Picture colorization is a genuinely wide theme. In this undertaking, we characterize picture colorization to be the way toward moving greyscale pictures into colored pictures. In present day PC illustrations, a colored picture is regarded as a 3-channel information cluster. For the most part, greyscale picture can be considered to be as a 1-channel information exhibit. Along these lines, inside the extent of this task, the goal of picture colorization is to become familiar with a mapping function y = f(x), which maps 1-channel information exhibit x into a 3-channel information cluster y, and produces conceivable and outwardly lovely outcomes. We attempt to mechanize the procedure of colorization by utilizing learning in convolutional neural net and another model of adversarial systems to enhance the consequence of the colorized pictures obtained.

### 1.1 RGB Color Space

RGB color space is a very common representation of colourful images in computer graphics. Specifically, a colourful image is represented by a 3-channel array. Each channel represents the chromaticity of Red, Green and Blue respectively. Chromaticity means the quality of the color regardless of its luminance. Red, green and blue together form the basis for additive as well as subtractive primary colors. In computer graphics, every data point in every channel can take the value from 0 to 255, normally as integers. If this value is closer to 0, the color of the corresponding channel looks darker. Several examples include Black : [R; G;B] =[0; 0; 0], White : [R; G;B] = [255; 255; 255], and Red : [R; G;B] = [255; 255; 255].

### 1.2 Lab Color Space

Lab color space is one of the other widely used color space. In Lab color space, a colorful image is also represented as 3-channel data array. Comparing with RGB, the difference is the meaning of each channel. The first channel L in Lab color space represents lightness, which is a black/white color intensity. When L is set to 0, it corresponds to black and at L set to 100, it corresponds to white. Thus L takes value from [0; 100]. The second channel a 2 [-128; 127] is the red/green channel, with green at negative a values and red at positive a values. The third channel b 2 [-128; 127] is the yellow/blue channel, with blue at negative values and yellow at positive values. When a = 0 or b = 0, both channels represent neutral grey 1 color. Lab and RGB color spaces are inter-changeable using some non-linear transformation, which is commonly available in image processing packages.

### 1.3 Convolutional Neural Network

Convolutional neural systems are profound artificial neural systems that are utilized principally to extract features from pictures. Convolutional Neural Networks are fundamentally the same as any ordinary neural network. These are formed out of neurons that have weights that can be changed or modified and biases. Each neuron obtains a little of data sources, performs a dot product operation and then finalizes it with a non-linearity. A differentiable score function is then expressed by the system: from the crude picture pixels toward one side to class scores on the other. Then comes a loss function like softmax acting on the final layer. ConvNet structures make the express presumption that the data sources are pictures, which enables us to encode certain properties into the engineering. These then make the forward capacity progressively effective to execute and tremendously decrease the measure of parameters in the system.

Neural Networks gets information in the form of a vector and transform it by undergoing a series of hidden layers. Every hidden layer is comprised of a lot of neurons, where every neuron is completely associated with all of the neurons in the previous layer, and where neurons of a one aloof layer work totally autonomously and don't share any connections. The last completely associated layer is known as the "output layer" and in arrangement settings it gives out the class scores. CNNs have two components:

- The Hidden layers/Feature extraction: In this part, the system will play out a progression of convolutions and pooling activities amid which the highlights are identified. On the off chance that we had a picture of a creature like zebra, this is where the system would perceive its stripes, two ears, and four legs.
- The Classification: Here, the completely associated layers will fill in as a classifier over these extricated highlights. The likelihood for the prediction being correct will be assigned to each article in the picture.

## 2. METHODOLOGY USED

Highly contrasting images (black and white) images can be represented in matrices of pixels. Every pixel has a value that compares to its brightness. The values of the matrix range from 0-255, from dark (black) to white.

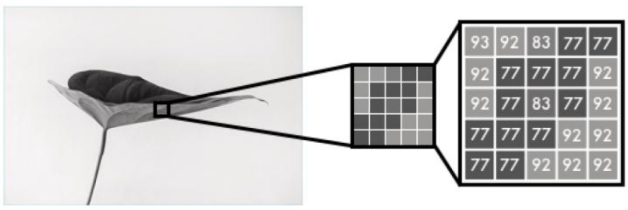


*Figure 2.1: Splitting of an image into a matrix*

Color images consist of three layers: a red layer, a green layer, and a blue layer.
To achieve the color white, for example, you need an equal distribution of all colors. By adding an equal amount of red and blue, it makes the green brighter. Thus, a color image encodes the color and the contrast using three layers:


*Figure 2.2: Combination of RGB for different shades of green*

Each layer in a color image has a value from 0 - 255. 0 meaning that it has no color in this layer (void or black). 255 means that it has all the colors in this layer which essentially means white.

### 2.1 Architecture of CNN used

The basic function of a neural network is to map a function between an input value and output value. Our aim here is to achieve re-colorization, hence it is imperative for the network to be designed to try and find the features present in the images that can link gray scale images with colored ones so as to achieve our goal.

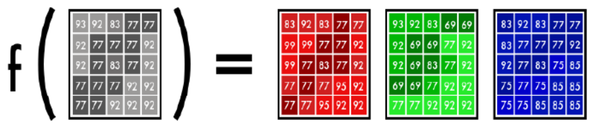

*Figure 2.3 Linking grayscale images with RGB format*

First and foremost, we need to convert RGB channel to Lab channel. **L** stands for lightness, a and b channel is given by the color combinations green–red and blue– yellow. Because of this we can use the original grayscale image for our final prediction. Lab channel also reduces the number of channels required to be predicted to two.

Like 3D glasses, convolutional filters turn one layer into two layers. The network in a similar way can either create a new image from a filter or combine several filters into one single image.

For a convolutional neural network, each of the filter gets automatically adjusted to help with the required outcome. We begin by stacking hundred and hundreds of filters and finally narrow them down into two layers which are the **a** and **b** layers of Lab.

The black and white image is represented by the input matrix. Two grids with color values comes in the output. We make filters between the input and output values to interface them together, a convolutional neural network.
The black and white layer of image is our input and the two colored layers of Lab channel are the output when we train the network using colored images.

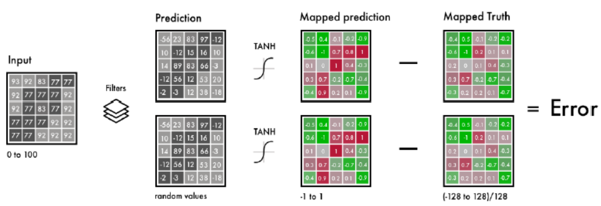


*Figure 2.4 Determine output image and compare it with ground truth image*

To the left side, we have the B&W image input, or filters and the predictions made by our neural network. We then map the predicted values and the real values within a unified range or interval. This way, we can serve the purpose of comparing the values with ease. To obtain

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

these predicted values we use a Tanh activation function. For any value you fed to the Tanh function, it will return -1 to 1.

The default value interval in the Lab color space goes from -128 to 128. So having them divided with 128, puts them to fall within the -1 to 1 interval, thus allowing us to contrast the error from the prediction made.

After calculating the final error, the filters are updated by the network to reduce total error. The network performs this iteration until the error is as low as possible. A final prediction is made which then gets converted into a picture once training of neural network is completed. Image in grayscale accepted as input and passed into trained neural network. All the output values are multiplied with 128 which were earlier in the range of -1 to +1 so as to give us the right color present in the Lab color spectrum.

Last but not the least, we create a complete black RGB color canvas by initialising it with three layers (three layers because R, G, B) of 0s. Then we emulate the grayscale layer from our test image followed by adding our previous two color layers to the RGB canvas. This array of pixel values we recently obtained is then converted into a fully colorized image.

### 2.1.1 Feature Extraction

First, we look for simple patterns as a diagonal edge line, various occurrences of black pixels, and so on. We look for the same exact pattern in each square and remove as many of the pixels that don‘t match. We generate 64 new images from our 64 mini filters.

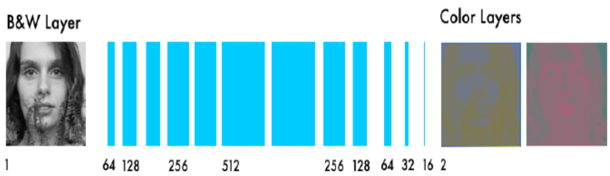


*Figure 2.5: Feature Extraction*

We scan the images again, to find that the same small patterns that we have already detected. To achieve a higher level of understanding in our image, we reduce the image size in half.

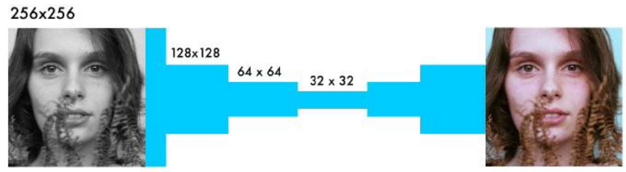


*Figure 2.6: Recolorising image using downsampling and upsampling*

We still only possess a three by three filter to scan each of the image. But by combining our new nine pixels with our lower level filters we can detect much more complex patterns than before. One pixel combination might tend to form a half circle, a small dot, or a diagonal line. We repeatedly extract the same pattern from the image and we now generate 128 new filtered images.

### 2.1.2 Color Extraction

The neural network operates in a trial and error methodology. Random prediction made for each pixel. Using error calculated, feature extraction is improved.

It starts adjusting all the situations where the error generated is the largest errors. In this case, it‘s whether to color or not and to locate different objects.

Since most of the training data is under consideration is similar, the network struggles to differentiate between different objects. It will adjust different tones of brown, but fail to pop out more nuanced colors.

The max-pooling layers also result in substantial distortion of the image apart from increase the information density. We can use a stride of 2 in coloring networks, to reduce the both width and height by half thereby augmenting information density and it does not mutilate the image.

Two further tasks are the up-sampling layers and maintaining the image ratio. Classification networks are primarily focused upon the final classification.

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

## 3. EXPERIMENTAL SETUP

Implementation of all our networks was done on Keras deep learning framework for Convolutional Neural Networks part. The experiments were run on Tesla K80 GPU with 12.72 GB RAM. The GPU we used were accessed through Google Colaboratory.

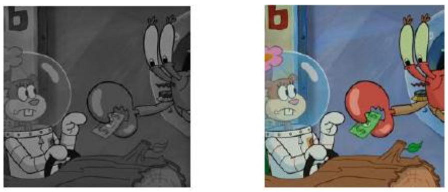

*Figure 3.1: Input image on left followed by ground truth image on right*

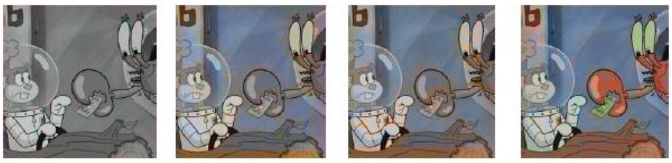

*Figure 3.2: Output in increasing number of epochs(Left to right)*

## 4. CONCLUSIONS

In this project, we explored methods for image colorization. The simple ConvNet is fairly easy to implement, but it produces averaged color in the output. CNN model is evaluated on published image dataset. There are several future research directions on image colorization. First is to increase the variety of colors. We verified that Euclidean distance will decreases the variety of colors. Majority of published datasets contains labels. For example, the OxFlower dataset contains labels indication the category of flower. Utilizing these models can be helpful for colorization. Intuitively, if the model can learn what the flower is before colorization, the results are expected to be better. Overall, the variety of color is improved, but the results become highly plausible for higher epochs.

## REFERENCES

[1] Cheng, Zezhou, Qingxiong Yang, and Bin Sheng. "Deep colorization." Proceedings of the IEEE International Conference on Computer Vision, 2015.

[2] Levin, Anat, Dani Lischinski, and Yair Weiss. "Colorization using optimization". ACM Transactions on Graphics (ToG). Vol. 23. No. 3. ACM, 2004.

[3] Huang, Yi-Chin, et al. "An adaptive edge detection based colorization algorithm and its applications." Proceedings of the 13th annual ACM international conference on Multimedia. ACM, 2005.

[4] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).

[5] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., -ImageNet: A large-scale hierarchical image database.|| In CVPR, 2009.

[6] K. Simonyan A. Zisserman "Very deep convolutional networks for large-scale image recognition" CoRR vol. abs/1409. 1556 2014.

[7] X. Wang X. Tang "Face photo-sketch synthesis and recognition" IEEE Transactions on Pattern Analysis and Machine Intelligence vol. 31 pp. 1955- 1967 Nov 2009.

[8] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 1447–1454, 2006.

[9] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. CoRR, abs/1603.08511, 2016.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to- image translation with conditional adversarial networks, 2016.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on computer Vision and Pattern Recognition, 2015.

[12] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks science, 2016.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2015.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, 2015.

[15] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In Proc. ICML, volume 30, 2013.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2015.

[17] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of machine learning research, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, 2015.

[19] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sen Gupta, and Anil A Bharath. Generative adversarial networks: An overview 2017.

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[21] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In NIPS'2012.
[22] Rifai, S., Bengio, Y., Dauphin, Y., and Vincent, P. (2012). A generative process for sampling contractive auto-encoders. In ICML'12.
[23] Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. arXiv preprint arXiv:1602.02644, 2016.
[24] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. arXiv preprint arXiv:1606.03498, 2016.
[25] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. ECCV, 2016.
[26] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. CVPR, 2016.