Image Colorization Using Convolution Neural Networks

Zahra Sayed
Computer Department
M.H.Saboo Siddik College of Engineering
Mumbai, India
zahrasayed27@gmail.com

Zeba R. A. Shaikh
Computer Department
M.H.Saboo Siddik College of Engineering
Mumbai, India
zebaahmed671@gmail.com

Mohammed Raza Sayed
Computer Department
M.H.Saboo Siddik College of Engineering
Mumbai, India
razasayed120201@gmail.com

Barirah Khan
Computer Department
M.H.Saboo Siddik College of Engineering
Mumbai, India
khanbarira84@gmail.com

Nazneen Pendhari
Computer Department
M.H.Saboo Siddik College of Engineering
Mumbai, India
mailnazneen@gmail.com

Abstract **— Image recolorization enhances the perception of a picture for style and creative functions. Colorization is now a computer-assisted method of adding color to a grayscale image/movie. Historically, this method needed user interaction, in the form of putting various color scribbles, gazing connected pictures, and performing arts segmentation. However, with advancements in technology, automatic colorization systems are created. In this work , we are developing a system that colorizes grayscale pictures automatically. Convolution neural networks is the tactic used here.**

Keywords **— Colorization, Convolution Neural Network, RGB color space, LAB color space, autoencoders.**

I. INTRODUCTION

Colorization of gray scale images is an appealing application of image processing, the visual appeal of black and white images is increased, classic movies or scientific demonstrations. In medicine, image modalities that solely acquire grayscale pictures such as Magnetic Resonance Imaging (MRI), X-ray and Computerized Tomography (CT) images can be enhanced with color for displays and demonstrations. In addition, the knowledge content of some scientific images are often perceptually enhanced with color by exploiting variations in color property further as luminosity [1]. For studying the past state of the environment, grayscale images are of the main available sources and its relationship to the present. However, these images lack spectral data thereby obstructing their use in current remote sensing approaches that think about spectral information for characterizing surfaces. Although grayscale pictures contain valuable

historical data, they continue to be poorly exploited because of their heterogeneous specifications and quality, e.g. scale, lens properties, spectral sensitivity, and film development . However, these products can still serve diverse purposes, including cartography, landscape dynamics analysis, and photogrammetry. The contribution of color, texture, and lightness features for land monitoring using remote sensing is well known, for example, Heller et al. highlight the importance of color for distinguishing tree species from aerial pictures by showing that identification accuracy improved by up to twenty seventh once foresters used pictures obtained with RGB films compared to panchromatic stills. Photography and picture industries, facing similar problems, were the primary to develop colorization techniques, which permit superimposing color to grayscale stills [2].

II. LITERATURE SURVEY

The various techniques existing for adding chrominance values to a Grayscale image are discussed in this section. Even though there are many techniques, there is no optimum solution for this problem. Some of the existing colorization techniques are luminance keying, scribble based colorization and colorization by sparse representation. The problem with these ways is that they demand considerable effort from the user.

A. Photo Colorization with Global and Local Consistency

The approach is as follows:

Step1: Drawing color scribbles. Locating on the approximate position of a region and the color used to represent the region. Coloring tools can choose any kind of computer drawing software.
Step2: Color space conversion. Given some colors scribbled by hand, in order to separate intensity from color, we work in the YUV color space, which can be converted from RGB color space, where Y is intensity, while the colors are U and V.
Step3: Graph model construction. The pixels in segmented regions are connected into a graph by their spatial relationship.
Step4: Optimize the modified energy function.
Step5: Integrating the three components. Integrating the solved U and V components along with the initial input Y components and converting into RGB color space to form the final output colored image [3].

B. Colorization Using Chrominance Blending

A key component of this algorithm is the computation of the intrinsic, gradient weighted, distance provided. Followed by this, the distance is computed. The algorithms used are inspired by the classical Dijkstra graph algorithm, fast sweeping techniques, and special data structures such as buckets and priority queues. A straightforward, swift and exact implementation of the algorithm is observed. It was found that there was no real need to implement the computation of the weighted distance using the aforementioned accurate techniques, and simply using classical Dijkstra turned out to be finely appropriate. Thereby, in theory, a graph was created where every image picture element may be a vertex

connected to its neighbors with edges. The proposed algorithm relaxation obtained by limiting the number of contributors to the blending equation gave a tight restriction on how far the chrominance will propagate in order to be included in the mixture. The restriction had been easily implemented by adding conditions in the pseudocode [4].

C. Colorization with given color Palette

PaletteNet was made to take two inputs: a source image to be re-colored and a target palette. It was then designed to vary the color conception of a supply image so that the palette of the output image is close to the target palette.PaletteNet includes two subnets: feature encoder (FE) and recoloring decoder (RD). The FE present in PaletteNet is a completely convolutional neural network. FE results in reducing the spatial size of every feature map in two equal parts by residual blocks. The hierarchical content feature is used by RD from FE. Also, the spatial information of the image is encoded using this. The training PaletteNet involved two phases: 1. Pretrain FE and RD with E-loss. 2. The parameters of FE and train RD are freezed along with extra Adv-loss. This splits training stabled learning colorization process with Adv-loss [5].

III. METHODOLOGY

A. RGB Color Space

RGB stands for red–green–blue. An RGB color space is any additive color space primarily based on the RGB color model. The three chromaticities of the red, green, and blue additive primaries are defined by a particular color space that employs RGB primaries for part of its specification and can produce any chromaticity that is the primary colors which define the 2D triangle (ie. excluding transfer function, white point, etc.). An RGB color is often understood by thinking of it as all potential colours that may be made of three coloured lights: red, green, and blue. For instance, imagine having three shining lights together onto a white wall in a closed dark room: one red light, one green light, and one blue light, each with dimmers. The wall will be red if only the red light is on and will look green if the green light is on. If both red and green lights are on at same time, the wall will appear yellow. The wall will become more sort of a yellow-green chromatic light when we dim the red light. When we dim the green light instead, the wall will become more orange. Referring the blue light a small amount will cause the orange to become less saturated and more whitish [11].

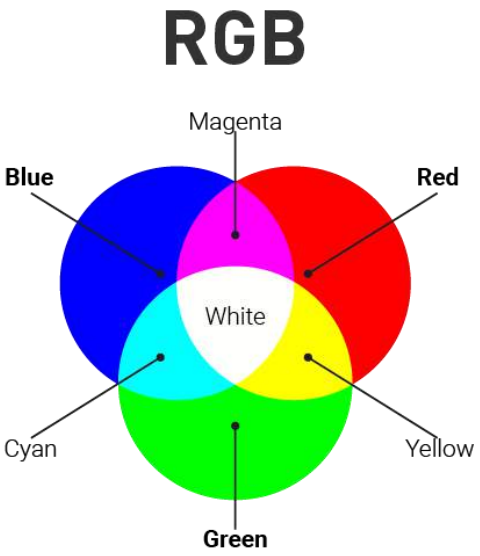

Fig 1. RGB Color area [15].

B. Lab Color Space

It expresses color as three major values: *L** for perceptual lightness, and *a** and *b** for the four distinctive colors of human vision: red, green, blue, and yellow. CIELAB was supposed to be a perceptually uniform lab area, where a given numerical change corresponds to similar perceived change in color. While the LAB space is not truly perceptually uniform, it nevertheless is helpful in industry for detecting minor differences in color [12].

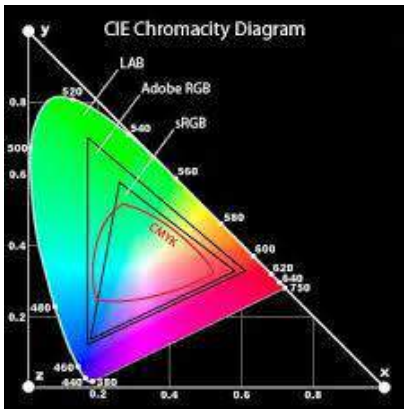

Fig. 2. *LAB color area [12].*

C. Autoencoders

Autoencoders are a type of Neural Networks' architecture that try to reduce the dimensionality of the input data They are built from two components:

Encoder — transforms the input data into a lower-dimensional portrayal (also known as the latent vector/representation). To attain this goal, the encoder must learn only the foremost vital features of the information.

Decoder — the input is recovered from the low-dimensional illustrations.
The following image presents an autoencoder network.

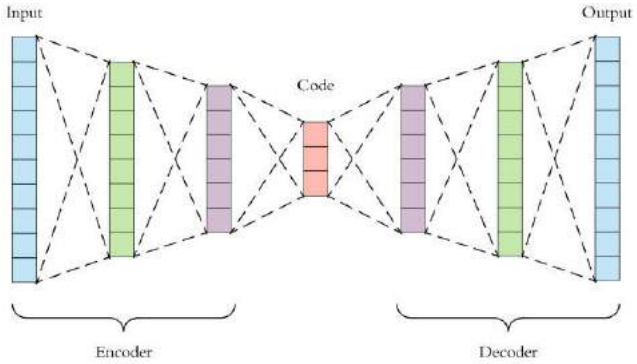

Fig. 3. *Autoencoders*

D. Convolution Neural Network

In deep learning, a convolutional neural network (CNN, or ConvNet) is a category of deep neural networks, most commonly applied to analyzing visual representational processes. They are also called as shift invariant or space invariant artificial neural networks (SIANN), based on the shared-weight design of the convolution kernels that scan the hidden layers and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, Image segmentation, medical image analysis, language process, brain-computer interfaces, and money statistics [13]. A Convolutional Neural Network is a neural network capable of handling image data. It consists of 3 layers, 1 convolution layer, 1 pooling layer, and 1 fully connected layer.

Pooling Layer: Convolutional networks might embody the local or global pooling layers to contour the underlying

computation. Pooling layers prune the size of the information by combining the outputs of neuron clusters at one layer into a single neuron within the next layer. Local pooling combines tiny clusters, typically 2 x 2. Global pooling acts on all the neurons present in the convolutional layer. There are two common sorts of pooling: max and average. *Max pooling* uses the utmost value of each cluster of neurons at the previous layer, whereas *average pooling* instead makes use of the average value.

Fully Connected Layer: Fully connected layers connect every neuron in one layer to every alternative neuron in another layer. It is similar to the standard multi-layer perceptron neural network (MLP). The two-dimensional matrix goes through a completely connected layer to classify the images.

IV. PROPOSED SYSTEM

Recent years have witnessed a rise in machine controlled colorization techniques that are deployed using numerous implementations and deep learning backends. The general technique is introduced here for "colorizing" grayscale images by transferring color between a supply/source, color image and a destination or target, grayscale image.The approach given here tries to supply the simplest way to help minimize the number of human labor required for this task.

A. Building Data Set

We have to choose a dataset that is consistent, for example, many of the datasets found have many black and white images and that can maximize your error in the training part because during this case rather than showing a grayscale image and its colored version for your model to learn the mapping from grayscale to colored image, you are showing a grayscale version and a grayscale image too as a target, nothing helpful for your model to learn during this case, therefore you must take away any grayscale pictures from your dataset and choose all of them to be coloured as RGB. In our project we will be using ImageNet dataset.

B. Data Preparation

Now we will convert all the images from RGB color space to LAB color space. Another issue to try before changing to *Lab* is to normalize your dataset(divide by the maximum value that *RGB* pixel can reach) because in RGB each image is represented as an array with dimensions [3, 288, 320], where numbers on the scale of 0–255 describe the intensity of a given channel.The idea behind using the *Lab* color space is to reduce the complexity of the problem, so the output of the autoencoder is an array of the size [2, 288, 320].This "normalization" enables us to match the error from our prediction and converge faster. Also, all the images ought to be of an equivalent size, so we will resize them to be "*256* x *256*"

By iterating on every image, we tend to convert the RGB to Lab. As our input to the network can be the L channel, we tend to place the L channel of each image in vector X and also the ab in vector Y, we need to divide Y by 128 as a result of the range of values of the ab channel is between (-127, 128). Once normalization is fully carried out, the

values obtained would be between (-1, 1). We add this line X=X.reshape(X.shape+(1,)) because we wish to have equivalent dimensions for X and Y.

C. Train Your Network

Encoder part: It includes some convolutional layers with activation function ReLU and Strides=2 inorder to decrease the width and height of the latent space vector.
Decoder part: It includes convolutional layers with upsampling layers to restore the dimensions of the original input image (256x256) and reconstruct the image with 2 filters at the last layer which represents the *ab* channels. You may notice here that in the last layer, we used tanh activation function to operate rather than ReLU. You must be able to recognize why we created this as a result of we tend to normalize the ab values to be between (-1,1) and tanh used for squashing the values between (-1,1).

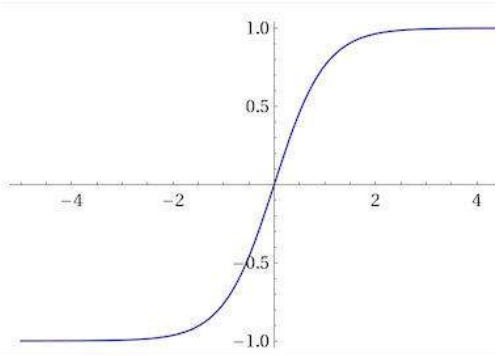

Fig. 4. *Activation function of tanh [14].*

After constructing the model,we train it using mean square error because the loss function and Adam as the optimizer. The amount of epochs may be a “Trial and error” selection.

D. Post Processing

As we tend to knock off the preprocessing for training, we do the same before testing. It necessarily doesn’t hold any importance whether the testing image is in grayscale or colored as in each way we extract the *L* channel and predict with the grayscale image. After assuming, we expect the values to be in (-1,1) range. We need to multiply the values by 128 inorder to restore the values of *ab* channels because of the tanh function used at the last layer. The last thing is to concatenate the L channel that you just used for testing along with your output ab channels to reconstruct the lab area, then save the image you just constructed after converting it to *RGB*.

V. CONCLUSION

Finally, we conclude that we can colorize grayscale images using this approach. This system can be used in the future of coloring of old black and white movies and images, as well as color-restoration of old images.We found We discerned that the dataset is essential and may modify your results, as an example, if you wish your model to be specific and colour pictures of flowers, you've got to settle on a dataset of flowers, the additional knowledge you give the higher the result. Once we train the CNN, it can generate best results automatically within a reasonable time. Unlike the other existing methods, this scheme can obtain good results within comparatively less time. The ReLU activation function and normalization method in the neural network helps to achieve better convergence.

REFERENCES

[1] Kumar, A. Swarnkar "Colorization of Gray Scale Images in lαß Color Space Using Mean and Standard Deviation",2012 IEEE Students' Conference on Electrical, Electronics and Computer Science
[2] Q. Poterek , P. Herrault , G. Skupinski, and D. Sheeren "Deep Learning for Automatic Colorization of Legacy
Grayscale Aerial Photographs" ,IEEE App Earth Observations and Remote Sensing, vol 13, 2020
[3] Gaigai Zong,Ying Chen,Guangcheng Cao,Jiawei Dong, Ying Chen "Image Colorization With Local and Global Consistency",2015 8th International Congress on Image and Signal Processing (CISP 2015)
[4] L. Yatziv and G.SapiroFast, "Image and Video Colorization Using Chrominance Blending"IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 15, NO. 5, MAY 2006
[5] Junho Cho, Sangdoo Yun, Kyoungmu Lee, Jin Young Choi, "PaletteNet: Image Recolorization with Given Color Palette",2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops
[6] Brian Sam Thomas, Rajat Dogra, Bhaskar Dixit, Aditi Raut "Automatic Image and Video Colourisation using Deep Learning"
[7] Swathy Titus,Jency Rena N.M "Fast Colorization of Grayscale Images by Convolutional Neural Network"
[8] Jeff Hwang, You Zhou "Image Colorization with Deep Convolutional Neural Networks"
[9] Satoshi Iizuka Edgar Simo-Serra Hiroshi Ishikawa, "Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification"
[10]https://towardsdatascience.com/image-colorization-using-convolutional-autoencoders-fdabc1cb1dbe
[11]https://en.wikipedia.org/wiki/RGB_color_space
[12]https://en.wikipedia.org/wiki/CIELAB_color_space
[13]https://en.wikipedia.org/wiki/Convolutional_neural_network
[14]https://becominghuman.ai/auto-colorization-of-black-and-white-images-using-machine-learning-auto-encoders-technique-a213b47f7339
[15]https://www.pinterest.com/pin/247909154473935541/