An Efficient Automatic Digital Image Colorization Method Using Convolutional Neural Network

Gaurav Subedi  
Department of Computer and Electronics Engineering  
Kantipur Engineering College  
Dhapakhel, Lalitpur  
gasu.nine@gmail.com

Subodh Raj Satyal  
Technology Director  
Upveda Technology  
Anamnagar, Kathmandu  
subodhsatyal@gmail.com

Abstract—Colors are the elements of human visual perception that portray the vivid liveliness of the creation. This perception of color is so diverse that there is no perfect color combination for a given artifact. Colorization is a generic term used to describe a computerized process for adding color to black and white pictures, or motion graphics. Digital Image Colorization is the colorization process applied to still image artifacts. Although people have controversial views about the artistic value of colorization, it is no doubt that colorization of monochrome artifacts enhances the visual effects.

Colorization is basically a mapping between the intensity values and the chrominance values with no 'correct' but plausible solution. Inspired by the current trends in deep learning, we propose a colorization framework that utilizes both the local and global features for colorization. Global features include the color rarity of each color class in the quantized ab plane calculated over the whole dataset. We have also implemented a custom loss function suitable for this purpose. We tackle the colorization by treating it as a multi-class classification by determining the probabilities of colors in the color gamut and combining the probabilities to a specific color. We have successfully implemented the method to various types of images ranging from legacy b&w images to modern images (having colored versions of their grayscale counterparts).

Index Terms—colorization, convolutional neural network, annealed mean, probability distribution, class rebalancing, grayscale

I. INTRODUCTION

Colorization is the art of adding color to a monochrome artifact. Digital Image Colorization is such process applied to digital images. This process has long been recognized as a laborious and tedious task. Despite recent advances in this field for the automation of the task, a considerable amount of manual effort is still required in many cases to achieve a plausible colorization result.

The technique of colorization is not new. Ironically, hand coloring of photographs is as old as photography itself. It was practised in the motion pictures in the early 1900s by French Company Pathé, where many films were colored by hand. It was widely practised for filmstrips in the 1930s. Computer assisted process was first introduced by Wilson Markle in 1970 for adding color to black and white movies [1].

There are mainly two techniques for image colorization [2].

1) Scribble Based Colorization

In this method, the user draws color scribbles on the monochrome image and the system performs colorization based on the user input colors. It works on the fact that regions with similar intensities have the same color. This method requires extensive user effort with near-accurate color scribbles, so it is time-consuming.

2) Example Based Colorization

Example-based methods use a reference image to learn the color characteristics of the input image. Authors in [3] segment the example image and determine for each pixel which example segment it should learn its color from. This is done automatically using a robust supervised classification scheme that analyzes the low-level feature space defined by small neighborhoods of pixels in the example image. Example-based colorizations can be further divided into two categories based on the source of the reference image:

a) Colorization using user-supplied example(s):

This method requires the user to provide a suitable reference image. Welsh et. al. [4] employ the pixel intensity and neighbourhood statistics to find a similar pixel in the reference image and then transfer the color of the matched pixel to the target pixel.

b) Colorization using web-supplied example(s):

To release user’s burden of finding a suitable image, Liu et al. [5] and Chia et. al. [6] utilize the massive image data on the Internet. Liu et al. compute an intrinsic image using a set of similar reference images collected from the Internet. This method is robust to illumination difference between the target and reference images, but it requires the images to contain
identical object(s)/scene(s) for precise per-pixel registration between the reference images and the target grayscale image. The goal is not necessarily to recover the actual ground truth color, but rather to produce a plausible colorization that could potentially fool a human observer. We can achieve this by modeling the statistical dependencies between the semantics and textures and combining them with the global image features to achieve visually compelling results.

We treat the colorization problem as multinomial classification problem. We quantize the ab output space into bins with grid size 10 and keep $Q = 313$, where $Q$ is the total number of quantized ab values, which are in-gamut, as shown in the figure 1.

![Fig. 1: Quantization of ab color space](image)

A deep neural network is a universal approximator that can represent arbitrarily complex continuous functions. [11] Given a set of exemplars $\Lambda = \{G, C\}$, where $G$ are grayscale images and $C$ are corresponding color images respectively, the method used in this project is based on the premise: there exists a complex gray-to-color mapping function $F$ that can map the features extracted at each pixel in $G$ to the corresponding chrominance values in $C$. Our goal is to create such a mapping function $F$ so that we can use $F$ to convert a new gray image to color image. In our model, we employ the Lab color space, because distances in this color space model the perceptual distance.

### A. Lab Color Space

It is one of the most popular color spaces for measuring object colors. The Lab Color Space defines color using three values: $L$ for lightness ranging from 0 (black) to 100 (white), $a$ for red-green difference and $b$ for blue-yellow difference. Basically, $ab$ components give the color value and $L$ gives the amount of the colors. The Lab color space includes all perceivable colors and its gamut exceeds that of RGB and CMYK color spaces. Since the $a$ and $b$ channels are computed as differences of lightness transformations of cone responses, it is also known as chromatic value color space. The ab plane defines the color plane. It is the output of the model, which is concatenated with the L plane of the original image to get the colorized output. This space is a three-dimensional real-number space containing infinite number of possible representation of colors.

![Fig. 2: Three-Dimensional Lab Color Space](image)

This color space was chosen for the research as it includes all the colors visible to the human eye and is device-independent.

### B. Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>ReLU</td>
<td>Rectified Linear Unit</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green and Blue</td>
</tr>
<tr>
<td>CONV</td>
<td>Convolution</td>
</tr>
<tr>
<td>L</td>
<td>Lightness</td>
</tr>
<tr>
<td>CMYK</td>
<td>Cyan, Magenta, Yellow and Black</td>
</tr>
</tbody>
</table>

### II. Literature Review

Many prominent research works have been done in the field of digital image colorization. Research papers regarding the very topic have proliferated in the past decade.

Levin et. al. [7] have proposed a colorization technique based on the premise that neighboring pixels in space-time that have similar intensities should have similar colors. They used quadratic cost function for this purpose. In their approach, an artist/user only needs to annotate the image with a few color scribbles, and the indicated colors are automatically propagated in both space and time to produce a fully colorized image or sequence.

The method in [8] uses automatic colorization technique producing vibrant and realistic colorizations. They have categorized the problem as classification task and used class-rebalancing at training time to increase the diversity of colors in the result. They
implemented the system as feed-forward pass in a CNN at test time and trained it on over a million color images.

Izuka et al. in [9] propose a colorization algorithm for automatic coloring of grayscale images by combining both global priors and local image features along with classification. Their deep network features a fusion layer that allows to elegantly merge local information dependent on small image patches with global priors computed using the entire image by classification. The merging is done in a fusion layer. The architecture can process images of any resolution, unlike most existing approaches based on CNN.

Cheng et al. in [2] have proposed a joint bilateral filtering based post-processing step to achieve artifact-free quality. They further develop an adaptive image clustering technique to incorporate the global color information.

Deshpande et al. in [10] pose automatic colorization technique as regression in continuous ab color plane and use quadratic objective function in the chromacity maps. The objective function admits correlations on long spatial scales, and can control spatial error in the colorization of the image. Images are then colorized by minimizing this objective function.

Li and Ng in [13] use second order variant of graph Laplacian for colorization. They formulate the colorization as an optimization problem by solving the resulting linear system.

We propose an efficient colorization algorithm that is enhanced for colorizing historical, cultural as well as other natural artifacts and has the ability to capture vibrancy as well as spatial coherency of the image all the while making the colorized image visually pleasing.

III. PERFORMANCE PARAMETERS

Colorization is a multimodal problem, so there may be more than one plausible color versions of the same image. So, there are no concrete parameters for testing the performance of the colorization algorithm. However, since colors are human visualization elements, we use Turing Test to test the effectiveness of our approach.

To test our method, we evaluated the feedback from 100 participants. The participants included students and staffs from various faculties of the college. Each participant was given a total of 10 colorized images and asked if it was real or fake. Each participant was given an unlimited time to decide whether it was colorized version or ground truth.

IV. METHODOLOGY

A. CNN Architecture

We utilize a single stream CNN architecture, whereas in paper [9], two network streams were used for colorization. Table I summarizes the architecture of the CNN used in the research:

<table>
<thead>
<tr>
<th>Block</th>
<th>O/PC</th>
<th>BN</th>
<th>Kernel</th>
<th>O/PR</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>224x224</td>
<td></td>
</tr>
<tr>
<td>conv1</td>
<td>64</td>
<td>Y</td>
<td>3</td>
<td>112x112</td>
<td></td>
</tr>
<tr>
<td>conv2</td>
<td>128</td>
<td>Y</td>
<td>3</td>
<td>56x56</td>
<td></td>
</tr>
<tr>
<td>conv3</td>
<td>256</td>
<td>Y</td>
<td>3</td>
<td>28x28</td>
<td></td>
</tr>
<tr>
<td>conv4</td>
<td>512</td>
<td>Y</td>
<td>3</td>
<td>28x28</td>
<td></td>
</tr>
<tr>
<td>conv5</td>
<td>512</td>
<td>Y</td>
<td>3</td>
<td>28x28</td>
<td>2-dilated CONV</td>
</tr>
<tr>
<td>conv6</td>
<td>512</td>
<td>Y</td>
<td>3</td>
<td>28x28</td>
<td>2-dilated CONV</td>
</tr>
<tr>
<td>conv7</td>
<td>256</td>
<td>Y</td>
<td>3</td>
<td>28x28</td>
<td>1-dilated CONV</td>
</tr>
<tr>
<td>conv8</td>
<td>128</td>
<td>-</td>
<td>4</td>
<td>56x56</td>
<td>1-dilated CONV</td>
</tr>
</tbody>
</table>

In the table, Block represents the data and convolutional blocks in the network. Each block consists of two to three convolutional layers, followed by ReLU activation. There is no presence of Pooling Layers. So, the resolution change is obtained by spatial upsampling or downsampling between the convolutional blocks. O/PC indicates the number of channels in output; BN specifies whether Batch Normalization was used after the convolution blocks; Kernel - the size of the kernel used; O/PR - the spatial resolution of the output from each convolutional block.

B. Extracting Color Value from Probability Distribution

When the L channel input (X) passes through the convolutional network, it is transformed to \( \hat{Z} \) by the neural network. Mathematically, this transformation can be represented by:

\[
\hat{Z} = T(X)
\]

\( \hat{Z} \) has the dimensions of \( W \times H \times Q \) where \( W = 56 \), \( H = 56 \). For each \( W \times H \) pixels, \( \hat{Z} \) contains a vector \( Q \) consisting of 313 values where each represents the probability of the pixel belonging to that color class in the quantized ab plane.

In order to convert the probability distribution to point color estimate, we could take mean of the distribution. But, the distribution is not Gaussian. So, mean is spatially coherent but it results in unnatural desaturated color. Mode of the distribution results in vibrant colors but it breaks the spatial consistency. So, we interpolate between mean and mode estimates, the result is the annealed-mean of the distribution, controlled by a parameter known as Temperature, \( T \). [8] This captures the vibrancy of the mode while maintaining the spatial consistency of the mean.
each pixel, we perform the following independent transformation:
\[ \hat{Y}_{h,w} = \mathcal{H}(\hat{Z}_{h,w}) \]  
(2)

\[ \mathcal{H}(\hat{Z}_{h,w}) = \mathbb{E} \left[ f_1(\hat{Z}_{h,w}) \right] \]  
(3)

\[ f_1(z) = \mathbb{E} \left[ f_1 \left( \frac{\exp(\log(z))}{\sum_q \exp(\log(z_q))} \right) \right] \]  
(4)

C. Loss Function

The Loss function used in [2] is the Euclidean Loss Function, between the predicted and the ground truth colors. However, this loss is not suited for colorization due to its inherent ambiguity and failure to address the multimodal nature of the colorization problem as optimal solution to Euclidean loss is the mean, which results in averaging effect favoring grayish, desaturated colors. [8]. So, we use a custom loss function tailored specifically for the colorization purpose:

\[ L(\hat{z}, Z) = -\frac{1}{HW} \sum_{h,w} v(Z_{h,w}) \sum_q Z_{h,w,q} \log(\hat{Z}_{h,w,q}) \]  
(5)

where,

\( \hat{Z} = \) predicted color,

\( Z = \) ground truth color, and

\( v(.) = \) color rebalancing term

D. Class Rebalancing

In our dataset comprising of the ImageNet dataset [14] and 5000 added images, the distribution of ab values is strongly biased towards lower values in ab space. So, these values dominate the loss function resulting in desaturated colors. So, we reweighted the loss of each pixel at training on the basis of the rarity of the color pixel. This approach is asymptotically equivalent to resampling of the training space. [12]. Here, each pixel is weighed by the factor \( w \) based on its closest ab bin.

\[ v(Z_{h,w}) = w_q \]  
(6)

\[ w \propto \left( 1 - \lambda \right) \rho + \lambda \frac{1}{Q} \]  
(7)

where,

\( Q = \) quantization bins in ab color space (\( \approx 313 \))

\( \lambda = \) weight factor For our research, we used \( \lambda = 0.5 \) for smooth empirical probability distribution \( \rho \).

V. ALGORITHM

A. Training Algorithm

The dataset consists of images from ImageNet as well as about 5000 images related to the natural, cultural and historical aspects of Nepal, downloaded from various sources over the Internet. Model training follows the following process (for each train image):

1) Convert the train image into Lab color space, with L values ranging from 0 to 100 and ab values form -100 to +100. Also, resize it to 224x224 for input to the neural network.

2) Slice the Lab plane into L and ab channels with L being the input signal and ab as the supervisory signal.

3) Perform CONV block operations on the L channel, obtaining 56x56 output, with each output pixel being a 313 probability distribution vector.

4) Perform interpolation between mean and mode with T=0.4, obtaining a single ab pair for each pixel’s distribution in the ab plane, using the equation 4.

5) Use the custom loss function to reconfigure the kernels in the CNN blocks.

6) Repeat the process 1-5 for each training image in the training set.

7) Obtain the model.

B. Colorization Algorithm

After the network is trained, we obtain a colorization model. Following steps are used to colorize an image from the fully trained model.

1) Convert the RGB image to Lab Color Space

2) Rescale the image to 224x224 resolution and slice the image into L and ab planes

3) Centre the L values in the range 0 to 100.

4) Feed the rescaled and centered L channel from the image to the CNN

5) Obtain the 56x56 ab channel output from the network. Upsample the output to 224x224 resolution.

6) Convateenate the output ab channel with the L channel from (3) to obtain the colorized image.

VI. OUTPUT

We performed the Turing Test of our colorization method to test its effectiveness. The images selected for the method were one of the best colorized outputs from the process. The results were varying, but an average of 4 images appeared as real to each participant. Thus, we can deduce that the process is 40% effective, which is higher than the previous methods.

We have successfully implemented the method and tested it on various images ranging from legacy grayscale images to modern images. Legacy images have no predefined colorization known to the model. But, the modern images can be readily colorized as model learned the artifacts from such images in the dataset. Some notable colorization outputs are as follows:

1) dhara.png

(a) Grayscale Input  (b) Colorized Output
2) royal bengal tiger.jpg

(a) Grayscale Input  
(b) Colorized Output

3) rainbow lorikeet.jpg

(a) Grayscale Input  
(b) Colorized Output

4) pug.jpg

(a) Grayscale Input  
(b) Colorized Output

5) mahalangur mountain range.jpg

(a) Grayscale Input  
(b) Colorized Output

6) gurkhas in north africa ww2.jpg

(a) Grayscale Input  
(b) Colorized Output

ACKNOWLEDGMENT

A lot of helping hands were involved in this research. We would like to express our heartfelt gratitude to Asst. Prof. Shankar Subedi, at the Central Department of English, TU for proofreading the paper. Our sincere thanks goes to Er. Rajeev Prajapati, Er. Surendra Tamrakar, Er. Rabindera Khati (HOD) at the Department of Computer and Electronics for their support, encouragement and continuous feedback. We thank everyone who was instrumental in the completion of this project. We would like to acknowledge Dinesh Gautam, Avishek Dev, Sushil Dahal and all the intellectuals and professionals whose work were contributory in this research.

REFERENCES


