Colorization of Grayscale Images using CNNs

Anushka Bhat1  
[bhata04@pfw.edu](mailto:bhata04@pfw.edu)

Abhinay Belde1[belda01@pfw.edu](mailto:belda01@pfw.edu)

Shubhdeep Sarkar1  
[sarks03@pfw.edu](mailto:sarks03@pfw.edu)

1 Computer Science Department, Purdue University Fort Wayne

Fort Wayne, IN, 46805,

**Abstract.**  
In In this paper, we discuss our findings of working with convolutional neural networks to colorize grayscale images. We start with introduction by giving a short description about the motivation, description, applications of colorization, challenges, data analysis and preprocessing, approach and the findings. This is followed by a detailed description of data exploration and preprocessing, methodology, experiments performed, model evaluation. In the end, we present our conclusion based on our findings. The paper also includes a section for related works and references.

**Keywords:** Convolutional Neural Networks, Colorization, Grayscale Images.

1. Introduction

**Motivation:** Colorization of images can play an important part in restoring the images with faded colors. It can also help in revitalizing the historical images which are originally not having color. Apart from this, leveraging convolutional neural networks for colorization contributes to a more comprehensive understanding of visual content, particularly benefitting users with visual impairments.

**Problem statement:** Colorizing the grayscale images using convolutional neural networks.

**Applications:**Colorization has many applications like photo restoration, film colorization, artistic rendering, and enhancing text-to-speech applications.

**Challenges encountered:**

* Availability of large enough grayscale dataset was a huge challenge as grayscale images are not common to be found.
* Even after getting the dataset, training was a huge challenge due to computational constraints. Many times code execution ended in session getting crashed.

**Description of data:** Data was taken from <https://cocodataset.org/#download> from which we took a subset of images to train the model.

**Description of approach, methods, or machine learning algorithms:**We employed two CNN architectures for the task. The first one was adopted from <https://www.kaggle.com/code/basu369victor/image-colorization-basic-implementation-with-cnn>,while the second model involved our own experimentation with the architecture. Our goal was to experiment with less complex CNN architectures to find their potential for image colorization.

**Result and findings:**The performance of both models was assessed across various hyper-parameter combinations using the MSE metric and human evaluation. Ultimately, a conclusion was drawn regarding which model outperforms the other under specific hyper-parameter settings. An interesting observation was made about MSE not being a perfect metric to measure accurate colorization.

1. Problem Statement

The primary objective is to explore and implement CNN architectures, evaluate their performance through metrics such as Mean Squared Error (MSE) and human evaluation, and ultimately determine the most effective model and hyperparameter combination for accurate and visually pleasing image colorization. To achieve this, several experiments were carried out. Two different combinations of hyperparameters (batch\_size, epochs, learning\_rate) were chosen to carry out multiple iterations for the two architectures. Validation loss and training loss were recorded for all the models and finally average MSE was recorded to evaluate their performance on unseen (test) data.

1. Related Works

Various computer vision research endeavors exist, with the colorization of grayscale images standing out as a longstanding challenge. Convolutional Neural Networks (CNNs) have recently gained prominence as potent tools for image-to-image translation tasks, including colorization. This section delves into a concise review of pertinent works in this domain.

**Recolorizing Black and White Landscape Images Using CNNs [2]**

The authors detail their efforts in colorizing black and white landscape images employing CNNs. The final model incorporates an Inception-ResNet-v2 Classifier, enhancing its object classification capabilities. Notably, this model exhibits superior results with reduced computational resources, employing eight two-dimensional convolutional layers, a maximum of 40 epochs, and a batch size of 20.

**Coloring Black and White Images using Neural Networks [3]**   
Emil Wallner's blogpost on Medium provides a beginner-friendly overview of his image colorization project using neural networks. He employs Inception-ResNet-v2 as the model base and implements three versions: alpha, beta, and a full version. The alpha version introduces a dual-tone RGB variant, while the beta version addresses overfitting, resulting in images with a brownish tone but more colors overall. In the full version, a fusion layer is added between the encoder and decoder, with the encoder running parallel to Inception ResNet v2 as a classification layer. This project serves as a valuable starting point, offering different levels of understanding for building a functional image colorization model.

**Deep Colorization [1]**

This research explores the utilization of deep neural networks for colorizing black and white images. Motivated by the ability of deep learning to handle large datasets and produce artifact-free results, the authors group images into clusters and extract global descriptors. Each cluster is associated with a dedicated deep neural network. During testing, the model identifies the network with descriptors closest to the test image, refines chrominance values, and combines them with the original grayscale input to generate the final-colored version. Despite achieving approximately 80% pixel accuracy, the model has limitations in colorizing synthetic images.

1. Dataset Description

The dataset used for our study was sourced from the official COCO (Common Objects in Context) dataset, accessible at <https://cocodataset.org/#download>. Initially, a pool of 84,000 images was obtained, from which a subset of 10,000 images was randomly chosen for the initial model training. However, to address computational limitations, dataset was further downsized to 5000 images. The dataset mostly had colored images but also included some black and white images.

1. Data exploration

During our dataset analysis, we noticed that many images were in color, and we needed to convert them to black and white for our training data. The colored versions of these images served as our target examples. On the other hand, some images were already in black and white, making them less useful for our model to learn the colorization process.

1. Data preprocess

In the data preprocessing phase, grayscale images were excluded as they lacked informative content crucial for the model's learning process. About 15 images out of the 5000 were eliminated, resulting in a final dataset of 4935 images. Subsequently, all images were resized to dimensions of 200\*200 pixels (length and breadth) to ensure a consistent size for the input layer of the model. This refined dataset was further divided into training and testing datasets, with the test size representing 0.01 percent of the entire dataset. The training dataset underwent additional division into training and validation datasets, where the validation dataset size constituted approximately 20 percent of the training dataset.

1. Our Approach

Two model architectures were implemented. One was a stacked convolutional network with MaxPooling, Batch normalization and up-sampling layers with LeakyRelu activation. The model begins with an input layer accepting grayscale images of shape (200, 200, 1). The encoder section employs Conv2D layers with increasing filters, Leaky ReLU activations, batch normalization, and max-pooling for spatial reduction. Following the encoder, intermediate layers with Conv2D, Leaky ReLU, and batch normalization are utilized. The decoder involves up-sampling and concatenation of feature maps, followed by Conv2D layers for decoding. The output layer produces RGB colorized images using Conv2D with 3 filters and 'tanh' activation. The second architecture entails less number of layers and is less tailored in contrast to the first one. The input layer accepts grayscale images with dimensions (200, 200, 1). The encoder section comprises two sets of Conv2D layers with 64 and 128 filters, respectively, each followed by ReLU activation and same-padding. Max-pooling is applied after the first set. The middle section consists of two Conv2D layers with 256 filters each. The decoder involves up-sampling and concatenation of feature maps from the encoder, followed by Conv2D layers to reconstruct the image. The output layer produces RGB colorized images with three channels, utilizing sigmoid activation.

1. Results

In the initial iteration for both *Model 1* and *Model 2*, we conducted training sessions with key parameters set as follows: *epoch=10, learning\_rate=0.001,* and *batch\_size=16*. The chosen optimizer for both models was the *‘Adam’* optimizer. Throughout the training process, we monitored and recorded both the training loss and validation loss. The loss function employed for this analysis was the *‘mean\_squared\_error’*. This approach allowed us to assess the models’ performance and convergence, providing insights into their learning dynamics.

A graph of training and training loss

Description automatically generated

A graph of training and training loss

Description automatically generated

Overall training loss and validation loss show a decreasing trend with validation loss showing some sharp edges as the number of epochs increase.

Subsequently, an average Mean squared Error was calculated for all the test images for all iterations of both the models.

|  |  |  |
| --- | --- | --- |
| **Model** | **Iteration** | **MSE score** |
| Model 1 | 1 | 0.0091 |
| Model 1 | 2 | 0.0088 |
| Model 2 | 1 | 0.0083 |
| Model 2 | 2 | 0.0075 |

Overall, we do see less values of MSE being shown by the second model. However, we do go  for human evaluation to make our observations more conclusive.

**Human evaluation: Unseen test image.**

Model 1 iteration 2 can be seen giving better color but the MSE is higher than recorded for the model 2. It could also mean that certain pixels are getting wrongly colored. In contrast, model 2 did well for iteration 2 by showing some color and scoring low on MSE metric.

A close-up of a boat

Description automatically generated

To conclude, the model 1 iteration 2 can be seen giving better color but the MSE is higher than recorded for the model 2. In contrast, model 2 did well for iteration 2 by showing some color and scoring low on MSE metric.

1. Conclusion

All iterations show some color added to the image indicating that the models were able to learn to some extent. However, it was concluded that MSE alone could not be used to find conclusive results therefore human evaluation was necessary which helped us make the results more conclusive. To conclude, Model 2 with second combination of hyperparameters (epochs=20,learning\_rate=0.01,batch\_size=16) could be considered a good candidate for further exploration for this task.

References

1. Cheng, Zezhou, Qingxiong Yang, and Bin Sheng. "Deep colorization." *Proceedings of the IEEE international conference on computer vision*. 2015.
2. Zhu, Lauren, Jacqueline Ennis, and Jacob Reiter. "Recolorizing Black and White Landscape Images Using Convolutional Neural Networks."
3. Emil Wallner, <https://emilwallner.medium.com/colorize-b-w-photos-with-a-100-line-neural-network-53d9b4449f8d>, 2017.