Proceedings of 13th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2023 Month: April Volume: 13

Image Colorization Using AutoEncoder

Bikram Shah [a], Manoj K. Guragai [b], Anil Verma [c]

[a, b, c] *Department of Electronics and Computer Engineering, Purwanchal Campus, IOE, Tribhuvan University, Nepal*
[a] bikramshah213@gmail.com, [b] mkguragai@gmail.com, [c] anil@ioe.edu.np

Abstract
This paper explores the application of autoencoders for image colorization. The purpose of the paper is to develop a model that can accurately predict the color of a grayscale image. We trained the autoencoder on a dataset of grayscale images and their corresponding colored images. The autoencoder architecture consists of an encoder network that compresses the grayscale image into a lower-dimensional representation and a decoder network that reconstructs the colored image from the compressed representation. Our experiments show that the proposed model achieves high accuracy on a validation dataset, with an accuracy of above 75 percent. The results demonstrate the effectiveness of the proposed approach for image colorization and highlight the potential of autoencoders for computer vision tasks. The developed model has potential applications in various fields, including image restoration, video editing, and digital art.

Keywords
autoencoder, decoder, grayscale, colorization, restoration

1. Introduction

In this paper, it is aimed to tackle the problem of automatically colorizing grayscale images using autoencoder network. The input to the network is grayscale image then we use autoencoder to output a prediction of a realistic colorization of the image.

Despite the absence of color information in a black and white image, people may fill it with realistic hues by using contextual cues from the image's content. This shows that, despite the fact that it might take an individual many hours to add color to a single picture, B/W photographs still contain latent information that might be sufficient for full colorization [1]. CNN's remarkable accomplishments in feature extraction from photos represent a promising approach for quick automatic colorization that might be used for effective colorization of B/W videos. A great number of different domains can be affected by the colorization of black and white photos. Remastering of old photographs and improving surveillance feeds are a couple of uses for black and white image colorization. Black and white photographs have very little information. Therefore, by including color components, we can enhance the image's meaning[2].

Automatic image colorization addresses the issue of automatically adding colors to monochromatic photos. Colorization has some useful uses, like restoring color from ancient films or photos, helping artists, and creating visual effects. However, automatic image colorization is a useful model for a lot of issues. There are a ton of applications where we wish to take an arbitrary image and forecast values or various distributions at each pixel of the input image, relying solely on this input image's data. Using colored picture datasets, pre-built models like Inception and ResNet are trained. If we colorize the black and white photographs before using these neural networks on them, the outcomes will be better[3]. How-ever Designing and implementing an active, dependable system to automate the entire colorization process is quite difficult nowadays. In this method, a deep convolutional neural network is constructed to take a grayscale image as input and generate a colorized image. Our black and white image is first converted to 224×224 pixels. This is what we feed into our neural network. By practicing on vivid photographs, our model is trained to produce shots with realistic colors. The created visuals could readily deceive a viewer[4].

The RGB color space is a 3-channel color space. CIE Lab color space is similar to RGB color space but the only difference is that the color information is encoded only in the "a" and "b" channels. The L (lightness) channel only encodes the intensity, so we can use it as our grayscale input to the neural network. The trained network will predict ab channels. Now the produced ab channels will be combined with L channel. Finally, we will convert the "Lab" image back to the RGB color space.

2. Literature Review

Early approaches to colorization relied on some amount of human effort, either to identify a relevant source color image from which the colors could be transferred or to get a rough coloring from a human annotator to serve as a set of "hints". More recently, there has been a surge of interest in developing fully automated solutions, which do not require human interaction. Most recent methods train a CNN to map a gray input image to a single color image. When such models are trained with L2 or L1 loss, the colorization results often look somewhat "washed out", since the model is encouraged to predict the average color. Some recent papers discretize the color space, and use a per-pixel cross-entropy loss on the SoftMax outputs of a CNN, resulting in more colorful pictures, especially if rare colors are upweighted during training[5]. However, since the model predicts each pixel independently, the one-to-many nature of the task is not captured properly, e.g., all of the pixels in a region cannot be constrained to have the same color.

This study[6] proposed an unsupervised method for image colorization that leverages self-supervised learning to learn the relationships between grayscale images and their corresponding color images. The approach was tested on various datasets and achieved competitive results without using any manual annotations.

3. Methodology

3.1 Convolution Neural Network

CNNs are a type of deep learning neural network that is commonly used for image classification and object recognition tasks. They work by processing the image in small, overlapping regions, or filters, and learning to recognize patterns in these regions. By stacking multiple layers of filters, the network can learn to recognize more complex features in the image[7].

3.2 Autoencoders

Autoencoders are a type of neural network that can learn to compress and decompress data. They consist of an encoder network that learns to encode the input data into a lower-dimensional representation, and a decoder network that learns to decode this representation back into the original data. In the case of image colorization, the encoder network takes in the grayscale image and produces a lower-dimensional representation, which is then decoded by the decoder network into a color image.

3.3 Optimization Algorithm

Optimization algorithms are used to update the model parameters during training in order to minimize the loss function. The most commonly used optimization algorithm is, Adam, which updates the model parameters based on the gradient of the loss function with respect to the parameters.

$$W_{t+1} = W_t - \alpha m_t \tag{1}$$

where, $m_t = \beta m_{t-1} + (1-\beta)\delta L/\delta W_t$

3.4 Loss Function

Loss functions are used to evaluate how well the model is performing during training. In the case of image colorization, a common loss function is mean squared error (MSE), which measures the difference between the predicted color image and the ground truth color image. Here we actually compute the reconstruction loss using Mean Square Loss.

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(y_i - y_{i'})^2 \tag{2}$$

3.5 Our Approach

3.5.1 Datasets

One of the best well known source is CIFAR 10 data set of 60,000×32×32 RGB color images in 10 classes.It is chosen for the reason that smaller images are much faster to train and limiting training classes constrains the space potential objects that the model would need ti learn how to color.

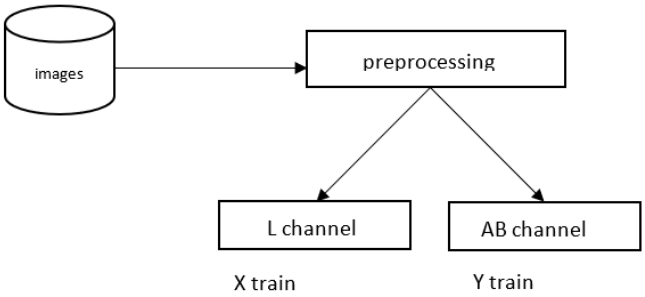

Figure 1: image preprocessing

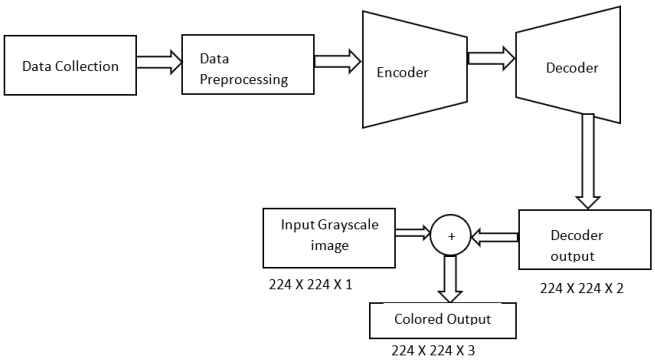

Figure 2: Model Architecture

3.5.2 Preprocessing

To each image in the dataset, the following steps is applied as shown in figure 1.

1. Convert the image from RGB color space to LAB color space.
2. Extract L channel from image so that it can be used as the X train for the model.
3. Extract AB channels from the image which can be used as Y train for the model.

3.5.3 Architecture

The autoencoder architecture consists of an encoder network that maps the input grayscale image to a lower-dimensional feature space and a decoder network that maps the encoded features to a colorized image. The goal is to train the autoencoder to learn the mapping from the grayscale image to its corresponding colorized version. During training, the autoencoder tries to minimize the difference between the output colorized image and the ground truth color image using a loss function such as mean squared error or binary cross-entropy. After training, the autoencoder can be used to generate colorized images for new grayscale input images.

The block diagram of proposed architecture is shown in figure 2.In this approach, a deep convolutional neural network based autoencoder can be used that takes a grayscale image as an input and produces a colorized image. Firstly B/W images are converted to 224×224 pixel then it is fed as an input to the neural network. The model can be trained to produce photos with realistic colors by training on colorful images. The images produced would easily fool a viewer. The RGB color space is a

3-channel color space. CIE Lab color space is similar to RGB color space but the only difference is that the color information is encoded only in the "a" and "b" channels. The L (lightness) channel only encodes the intensity, so it can be used as the grayscale input to the neural network. The trained network will predict ab channels. Now the produced ab channels can be merged with L channel as the process of post processing. Finally, the "Lab" image can be converted back to the RGB color space. The proposed method includes VGG16 model whose top 3 dense layers are excluded as the feature extractor. It can be considered as the encoder part of the auto encoder.

The summary of the encoder part is as follows:
Convolution +ReLU = 5 layers
Max pooling layers = 5 layers

The functionality of each layer can be summarized in following ways. The input layer takes the grayscale image of size 224×224×1. It is fed to the first convolution layer. The conv1 is constructed of 64 filters and of (3×3) size. Hence it produces a feature of size (224×224×64). Then a max pooling layer of size 2×2 is applied that produces output of size 112×112×64. Again, a conv2 layer is applied that produces out put of size 112×112×128. Similarly, 5 convolution layer and max pooling layer is used that produces features of size 7×7×512 that can be used as the input to the decoder part.

The input to the decoder is (7×7×512) features of image. It is passed through a series of up-sampling and convolutional layers. A custom decoder is used to regenerate the colorized version of the input image. Then the input grayscale image is added with the output that is generated by the decoder network to produce the colored version of the input image.

4. Results and Conclusion

After the successful training of the model upto 20000 epochs, we got the model accuracy of about 77percent.The optimizer used was Adam optimizer with learning rate of 0.0001 and the loss function used was Mean Square Error (MSE) loss function.The following images describes the output the work.

Figure 3: Output1

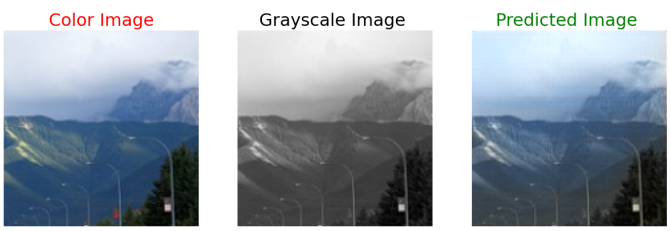

Figure 4: Output2

Figure 5: Output3

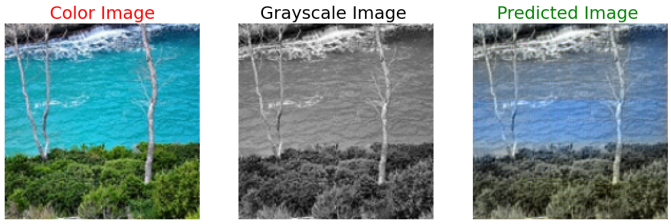

Figure 6: Output4

Figure 7: Output5

Acknowledgments

The work on image colorization is supported by Department Of Electronics and Computer Engineering, IOE Purwanchal Campus Dharan in 2023. The authors are grateful and pay deepest gratitude to Asst. Professor Om Prakash Dhakal, Asst. Professor Sabin Kafley and Asst. Professor T. N. Jha whose sharp observations and suggestions in the very early and last phase of writing certainly enhanced the quality of the work.

References

[1] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

[2] Firas Laakom, Jenni Raitoharju, Alexandros Iosifidis, Jarno Nikkanen, and Moncef Gabbouj. Color constancy convolutional autoencoder. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1085–1090. IEEE, 2019.

[3] Sergio Guadarrama, Ryan Dahl, David Bieber, Mohammad Norouzi, Jonathon Shlens, and Kevin Murphy. Pixcolor: Pixel recursive colorization. *arXiv preprint arXiv:1705.07208*, 2017.

[4] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.

[5] Jeff Hwang and You Zhou. Image colorization with deep convolutional neural networks. In *Stanford University, Tech. Rep.* 2016.

[6] Qian Sun, Yan Chen, Wenyuan Tao, Han Jiang, Mu Zhang, Kan Chen, and Marius Erdt. A gan-based approach toward architectural line drawing colorization prototyping. *The Visual Computer*, pages 1–18, 2022.

[7] Min Wu, Xin Jin, Qian Jiang, Shin-jye Lee, Wentao Liang, Guo Lin, and Shaowen Yao. Remote sensing image colorization using symmetrical multi-scale dcgan in yuv color space. *The Visual Computer*, 37:1707–1729, 2021.