CIS 490I-590K Deep Learning, Winter 2023

Final Project Report – Image Colorization with CNN

Omkar Bhatkande

1. Tasks completed:

- Dataset collection
- Review related technologies approach methods and architecture details
- Model coding and test runs
- Data loading and pre-processing
- Possible improvements and Hyperparameter tuning on default model

2. Time schedule for completing the project.

- Literature review and data collection - 19 March
- Model evaluation and coding - 27 March
- Generating results, reports - 9 April

3. Introduction:

With image colorization, we want to take a grayscale input image and turn it into a colorful image. Due to the multimodal nature of this issue, numerous possible colored images may correlate to a single grayscale image. Traditional models, as a result, frequently required substantial human input in addition to a grayscale image.

Deep neural networks have demonstrated astounding ability in automatically colorizing images, converting them from monochrome to color without any extra human input. Although we are yet unsure of exactly what makes these models perform so well, their ability to acquire and apply semantic information—i.e., what the image is—during colorization may be a contributing factor.

In this study, image colorization is investigated using a deep learning approach with CNN. In this paper, we will provide our research on various methodologies, assess it, and talk about the findings to draw conclusions and future directions.

4. Related technologies:

Deep Learning Algorithms

A family of machine learning algorithms known as "deep learning" may discover patterns and features in huge datasets. By training on a dataset of grayscale photos and their equivalent color versions, they can also be utilized for image colorization. Based on its learnt patterns and attributes, the neural network may learn to forecast the most likely colors for a given grayscale image.

The autoencoder, a well-liked deep learning approach for picture colorization, combines a decoder network that reconstructs the input image from the compressed representation with an encoder network that compresses the input image into a lower-dimensional representation. To train the autoencoder to predict colors for fresh grayscale images, grayscale photos and their corresponding color counterparts can be used as training data.

Convolutional Neural Networks (CNNs)

CNNs are a class of deep learning algorithm frequently used for colorization and image identification applications. They are made up of several convolutional filter layers that may be trained to recognize patterns and features in an input image. To train CNNs to predict colors for new grayscale photos, a dataset of color versions of the corresponding grayscale images can be used.

The Colorful Image Colorization technique, which employs a CNN to predict color values in the Lab color space, is one well-liked CNN-based method for image colorization. To anticipate the a and b color values, the system first converts a grayscale image to the Lab color space. The grayscale image and the expected color values are combined to create the final color image.

Generative Adversarial Networks (GANs)

A generator network and a discriminator network make up the two networks that make up the deep learning method known as GANs. While the discriminator network tries to tell the difference between fake and real images, the generator network creates artificial images. Until the generator creates images that are identical to real photos, this procedure is continued.

By teaching the generating network to create color versions of grayscale photos, GANs can also be used to colorize photographs. To distinguish between generated color images and actual color images, the discriminator network is trained. By attempting to deceive the discriminator network, the generator network gains the ability to create realistic color visuals.

Image segmentation

Image segmentation is the process of breaking a picture into several sections or segments based on specific traits, such as texture or color. By giving distinct parts of a grayscale image a color, this method can be used to colorize the image.

SLIC (Simple Linear Iterative Clustering), a well-liked image segmentation algorithm used for picture colorization, separates an image based on its color and spatial proximity. Colors can be sampled from a training dataset of color images or applied to the segments using a user-defined color palette.

5. Data Description:

We will be using images from the Oxford IIIT pet dataset: https://www.kaggle.com/datasets/tanlikesmath/the-oxfordiiit-pet-dataset. The Oxford-IIIT Pet Dataset is a 37-category pet dataset with roughly 200 images for each class created by the Visual Geometry Group at Oxford. We will be training the model on images of basset hound which is a subset from the above dataset. Training images = 134, Validation images = 36, Test images = 30.

6. Methodology

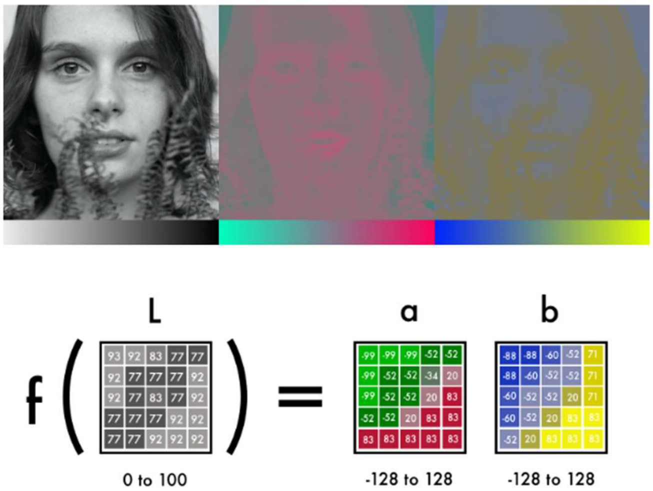

Figure 1: Lab color space, L stands for lightness, and a and b for the color spectrums green–red and blue–yellow.

The color channels will first be switched from RGB to Lab. L represents for lightness, whereas a and b stand for the green-red and blue-yellow color spectra, respectively. As you can see in the Figure 1, a Lab encoded image includes two layers of color and one layer of grayscale. This indicates that for our final prediction, we can use the original grayscale image. We only have two channels available to us for prediction.

94% of the cells in our eyes, according to science, control brightness. Only 6% of our receptors are still available to serve as color sensors. The grayscale image is much sharper than the color layers, as you can see in the image above. Another justification for keeping the grayscale image in our final prediction is this.

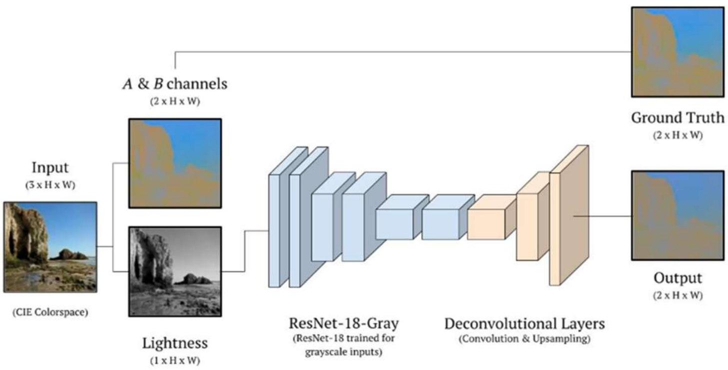

Figure 2: Model architecture having 6 layers of ResNet-18 followed by deconvolutional layers.

As seen in Figure 2, The model will have 6 layers of ResNet-18, an image classification network with 18 layers and residual connections, will serve as the foundation of our model, to extract features from the grayscale image. The first layer of the network is modified to accept grayscale images. Deconvolutional layers are added to upscale (increase the spatial resolution) the final image. ResNet-18, an image classification network with 18 layers and residual connections, will serve as the foundation of our model.

The 3 Lab channel image will be split into L (grayscale) image and ab (color) image. The grayscale image will be used by the model to predict the ab values & generate a ab image and will be compared with the original ground truth ab image. In the end, the generated ab image will be combined with the original grayscale to get the final output.

Data

200 images of basset hound were selected as a dataset in this project. 134 images were selected for training, 36 were used for validation and 30 were used as test images. All the images were resized to 224 by 224 for training.

Experiment

MSE was the loss function used for training. In the convolution layers, relu was used as the activation function. The starting learning rate was 1e-2 used with adam optimizer. The model was trained for 1000 epochs which took around approximately 45 minutes to completely train the model. Training Loss at 1000^{th} epoch = 0.0008 & validation Loss at 1000 epoch = 0.0025.

Evaluation

Model(s) to be evaluated using PSNR ((Peak Signal-to-Noise Ratio)

Results

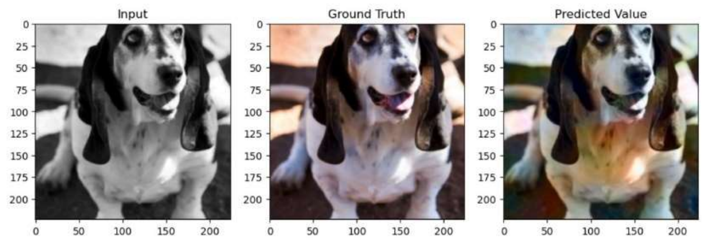

Figure 3: PSNR = 69.1381

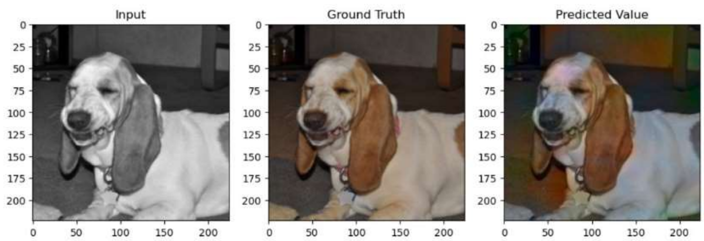

Figure 4: PSNR = 72.4457

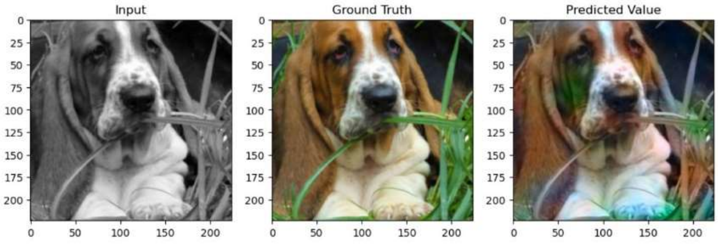

Figure 5: PSNR = 72.4457

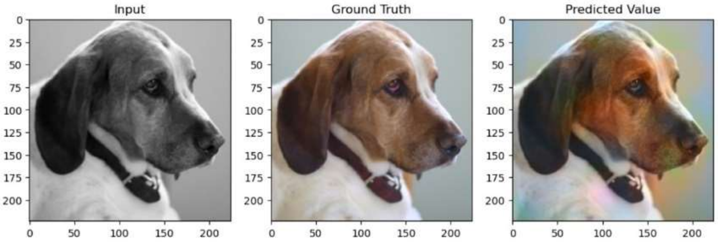

Figure 6: PSNR = 71.8678

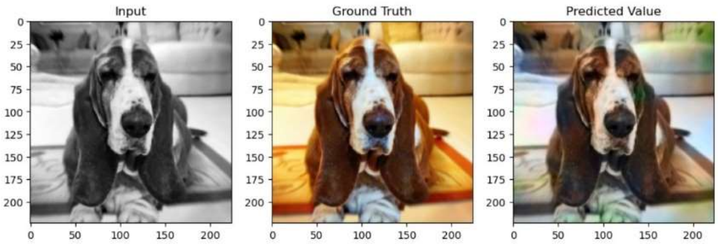

Figure 7: PSNR = 65.4298

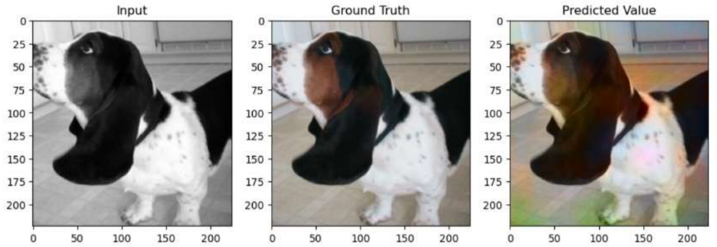

Figure 8: PSNR = 71.9322

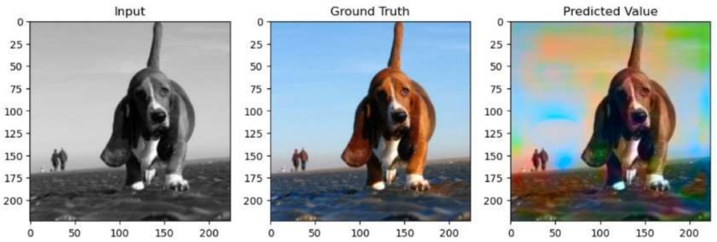

Figure 9: PSNR = 63.0434

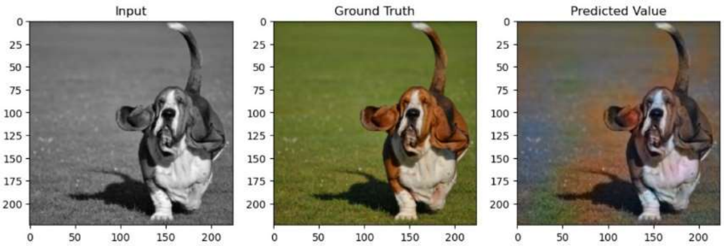

Figure 10: PSNR = 68.8375

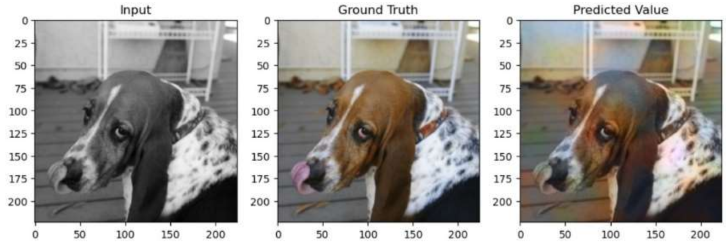

Figure 11: PSNR = 71.3505

Figure 12: PSNR = 68.5611

Average PSNR value is around 69.50. From Figure 3 to Figure 8 we can see that the model does a good job at recreating the original colors of the Basset Hound, though there are some color errors in the background. The model struggle with background such as grass and sky can be seen in Figure 9 and Figure 10. Also in Figure 12, there are some green color errors on the basset hound as the model may wrongly think of the texture as grass.

8. Conclusion

The model does a good job at coloring most of the basset hound images. The model struggles on green grass and blue-sky backgrounds, the reason for this maybe the small size of training data. The model can be trained for more epochs with a larger dataset for better performance. The model can be further improved by adding some novel feature to get the semantic information from the image i.e. adding image segmentation and training the model on different segments of the image.

9. References

[1] Zhang, R., Isola, P., Efros, A.A. (2016). Colorful Image Colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9907. Springer, Cham.

[2] Anwar, Saeed & Tahir, Muhammad & Li, Chongyi & Mian, Ajmal & Khan, Fahad & Muzaffar, Abdul. (2020). Image Colorization: A Survey and Dataset.

[3] Chen, Shu-Yu & Zhang, Jia-Qi & Zhao, You-You & Rosin, Paul & Lai, Yu-Kun & Gao, Lin. (2022). A review of image and video colorization: From analogies to deep learning. Visual Informatics. 6. 10.1016/j.visinf.2022.05.003.

[4] https://lukemelas.github.io/image-colorization.html

[5] https://blog.floydhub.com/colorizing-b-w-photos-with-neural-networks/