Image Colorization Using CNN

Angela Zhuang
Department of ECE
University of Toronto
angela.zhuang@mail.utoronto.ca

Amos Zhou
Department of ECE
University of Toronto
hr.zhou@mail.utoronto.ca

Yue Chen
Department of ECE
University of Toronto
cherylyue.chen@mail.utoronto.ca

Abstract

In this report, we presented our adapted UNet implementations for grayscale image colorization on CIFAR10 and Flower102 datasets. We have proposed two UNet variations with different upsampling and downsampling techniques and configurations, and experiments and results are discussed. On both datasets, two variations of UNets result in similar losses, but the colorization outcome using Flower102 dataset results in better visual quality compared to CIFAR10 dataset on both models.

Assentation of Teamwork

Angela Zhuang	Implement adapted UNets, hyperparameter tuning, and collaborate on the report.
Amos Zhou	Implement adapted UNets, hyperparameter tuning, and collaborate on the report.
Yue Chen	Implement adapted UNets, hyperparameter tuning, and collaborate on the report.

1 Introduction

This project explores the use of neural network models to solve the challenge of colorizing grayscale images. Traditionally, this task requires meticulous manual effort and artistic input to achieve realistic results. The process of transforming monochrome photographs or frames into colorized versions has both aesthetic and practical significance, serving purposes across digital media restoration, visual arts, medical imaging, enhancing historical archive materials, etc.. Automating this task reduces the time and effort required, allowing for images processing at scale.

Our approach employs deep learning techniques, specifically focusing on CNNs, autoencoders, and UNet architectures, to learn the complex image-to-image mappings between grayscale inputs and their colorized outputs. The ultimate goal is to predict the color values (RGB/YUV) for each pixel in a grayscale image, guided by its inherent features and patterns. Convolutional layers are thus essential building blocks when constructing our model for its ability in feature extraction and learning hierarchical representations. On top of this, we utilized UNet architecture, a variant of autoencoder known for its effectiveness in image segmentation, and adapted a simple version of it for our target problem. UNet, which has a symmetric encoder-decoder structure with skip connections in between, facilitates the flow of image compression and reconstruction processes and allows the network to make more accurate color predictions by leveraging both low-level detail and high-level information.

2 Problem Specification

To automatically colorize grayscale images, as discussed above, the model developed needs to predict the pixel values as closely as the original values possible given the grayscale representations. Thus, to train the model, we would need grayscale and colorful image pairs as model inputs and targets. We first used the CIFAR-10 [1] dataset which consists of 60,000 32x32 color images (RGB 3-channels) in 10 classes, and 6000 images per class. This dataset allowed us to implement the model on various types of images and evaluate the colorization performance on different object and background combinations. We used 40,000 images for training, 10,000 for validation to test the model, tune and select the hyperparameters, and 10,000 for testing. Colorful images were converted into grayscale representations as inputs for training and the original colorful images were used as the target outputs of the model. We later experimented with an additional dataset of Oxford Flower 102 [2] and explored the model performance difference on a specific type of images. Results are discussed in later sections.

After researching and analyzing different deep learning techniques and model complexity, an autoencoder with UNet architecture was chosen to be the base of our model due to its ability and effectiveness in processing image related tasks. Autoencoder generally come with two parts, encoder and decoder. The encoder part consists of several convolutional layers and implements max pooling to get the latent representation of the grayscale image. Inputs are downsized and important features of the images are extracted. The decoder part also consists of several convolutional layers and the pictures are rebuilt and upsized to add back details and predicted colors. Encoder and decoder blocks are bridged with skip connections to retain early features and preserve details which may be lost in deeper layers after successive pooling operations.

During implementation, several aspects and different configurations of the model were considered and investigated to improve performance. Hyperparameters like the learning rate, the number of convolution, max-pooling layers and upsampling layers (model depth), the sequence of connection, and the skip connection structure were all examined, tested and tuned based on numerical and prediction performance.

The output dimension was kept the same as the input dimension. We used Adam as the optimizer due to its robustness for a wide range of applications. Mean squared error and L1 loss were used and results were compared since they were both suitable for image processing tasks. Batch normalization was also applied to center the outputs from layers and reduce noise in data during

training. We further considered dropout to tackle potential overfitting issues. The models were evaluated and compared using validation/test loss values between test output and actual target and the plots of learning curves. The colorization results were visually compared through side-by-side original and predicted image presentations.

3 Design Details

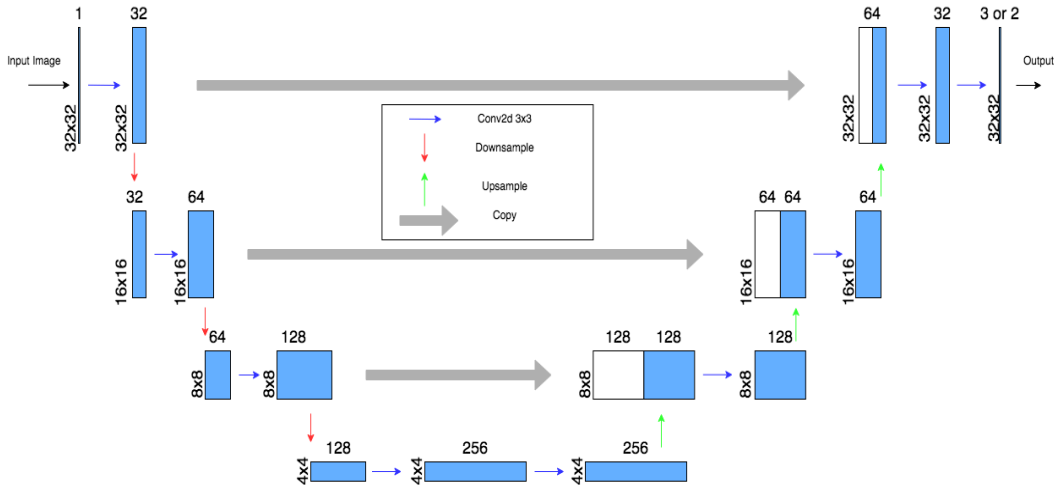

Figure 1: Implemented UNet Architecture for CIFAR10

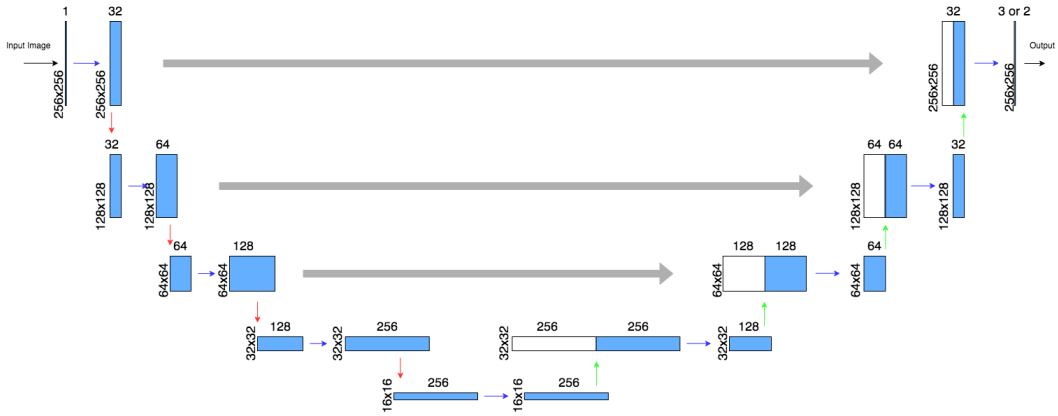

Figure 2: Implemented UNet Architecture for Flower102

The models implemented as illustrated in the figures above consist of a convolutional neural network designed for image colorization tasks. We chose to adapt UNet architecture for our autoencoder [3]. The network architecture is symmetric, with a contracting path to capture local image context, helping to extract and condense important features from input images. Then succeeded by an expansive path to progressively restoring the images that were condensed during the downsampling phase, enabling color restoration.

The contracting path comprises three sequential convolutional blocks (each with a convolution,

batch normalization, ReLU activation, and max pooling), progressively doubling the number of filters while reducing the spatial dimensions of the feature maps. This downsampling process captures increasingly abstract features at different scales. The bottleneck, a transitional layer without pooling, maintains the number of filters while applying convolution, batch normalization, and ReLU activation to process the most abstracted form of the input data. The expansive path mirrors the contracting path with three up-convolutional layers, each followed by a skip connection that concatenates the feature map from the corresponding level of the contracting path, enabling the network to use both high-level and low-level features for reconstruction.

The final output layer applies a Tanh activation function, normalizing the output values between -1 and 1. This design allows the UNet to efficiently learn and combine local and global information, making it highly effective for tasks that require detailed segmentation and precise spatial information such as image colorization.

Regarding the network's input and output channels, we used a single-channel grayscale as input, and the output channel from the final layer outputs varied based on images' color space. We first experimented with traditional RGB colorspace which has three channels. Later, during the data preprocessing stage, we transformed the RGB images into the YUV color space, which allowed us to predict only two channels, considering the YUV's first channel (Y, the luma component) represents the grayscale image. Our experimentation aimed to compare the results of predicting two channels in YUV space to predicting all three channels in RGB space.

Adding extra layers to the network architecture on top of the baseline architecture implemented in Figure 1 can potentially improve the performance because of the increased model complexity and the network's ability to learn more detailed features. In the image colorization model created by George Kamtziridis, the predicted color appears to be more vibrant and closer to the true RGB images as it moves to UNet architecture with an extra depth [4]. Thus, we chose to adapt the UNet architecture with additional layers for the Flower 102 dataset, aiming to achieve a better colorization result with a deeper UNet in Figure 2.

4 Numerical Experiments

We mainly implemented two versions of adapted UNet. The first version uses convolutions to extract features, max-pooling of size 2 to downsample, and upsampling of size 2 to interpolate. The second version, which is more complex, uses convolution of stride 2 to downsample, and transpose convolution function to upsample. In addition, we also performed experiments on different color spaces, RGB and YUV. Initially, the UNets output RGB images directly. After some research, we found YUV colorspace might be a better choice since we can keep the Y component as is and only generate UV components.

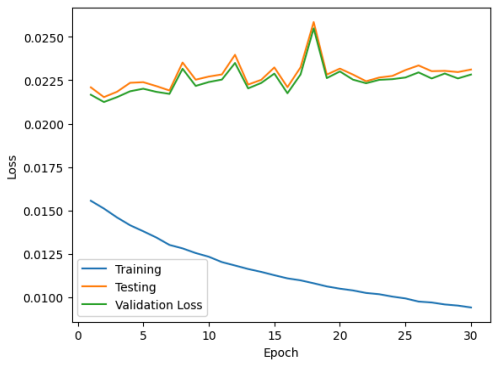

Figure 3: Simple UNet with RGB Output

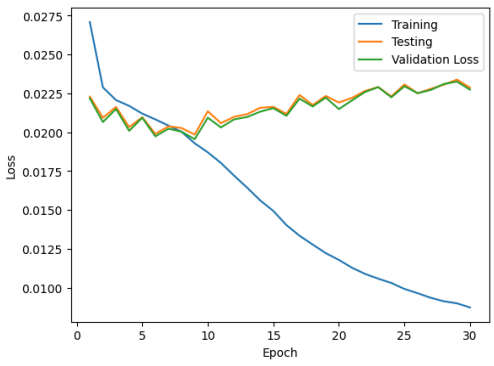

Figure 4: Complex UNet with RGB Output

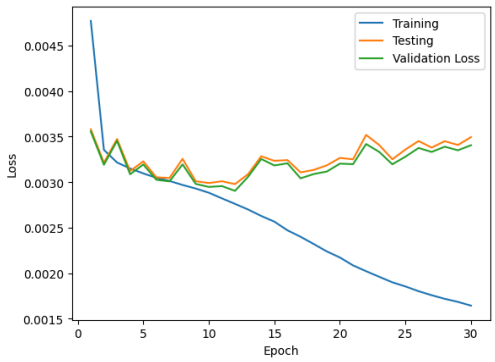

Figure 5: Simple UNet with UV Output

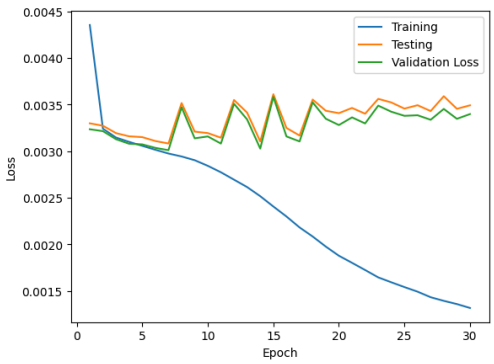

Figure 6: Complex UNet with UV Output

From the four plots above, we could see that the simple and complex adapted UNets eventually ended up with similar training, validation, testing loss under the same color space. These loss values numerically proved the simple and complex models generate similar colorization quality. By cross comparing the RGB outputs and UV outputs, we can see the UV output plots have significantly lower losses than RGB outputs. However, we can hardly notice the difference in colorization quality in RGB and YUV outputs shown by the comparison images below.

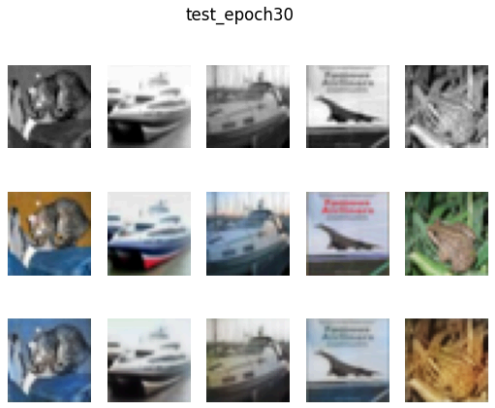

Figure 7: Simple UNet RGB output

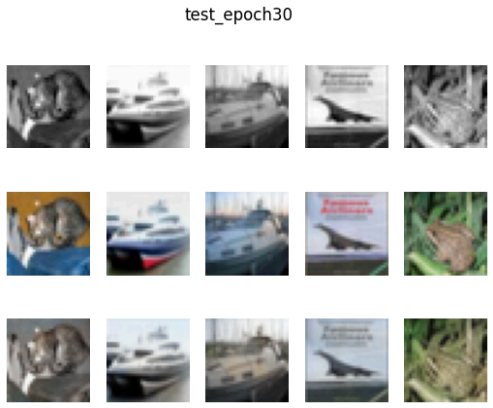

Figure 8: Complex UNet RGB output

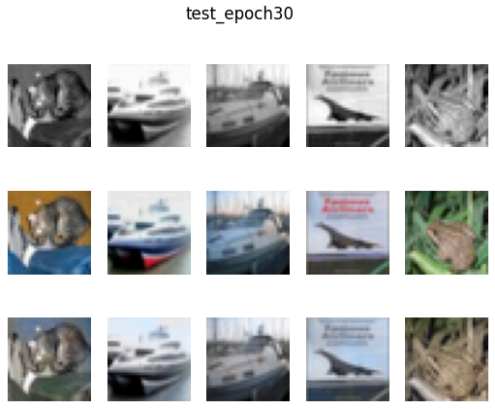

Figure 9: Simple UNet UV output

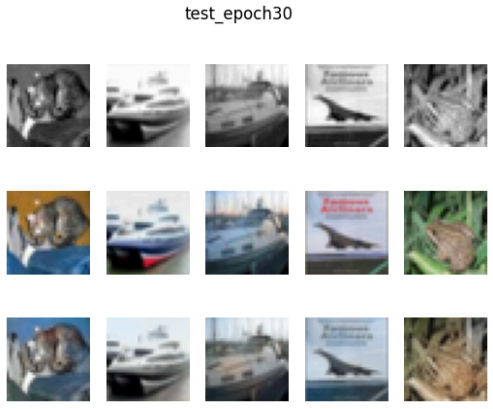

Figure 10: Complex UNet UV output

The above four images show the colorization quality of the adapted UNet models, with the first row being the input grayscale images, the second row being the target/original colorful images,

and the third row being the predicted colorized images. In general, the predicted images are not as vibrant and some colors are not predicted correctly. Images are paler than the original one with gray/brown hue, and red color is barely shown in the predicted images.

Later, we implemented simple and complex UNets using the Oxford Flower 102 dataset, a dataset consisting of colorful flowers resized to 256x256. We expected this dataset to be more appropriate for image colorization tasks as it has a single type of objects and more consistent patterns, so should be easily predictable when generating colorization results. To accommodate for the 256x256 image size, we modified the UNets with an extra layer of downsample and upsample to extract more features.

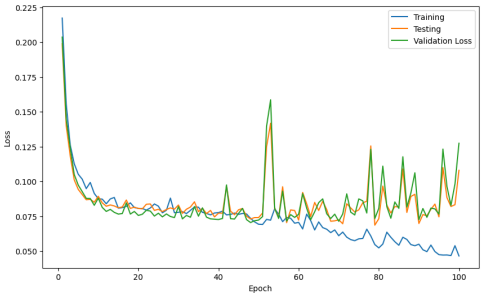

Figure 11: Simple UNet with RGB Output

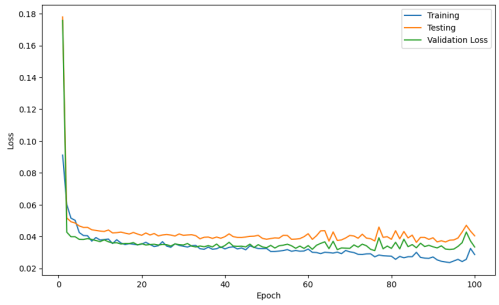

Figure 12: Simple UNet with UV Output

Comparing the loss plots on simple UNet and complex adapted UNet on Flower 102 dataset, we observed a similar pattern as using UV output results in a much lower loss. Although the CIFAR10 dataset results in a lower loss, we can visually observe that the model trained using Flower 102 dataset generates a much better colorization outcome, as shown below.

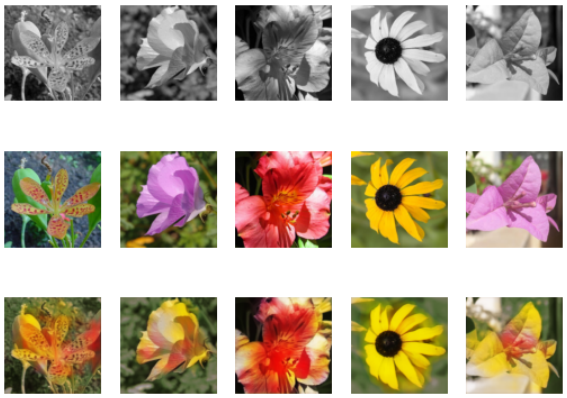

Figure 13: Test Images using UNet with RGB Output - 100 Epochs

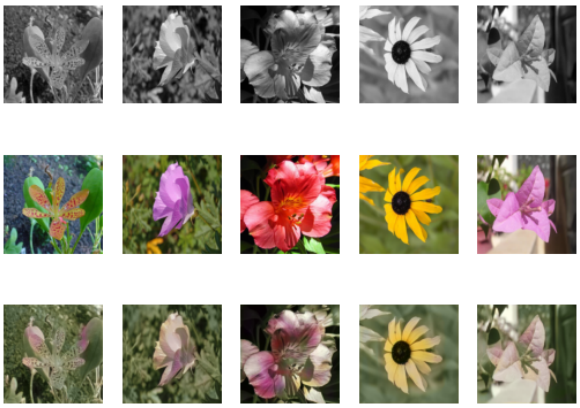

Figure 14: Test Images using UNet with UV Output - 100 Epochs

The two images above are generated using UNet and Flower102 dataset with the grayscale images as the first row, the original colorful images as the second row, and the predicted images as the third row. We can see that even though some flowers are still colored with different colors, overall the image colorization quality of RGB outputs is better than YUV outputs, and at the same time, much better than the one using the CIFAR10 dataset. This shows that the same model would generate different quality outputs using different color space representations or trained with different datasets.

Here we could conclude three main factors that affect model performance. First is that the Flower

102 dataset only has one class type, which is flowers, so naturally the model would perform well as it only sees the same type of inputs and are consistent in patterns. This allows us to implement a deeper UNet architecture to extract more features, which results in much more colorful output images compared to models trained with CIFAR10.

On the other hand, the CIFAR10 dataset contains ten different classes, which significantly increases the colorization difficulty as a lot of objects have random colors that are not consistent with related patterns and features. For example, there are cars with different colors, animals on a random colored carpet, and even images with color distribution that is very close to grayscale. The colors of those images have no specific pattern, as the same shape can have various colors, and a lot of images are even difficult for humans to predict the original colors from the grayscale images. Therefore, it is reasonable that the neural network model could not predict correct color results.

The second factor is the image resolution. We used Flower 102 dataset with 256x256 resolution and CIFAR10 with 32x32 resolution. Using UNet's architecture, each downsampling makes the image resolution be one quarter of the original one in order to extract features. With three stages of downsampling, Flower102 dataset results in a feature map of size 32x32, whereas CIFAR10 only has a feature map whose size is 4x4, which probably retains too few of extracted features and would be hard for the model to reconstruct correct outputs.

The third factor is that different color space representations would also affect the model outcome. As pixel values were transformed, the model was trained based on different ranges of data. Experiments were needed to determine the best performing color space for different neural network models and datasets.

5 Conclusions

In this project, we explored and adapted the UNet architecture to solve the challenge of colorizing grayscale images. We experimented with various UNet depth, applied different upsampling and downsampling methods, tested various batch sizes and learning rates, examined the impact of different color space transformation, utilized two datasets, and analyzed and optimized model performance. We discovered three factors that affect model performance as one is the appropriate dataset selection, second is image resolution, and thirdly, color space representations.

There are several improvements we can implement in the future. The first one is improving the existing UNet using the Generative Adversarial Networks (GANs) architecture, in which a new model called discriminator can be introduced. The discriminator determines if the input image is fake or real, whereas UNet tries to cheat the discriminator in order to improve the image quality. The second improvement is switching to a different dataset, such as CelebA. CelebA is a dataset of human faces. Images from a single category would allow the model to predict better results as discussed above. We could also further enhance the model by adding more layers and using datasets with higher image resolution. The third approach is adapting the new transformer model [5]. It should be able to achieve high fidelity and more detailed image colorization than traditional UNet.

References

[1] A.Krizhevsky, “CIFAR-10 and CIFAR-100 datasets,” Toronto.edu, 2009. https://www.cs.toronto.edu/~kriz/cifar.html

[2] M.-E. Nilsback and A. Zisserman, “Visual Geometry Group - University of Oxford,” www.robots.ox.ac.uk. https://www.robots.ox.ac.uk/~vgg/data/flowers/102/

[3] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” arXiv.org, May 18, 2015. https://arxiv.org/abs/1505.04597

[4] G. Kamtziridis, “Building an Image Colorization Neural Network — Part 4: Implementation,” *Medium*, Sep.19,2022.https://medium.com/@geokam/building-an-image-colorization-neural-network-part-4-implementation-7e8bb74616c (accessed Apr. 06, 2024).

[5] Kumar, Manoj, et al. “Colorization Transformer.” Computer Vision and Pattern Recognition, 7 Mar. 2021.

Appendix

Source Code File Description:

simpleUnet_RGB_CIFAR10.ipynb: contains the training and visualization functions for the simple UNet model with RGB output and the CIFAR10 dataset.

complexUnet_RGB_CIFAR10.ipynb: contains the training and visualization functions for the complex UNet model with RGB output and the CIFAR10 dataset.

simpleUnet_YUV_CIFAR10.ipynb: contains the training and visualization functions for the simple UNet model with UV output and the CIFAR10 dataset.

complexUnet_YUV_CIFAR10.ipynb: contains the training and visualization functions for the complex UNet model with UV output and the CIFAR10 dataset.

Unet_RGB_flower_smalldataset.ipynb: contains the training and visualization functions for the UNet model with RGB output and the Flower102 dataset.

Unet_YUV_flower_smalldataset.ipynb: contains the training and visualization functions for the UNet model with UV output and the Flower102 dataset.