CS 440
Spring 2021

Project 4: Colorization
Authors: Alden Lu (al1186), Haoran Wen (hw408)

1 Preface

To run the program, first run processing.py. Processing.py will cut the original images into a left and right and ready them to be used by the basic and improved agent.

Figure 1: Original Image

Figure 2: Left Cut

Figure 3: Right Cut

To run the basic agent run basic.py

To run the improved agent, run improve.py
The acronym for improved agent would be NCAA (Not-Completely-Accurate-Agent)

2 Basic

The basic agent came back looking somewhat visually satisfying as although there some random patches of color that shouldn't be there, we can identify the shape and overall color of a given portion of an image. Numerically, it seem to hover around 73 percent accuracy in comparison to a recolored version(using the colors of k-means clustering to color right side).Below will include Mat lab plot showing the image created . The procedure was as described in the assignment. The only real adjustment made is we sample 1000 pixels instead of the entire training data to

predict colors to reduce run time.

Procedure:

First take the left side of the image and run k-means clustering. This will create 5 colors to represent the original image. Then use those 5 colors and color in the left side and right side of the image. Next, take the grey scaled image of left and right side of the original image and run the prediction algorithm. This algorithm is essentially iterate every pixel of the grey scaled right image and identify 6 patches that look similar to a patch around the iterated pixel. To compare, we used the euclidean distance formula. Now for each of the 6 patches, we identify the corresponding pixel of the patch to the recolored left side and recorded the color. It is also important to note that the smaller the euclidean distance between the iterated patch (pixel patch we are trying to predict) and the 6 patches we identify as similar, the more likely the color is correct.Thus a priority queue was used to keep track of this. In addition, we took the mode of the collection of colors from the priority queue and used that color as the predicted color of our iterated pixel. If there happens to be a tie for the mode, the pixel patch with the smallest euclidean distance of the two will be chosen.

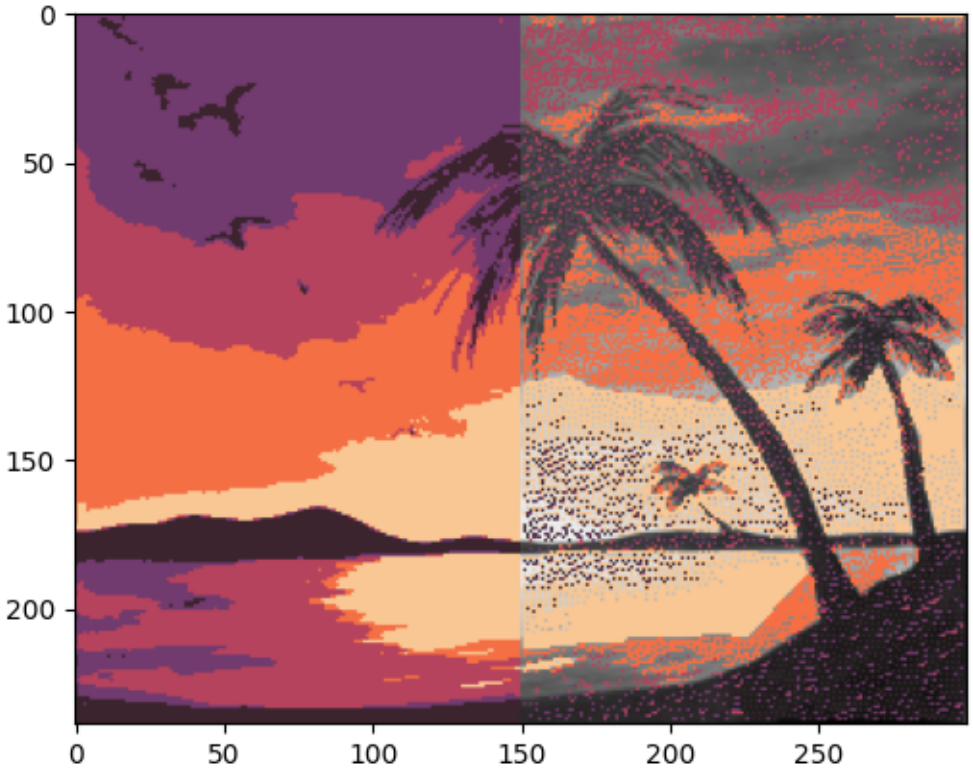

Figure 4: Basic Agent Result

3 Improved

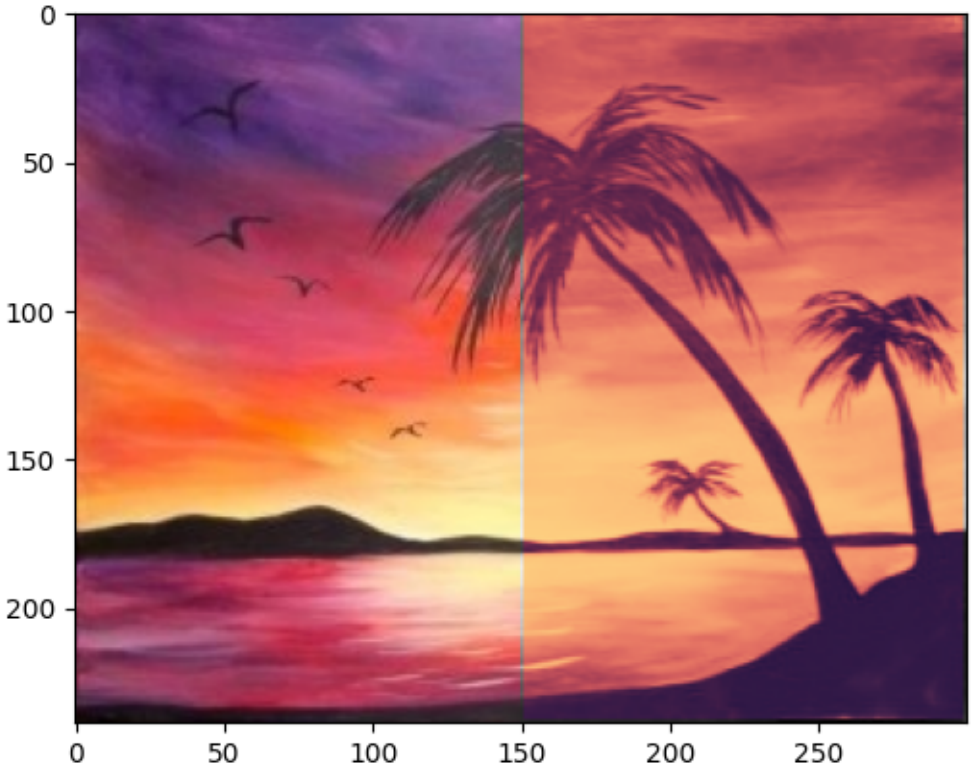

Figure 5: Improved Agent Result

For the improved Agent we used a neural network. Using the left image, the neural network trains and build a model that would then be used to colorize the right image. The neural network consist of 3 layers; an input layer, a hidden layer, and an output layer. The input layer has 9 nodes, the hidden layer has 5 nodes, and the output layer has 3 nodes. The input of the network are the 3 by 3 patches from the greyed image similar to the basic agent and the 9 nodes in the input layer corresponds to each pixel in a given 3 by 3 pixel patch. For the hidden layer we wanted to capture enough color data, not too much and not too few at the same time, and from online research and discussions 5 was the most commonly recommended number of nodes for a single hidden layer approach. The 3 nodes of the outer layer corresponds to the RGB color channels of the center target pixel in the 3 by 3 pixel patch. The input space are all the pixel patches and the output space are all the predicted color values.

In the pre-processing phase we created all the patches and stored them into a list and randomized the patches. We also operated out the patches into groups so that we can train on groups of patches multiple time instead of just training per patch. We also set the initial activation function based on the weights for the first patch in the list. In the training phase we

used gradient descent to adjust the weights for each of the patches in the first group. During this process we key track of derivatives of each patch and at the end using the average of the derivatives modify the weights. We repeat this processing on the same group using the newly adjusted weights for 100 iterations. Where each iterations the weighs are being adjusted. This process would be done on each of the groups of patches to produce the final weights that would be used to recolor the right side image.
In order to avoid over-fitting we tested out different patch group sizes and different iterations counts. If either the group size or iteration count is too high or too low the resulting image remains grey or the colors no longer resembles the original image. To allow to decrease the training time we also tried using an smaller image as well as setting the bias value to 0.

To find the cost function we computed the sum of squares of the difference between the predicted color values from the greyed left cut and the original colored left cut: $Cost = \sum(Original_i - Prediction_i)$. To find the weight derivative between the hidden and output layers we used: $\frac{dL}{dw_k^c} = \frac{dL}{dOutput_c^2} o'(w^c * output^1) output_k^1$ To find the weight derivative between input and hidden layer we sued: $\frac{dL}{dw_i^k} = \frac{dL}{dOutput_k^1} o'(w^k * output^0) output) i^0$

If given more resources we would use more hidden layers with more nodes per layers in our neural network. We would also train the network on a larger data set of taht consist of multiple images as well as including images containing varying different colors and gradients in the data set. This would expose the network to train and learn on a large breath of images with a wide array of colors.

4 Comparison

The basic agent and the improved agent performed better than expected. But to compare the two, we must have some way to quantify the results. Similar to how we trained and predicted them we used the sum Euclidean distance between the original picture and the 2 agents pixel by pixel. Below is a graph of the total average distance of each agent is from the original in 10000 random samples. As you can see the improved agent is definitely closer to the original image because the distance is smaller between the improved and the original compared to the distance between basic and original. This is not a completely fair comparison between the two because the color of the basics are reduced down to 5 main colors while the colors of the improved have a larger range. However, at the end of the day, we are trying to predict an image that is visually similar to the original. Also from figure 4 and figure 5 we can see that the improved agent's resulting image is smoother and more realistic than basic as the basic agent is limited to 5 colors and there is a hard non-smooth transitions from one color to the another while the imprvoe agent's resulting images blends colors on transitions,

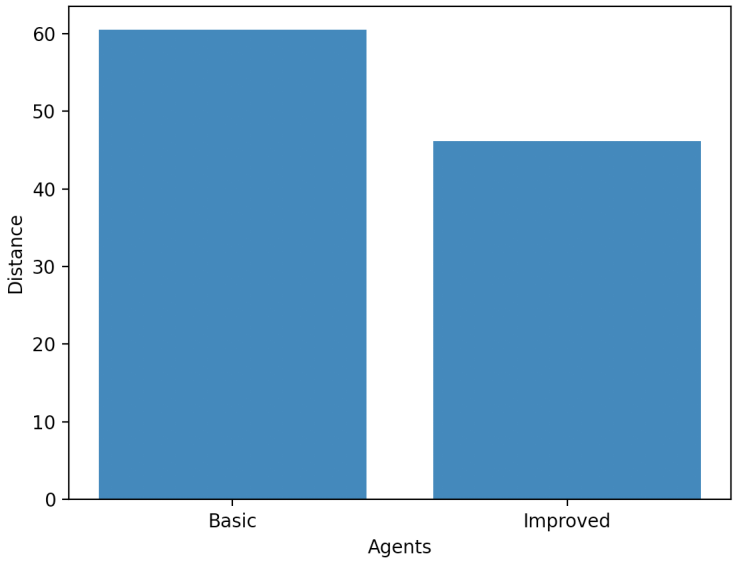

Figure 6: Basic vs NCAA for distance from Original