Intro to Artificial Intelligence

Project 3 Report & Answers - 08/15/2017

Murtala Aliyu / Ofomezie Emelle

Github: github.com/murtalaliyu/AIP3

1.
    a. 256^3 possible color values
    b. 256 possible grays
    c. 256^9 possible gray pixel patches
    d. The situation that arises to me where you'd want to color on a pixel by pixel basis is one that has a picture that mixes a lot of colors all over the picture (like a famous painting for example). You wouldn't be able to generalize to a pixel patch if it's important that every pixel is colored the right color.

2. Since there are less color options to choose from, having the algorithm narrow down to a specific color is a lot easier than determining the exact shade of the color. In other words, if we're sure that the color is a shade of blue, it's a lot less costly to just say that the color is blue than to try and make the algorithm differentiate cyan from sky-blue. A good coloring algorithm would be able to find out whether red or green affects the color of the pixel more strongly than the other, and give you an educated guess as to what the color should be.

3.

a. Generating representative colors was fairly simple I felt. Using any given RGB value I was able to narrow down to a representative color of the rainbow (as well as pink, black, white, and gray) by first ranking the r, g, and b values in order of which is highest, 2nd highest, and lowest. From there I compared the difference between the highest/2nd highest and the difference between the 2nd highest/lowest to determine what the representative color was.
b. For the representative patches, I chose to average the 9 grays and compare the average to representative r, g, b values of the colors Black, Brown, Blue, Green, Yellow, and White. The lower the average was, the darker the color, and considering this is a picture of palm trees it wasn't difficult choosing the representative r, g, b values.
c. The difficulties were in generalizing the 3x3 patches to a specific color as there are a lot of extreme cases that could return you a completely wrong color, but we had to fit a generalization to our algorithm regardless so there was a lot of trial and error.

4. We can use the actual RGB values from colors.csv and input them into any RGB coloring site online that uses the same values. We can compare the color the patch is said to be by our algorithm to the actual color that the color in colors.csv is.

5. In the Zip file

6. It performs well when it comes to guessing the right color but it over generalizes the shade of the exact color you're looking for. We could improve its performance by enhancing the representative colors algorithm and making more representative colors for given r, g, b values but that would be a lot costlier an algorithm. Diverse images would definitely hinder the performance of this algorithm as we didn't use any complex colors.

7. It probably wouldn’t color in very accurate colors, but the shading would remain accurate. For example, we might be getting a tiger that’s colored blue or green because we’ve generalized patches to return a lot of blues or greens for lightly colored patches, but we didn’t need as many yellows.