Assignment 2- Visual Information Retrieval

COMS W4735: Visual Interfaces to Computers

Spring 2015

Sabina Smajlaj

ss3912

Table of Contents

Part 1- Gross Color Matching

Part 1 asked us to find the gross color matching for each of the 40 images. In order to do this, we had to read in the images and get their histograms.

A histogram acts as a graphical representation of the tonal distribution of an image. It does this by plotting the number of pixels for each tonal value, giving us a good idea of the entire tonal distribution at a glance. The left side of the horizontal axis represents the black and dark areas, the middle represents medium grey and the right hand side represents light and pure white areas. Histograms are a useful tool for thresholding since they can useful for analyzing peaks and troughs. This threshold value can then be used for edge detection and image segmentation.

When we are dealing with three-dimensional color spaces like RGB, the image histogram is usually referred to as a "color histogram." Often times, the space is divided into an appropriate number of ranges, often arranged as a regular grid, each containing many similar color values. A color histogram is produced first by discretization of the colors in the image into a number of bins and then counting the number of image pixels in each bin. For a three-dimensional RGB image, we can think of this as creating a three-dimensional array where each index represents a range of pixels.

Since we wanted the most accurate representation of the pixels, I chose the read in the images in their PPM form. This decision was also influenced by a lack of ability to get the histograms of the JPG images through the blackbox methods of OpenCV. Ultimately, reading in the images myself helped me get better results and understand the concepts better. After reading in the images, the next decision revolved around how to place the pixels for each image into bins. Since each image had 89x60 pixels, and our 3D histogram would have RxGxB bins, having 256x256x256 bins would have been too much. Instead, I chose to create a range for each RGB value. This was a range of 32.

Since the images were 8-bit, I partitioned them into 8 ranges, of 32 values each. Hence for R, it's ranges were 0-31, 32-63, 64-95, 96-127, etc. This gave me 8x8x8=512 bins. The next question was how to handle "black pixels". Since many of the images had a black background, counting black pixels would have given me many false similarities. In order to discount for this, I ignored all black pixels. Black pixel was defined as anything that fell into the 0th bin, ie was in the range of 0-31. The total number of black pixels was then subtracted from my value of 2N since we didn't count these pixels.

In order to figure out their degree of similarity, I created a method for the L1 comparison algorithm. For me, the algorithm returning a zero means total similarity, and a one means no similarity. I inverted them for ease of use.

When I run the Color Matching code in the ColorMatching class, I get the following output. This output gives me the names of the 3 most and least similar images for each one of the 40 images. In addition, it gives me the overall 4 most similar and overall 4 least similar images.

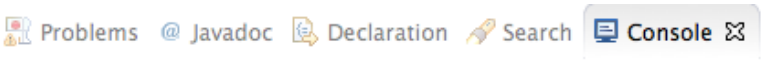
The way the 3 most similar images are organized is in order of similarity. The image displayed first is the most similar, and so on. Likewise, the least similar images are also displayed in that manner; the least similar image is displayed first. So for example:

```
The three most similar images to i01.ppm are:
i10.ppm
i19.ppm
i24.ppm
The three least similar images to i01.ppm are:
i26.ppm
i15.ppm
i12.ppm
```

The 7-tuple for the above would be:

> i01 → i10 → i19 → i24 → i12 → i15 → i26

In order to determine which 4 images were the most similar or the least similar, I took the absolute value of their L1 difference comparisons. For most similar, the closer to zero that their sum was, the more similar the images were. For the least similar, the larger their sum was, the more different the images were. For example, if three images have L1 scores of 0.22, 0.24, and 026. their sum would be 0.72. for another set with L1 scores of 0.51, 0.52, and 0.51, their sum would be 1.54. We can see that the smaller their L1 difference, the more similar the images are to each other, and vice versa.




```
<terminated> ColorMatching [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0_67.jdk/Content
The three most similar images to i01.ppm are:
i10.ppm
i19.ppm
i24.ppm
The three least similar images to i01.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i02.ppm are:
i34.ppm
i21.ppm
i17.ppm
The three least similar images to i02.ppm are:
i26.ppm
i15.ppm
i06.ppm
The three most similar images to i03.ppm are:
i24.ppm
i04.ppm
i08.ppm
The three least similar images to i03.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i04.ppm are:
i03.ppm
i08.ppm
i24.ppm
The three least similar images to i04.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i05.ppm are:
i23.ppm
i28.ppm
i11.ppm
The three least similar images to i05.ppm are:
i26.ppm
i12.ppm
i02.ppm
The three most similar images to i06.ppm are:
i05.ppm
i11.ppm
i28.ppm
The three least similar images to i06.ppm are:
i02.ppm
i12.ppm
i26.ppm
The three most similar images to i07.ppm are:
i09.ppm
i11.ppm
i28.ppm
The three least similar images to i07.ppm are:
i26.ppm
i02.ppm
i12.ppm
The three most similar images to i08.ppm are:
i24.ppm
i03.ppm
i19.ppm
```

Problems Javadoc Declaration Search Console

<terminated> ColorMatching [Java Application] /Library/Java/JavaVirtualMachines/jdk1.7.0

```
i15.ppm
i12.ppm
The three most similar images to i09.ppm are:
i07.ppm
i11.ppm
i28.ppm
The three least similar images to i09.ppm are:
i26.ppm
i15.ppm
i02.ppm
The three most similar images to i10.ppm are:
i01.ppm
i24.ppm
i27.ppm
The three least similar images to i10.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i11.ppm are:
i28.ppm
i27.ppm
i09.ppm
The three least similar images to i11.ppm are:
i26.ppm
i15.ppm
i02.ppm
The three most similar images to i12.ppm are:
i09.ppm
i28.ppm
i27.ppm
The three least similar images to i12.ppm are:
i26.ppm
i15.ppm
i06.ppm
The three most similar images to i13.ppm are:
i10.ppm
i27.ppm
i28.ppm
The three least similar images to i13.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i14.ppm are:
i39.ppm
i09.ppm
i28.ppm
The three least similar images to i14.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i15.ppm are:
i07.ppm
i40.ppm
i11.ppm
The three least similar images to i15.ppm are:
i26.ppm
i12.ppm
i02.ppm
```

Problems @ Javadoc Declaration Search Console

<terminated> ColorMatching [Java Application] /Library/Java/JavaVirtualMachines/j

```
i02.ppm
The three most similar images to i16.ppm are:
i10.ppm
i03.ppm
i27.ppm
The three least similar images to i16.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i17.ppm are:
i25.ppm
i28.ppm
i21.ppm
The three least similar images to i17.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i18.ppm are:
i28.ppm
i27.ppm
i35.ppm
The three least similar images to i18.ppm are:
i15.ppm
i26.ppm
i12.ppm
The three most similar images to i19.ppm are:
i24.ppm
i03.ppm
i08.ppm
The three least similar images to i19.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i20.ppm are:
i28.ppm
i27.ppm
i23.ppm
The three least similar images to i20.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i21.ppm are:
i28.ppm
i27.ppm
i17.ppm
The three least similar images to i21.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i22.ppm are:
i27.ppm
i28.ppm
i01.ppm
The three least similar images to i22.ppm are:
i26.ppm
i15.ppm
i12.ppm
```

Problems @ Javadoc Declaration Search Consol

<terminated> ColorMatching [Java Application] /Library/Java/JavaVirtu

```
The three most similar images to i23.ppm are:
i28.ppm
i27.ppm
i24.ppm
The three least similar images to i23.ppm are:
i26.ppm
i12.ppm
i15.ppm
The three most similar images to i24.ppm are:
i08.ppm
i03.ppm
i19.ppm
The three least similar images to i24.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i25.ppm are:
i28.ppm
i17.ppm
i27.ppm
The three least similar images to i25.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i26.ppm are:
i33.ppm
i28.ppm
i27.ppm
The three least similar images to i26.ppm are:
i15.ppm
i02.ppm
i12.ppm
The three most similar images to i27.ppm are:
i28.ppm
i32.ppm
i24.ppm
The three least similar images to i27.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i28.ppm are:
i27.ppm
i24.ppm
i08.ppm
The three least similar images to i28.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i29.ppm are:
i30.ppm
i10.ppm
i27.ppm
The three least similar images to i29.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i30.ppm are:
i27.ppm
i29.ppm
i28.ppm
```

Problems @ Javadoc Declaration Search Console

<terminated> ColorMatching [Java Application] /Library/Java/JavaVirtual

```
i28.ppm
The three least similar images to i30.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i31.ppm are:
i27.ppm
i28.ppm
i32.ppm
The three least similar images to i31.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i32.ppm are:
i27.ppm
i28.ppm
i24.ppm
The three least similar images to i32.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i33.ppm are:
i28.ppm
i27.ppm
i24.ppm
The three least similar images to i33.ppm are:
i15.ppm
i26.ppm
i12.ppm
The three most similar images to i34.ppm are:
i21.ppm
i28.ppm
i27.ppm
The three least similar images to i34.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i35.ppm are:
i28.ppm
i27.ppm
i18.ppm
The three least similar images to i35.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i36.ppm are:
i04.ppm
i38.ppm
i03.ppm
The three least similar images to i36.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i37.ppm are:
i27.ppm
i28.ppm
i03.ppm
The three least similar images to i37.ppm are:
i26.ppm
i15.ppm
```

```
The three most similar images to i37.ppm are:
i27.ppm
i28.ppm
i03.ppm
The three least similar images to i37.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i38.ppm are:
i03.ppm
i19.ppm
i24.ppm
The three least similar images to i38.ppm are:
i26.ppm
i15.ppm
i12.ppm
The three most similar images to i39.ppm are:
i34.ppm
i28.ppm
i27.ppm
The three least similar images to i39.ppm are:
i26.ppm
i15.ppm
i06.ppm
The three most similar images to i40.ppm are:
i07.ppm
i33.ppm
i09.ppm
The three least similar images to i40.ppm are:
i12.ppm
i26.ppm
i02.ppm
The 4 most similar images out of the whole are:
i24.ppm
i08.ppm
i03.ppm
i19.ppm
The 4 least similar images out of the whole are:
i26.ppm
i33.ppm
i28.ppm
i27.ppm
```

We can see some examples of it working below (the groups are shown vertically from original to most similar to least similar):

i36.jpg

i04.jpg

i38.jpg

i03.jpg

i12.jpg

i15.jpg

i26.jpg

i40.jpg

i07.jpg

i33.jpg

i09.jpg

i02.jpg

i26.jpg

i12.jpg

i10.jpg

i01.jpg

i24.jpg

i27.jpg

i12.jpg

i15.jpg

i26.jpg

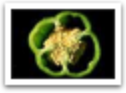
i09.jpg

i07.jpg

i11.jpg

i28.jpg

i27.jpg



i24.jpg

i01.jpg

i05.jpg

i02.jpg

i01.jpg

i23.jpg

i34.jpg

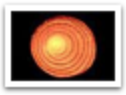
i10.jpg

i28.jpg

i21.jpg

i19.jpg

i11.jpg



i17.jpg

i20.jpg

i02.jpg

i06.jpg

i12.jpg

i12.jpg

i15.jpg

i15.jpg

i26.jpg

i26.jpg

i26.jpg

Part 2- Gross Texture Matching

In this section, we are asked to measure and compare image textures. The texture, or edginess, of an image is measured as the points at which the image brightness changes abruptly, or rather has discontinuities. These points at which the image brightness changes sharply are, in mathematics, organized into a set of curved line segments that we can call "edges."

In order to find these discontinuities, we first create a black and white version of the image. I did this by following Professor Kender's advice and just converting each RGB value to (R+G+B)/3. Then we want to get the Laplacian of the image. This is done by multiplying each pixel's value by either 8, 5, or 3, depending on how many surrounding neighbors it has. Then we subtract the sum of it's surrounding neighbors from that and that number become the new pixel value. Lastly, we can create a 1D histogram from these pixel values.

The range for the histogram values is from -2048 to 2048. This is because we have a input pixel range of 0 to 255, which is 256 values. Then we're comparing it to the 8 (or 5 or 3, but the greatest possible number is 8) neighbors surrounding it. [As a side note, the laplacian value is calculated differently based on where the pixel lies on the grid and how many surrounding neighbors it has: 8, 5, or 3.] With 8 possible neighbors, we have 256*8 possible values. Our max and min can be thought about in terms of having a white pixel surrounded by black pixels or having a black pixel surrounded by white pixels. The Laplacian of the former would be (255*8) - (0*8) = 2040. For the latter case, this would be (0*8) - (255*8) = -2040. This can give us over 4000 bins for the histogram, which is a lot. We can cut this number down by again creating ranges for the bins. One possible value could be having 512 bins. This number is not as large as 4096 but also not too small to give us false positives. I had previously tried having just 32 bins, but that proved to be too small for correct answers. Here, I handled black the same way I did in the previous step.

The rest of the problem was then very similar to Part 1. The output is included below and again follows a similar structure to Part 1.

```
The three most similar images to i01.ppm are:
i08.ppm
i07.ppm
i03.ppm
The three least similar images to i01.ppm are:
i22.ppm
i26.ppm
i34.ppm
The three most similar images to i02.ppm are:
i12.ppm
i06.ppm
```

```
i11.ppm
The three least similar images to i02.ppm are:
i22.ppm
i18.ppm
i29.ppm
The three most similar images to i03.ppm are:
i07.ppm
i04.ppm
i09.ppm
The three least similar images to i03.ppm are:
i26.ppm
i33.ppm
i22.ppm
The three most similar images to i04.ppm are:
i03.ppm
i08.ppm
i07.ppm
The three least similar images to i04.ppm are:
i26.ppm
i33.ppm
i22.ppm
The three most similar images to i05.ppm are:
i11.ppm
i12.ppm
i31.ppm
The three least similar images to i05.ppm are:
i22.ppm
i34.ppm
i18.ppm
The three most similar images to i06.ppm are:
i02.ppm
i26.ppm
i11.ppm
The three least similar images to i06.ppm are:
i22.ppm
i18.ppm
i17.ppm
The three most similar images to i07.ppm are:
i03.ppm
i31.ppm
i04.ppm
The three least similar images to i07.ppm are:
i22.ppm
i26.ppm
i34.ppm
The three most similar images to i08.ppm are:
i01.ppm
i04.ppm
i07.ppm
The three least similar images to i08.ppm are:
```

```
i22.ppm
i26.ppm
i34.ppm
The three most similar images to i09.ppm are:
i03.ppm
i10.ppm
i07.ppm
The three least similar images to i09.ppm are:
i26.ppm
i22.ppm
i33.ppm
The three most similar images to i10.ppm are:
i09.ppm
i16.ppm
i03.ppm
The three least similar images to i10.ppm are:
i26.ppm
i33.ppm
i06.ppm
The three most similar images to i11.ppm are:
i05.ppm
i12.ppm
i31.ppm
The three least similar images to i11.ppm are:
i22.ppm
i34.ppm
i18.ppm
The three most similar images to i12.ppm are:
i05.ppm
i11.ppm
i02.ppm
The three least similar images to i12.ppm are:
i22.ppm
i34.ppm
i18.ppm
The three most similar images to i13.ppm are:
i14.ppm
i16.ppm
i39.ppm
The three least similar images to i13.ppm are:
i26.ppm
i22.ppm
i33.ppm
The three most similar images to i14.ppm are:
i13.ppm
i39.ppm
i16.ppm
The three least similar images to i14.ppm are:
i26.ppm
i34.ppm
```

```
i33.ppm
The three most similar images to i15.ppm are:
i12.ppm
i13.ppm
i08.ppm
The three least similar images to i15.ppm are:
i22.ppm
i34.ppm
i29.ppm
The three most similar images to i16.ppm are:
i10.ppm
i03.ppm
i04.ppm
The three least similar images to i16.ppm are:
i26.ppm
i33.ppm
i06.ppm
The three most similar images to i17.ppm are:
i38.ppm
i21.ppm
i19.ppm
The three least similar images to i17.ppm are:
i26.ppm
i06.ppm
i02.ppm
The three most similar images to i18.ppm are:
i20.ppm
i35.ppm
i19.ppm
The three least similar images to i18.ppm are:
i26.ppm
i06.ppm
i33.ppm
The three most similar images to i19.ppm are:
i04.ppm
i35.ppm
i03.ppm
The three least similar images to i19.ppm are:
i26.ppm
i06.ppm
i33.ppm
The three most similar images to i20.ppm are:
i35.ppm
i18.ppm
i39.ppm
The three least similar images to i20.ppm are:
i26.ppm
i06.ppm
i33.ppm
The three most similar images to i21.ppm are:
```

```
i25.ppm
i17.ppm
i38.ppm
The three least similar images to i21.ppm are:
i26.ppm
i06.ppm
i12.ppm
The three most similar images to i22.ppm are:
i17.ppm
i36.ppm
i38.ppm
The three least similar images to i22.ppm are:
i26.ppm
i06.ppm
i12.ppm
The three most similar images to i23.ppm are:
i07.ppm
i04.ppm
i08.ppm
The three least similar images to i23.ppm are:
i22.ppm
i34.ppm
i26.ppm
The three most similar images to i24.ppm are:
i09.ppm
i03.ppm
i07.ppm
The three least similar images to i24.ppm are:
i26.ppm
i22.ppm
i33.ppm
The three most similar images to i25.ppm are:
i36.ppm
i40.ppm
i08.ppm
The three least similar images to i25.ppm are:
i26.ppm
i06.ppm
i22.ppm
The three most similar images to i26.ppm are:
i06.ppm
i02.ppm
i12.ppm
The three least similar images to i26.ppm are:
i22.ppm
i18.ppm
i17.ppm
The three most similar images to i27.ppm are:
i28.ppm
i32.ppm
```

```
i11.ppm
The three least similar images to i27.ppm are:
i22.ppm
i18.ppm
i29.ppm
The three most similar images to i28.ppm are:
i27.ppm
i24.ppm
i32.ppm
The three least similar images to i28.ppm are:
i18.ppm
i22.ppm
i29.ppm
The three most similar images to i29.ppm are:
i30.ppm
i39.ppm
i38.ppm
The three least similar images to i29.ppm are:
i26.ppm
i06.ppm
i02.ppm
The three most similar images to i30.ppm are:
i25.ppm
i36.ppm
i29.ppm
The three least similar images to i30.ppm are:
i26.ppm
i06.ppm
i11.ppm
The three most similar images to i31.ppm are:
i07.ppm
i05.ppm
i09.ppm
The three least similar images to i31.ppm are:
i22.ppm
i34.ppm
i33.ppm
The three most similar images to i32.ppm are:
i05.ppm
i11.ppm
i31.ppm
The three least similar images to i32.ppm are:
i22.ppm
i29.ppm
i18.ppm
The three most similar images to i33.ppm are:
i37.ppm
i40.ppm
i34.ppm
The three least similar images to i33.ppm are:
```

```
i18.ppm
i22.ppm
i29.ppm
The three most similar images to i34.ppm are:
i37.ppm
i36.ppm
i33.ppm
The three least similar images to i34.ppm are:
i26.ppm
i06.ppm
i18.ppm
The three most similar images to i35.ppm are:
i19.ppm
i20.ppm
i04.ppm
The three least similar images to i35.ppm are:
i26.ppm
i06.ppm
i22.ppm
The three most similar images to i36.ppm are:
i25.ppm
i40.ppm
i37.ppm
The three least similar images to i36.ppm are:
i26.ppm
i06.ppm
i18.ppm
The three most similar images to i37.ppm are:
i40.ppm
i36.ppm
i33.ppm
The three least similar images to i37.ppm are:
i18.ppm
i22.ppm
i26.ppm
The three most similar images to i38.ppm are:
i17.ppm
i16.ppm
i39.ppm
The three least similar images to i38.ppm are:
i26.ppm
i06.ppm
i33.ppm
The three most similar images to i39.ppm are:
i14.ppm
i16.ppm
i13.ppm
The three least similar images to i39.ppm are:
i26.ppm
i33.ppm
```

```
i06.ppm
The three most similar images to i40.ppm are:
i37.ppm
i25.ppm
i36.ppm
The three least similar images to i40.ppm are:
i18.ppm
i22.ppm
i29.ppm
The 4 most similar images out of the whole are:
i03.ppm
i07.ppm
i04.ppm
i09.ppm
The 4 least similar images out of the whole are:
i22.ppm
i17.ppm
i36.ppm
i38.ppm
```

i27.jpg

i39.jpg

i10.jpg

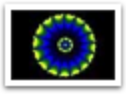
i07.jpg

i28.jpg

i14.jpg

i09.jpg

i03.jpg



i32.jpg

i16.jpg

i16.jpg

i31.jpg

i11.jpg

i13.jpg

i03.jpg

i04.jpg

i29.jpg

i06.jpg

i06.jpg

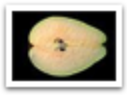
i34.jpg

i18.jpg

i33.jpg

i33.jpg

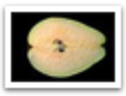
i26.jpg



i22.jpg

i26.jpg

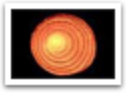
i26.jpg



i22.jpg

i02.jpg

i01.jpg



i17.jpg

i19.jpg

i12.jpg

i08.jpg

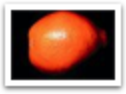
i38.jpg

i04.jpg

i06.jpg

i07.jpg

i21.jpg



i35.jpg

i11.jpg

i03.jpg

i19.jpg

i03.jpg

i29.jpg

i34.jpg

i02.jpg

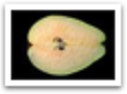
i33.jpg

i18.jpg

i26.jpg

i06.jpg

i06.jpg



i22.jpg

i22.jpg

i26.jpg

i26.jpg

Part 3- Combine Similarities and Cluster

One important aspect of this part was defining a good value for r. One value I chose was having r be the average of the color similarity and the texture similarity. I decided on this because if we're going to use both similarities, we can try to take both into account the same amount.

Complete Link

For Complete link, we want to define nearness as the distance between the farthest image in one cluster to the furthest image in the other cluster.

Using r as the average of the color similarity and the texture similarity, I get the following 7 clusters:

```
String: i01.ppm
String: i04.ppm
String: i06.ppm
String: i08.ppm
String: i10.ppm
String: i12.ppm
String: i14.ppm
String: i16.ppm
String: i18.ppm
String: i20.ppm
String: i22.ppm
String: i24.ppm
String: i28.ppm
String: i30.ppm
String: i32.ppm
String: i34.ppm
String: i36.ppm
String: i38.ppm
String: i40.ppm
This is the end of the cluster.
String: i03.ppm
String: i26.ppm
String: i05.ppm
String: i09.ppm
String: i13.ppm
String: i17.ppm
String: i21.ppm
String: i25.ppm
String: i33.ppm
String: i37.ppm
This is the end of the cluster.
```

```
String: i07.ppm
String: i02.ppm
String: i29.ppm
String: i11.ppm
String: i19.ppm
String: i27.ppm
String: i35.ppm
This is the end of the cluster.
String: i15.ppm
This is the end of the cluster.
String: i23.ppm
This is the end of the cluster.
String: i31.ppm
This is the end of the cluster.
String: i39.ppm
This is the end of the cluster.
```

The way the clusters are defined is that the links are listed vertically, ie in the above example, file i01.ppm is added to the cluster first, followed by i04.ppm.

Another value of r that we can use that might work better is using only one fourth of the texture since texture can be harder to define and measure than color, and using three fourths of color. Using r as the value of (0.25*T) + (0.75*C), I get the same clusters, leading me to believe that both values of r are good values:

```
String: i01.ppm
String: i04.ppm
String: i06.ppm
String: i08.ppm
String: i10.ppm
String: i12.ppm
String: i14.ppm
String: i16.ppm
String: i18.ppm
String: i20.ppm
String: i22.ppm
String: i24.ppm
String: i28.ppm
String: i30.ppm
String: i32.ppm
String: i34.ppm
String: i36.ppm
String: i38.ppm
String: i40.ppm
This is the end of the cluster.
String: i03.ppm
String: i26.ppm
String: i05.ppm
String: i09.ppm
```

```
String: i13.ppm
String: i17.ppm
String: i21.ppm
String: i25.ppm
String: i33.ppm
String: i37.ppm
This is the end of the cluster.
String: i07.ppm
String: i02.ppm
String: i29.ppm
String: i11.ppm
String: i19.ppm
String: i27.ppm
String: i35.ppm
This is the end of the cluster.
String: i15.ppm
This is the end of the cluster.
String: i23.ppm
This is the end of the cluster.
String: i31.ppm
This is the end of the cluster.
String: i39.ppm
This is the end of the cluster.
```

Single Link

Single Link defines nearness as the distance between the closest image in one cluster and the closest image in the other cluster.

With single link and a value of r of (0.25*T) + (0.75*C), I get the following 7 clusters:

```
String: i01.ppm
String: i04.ppm
String: i08.ppm
String: i10.ppm
String: i12.ppm
String: i16.ppm
String: i14.ppm
String: i18.ppm
String: i06.ppm
String: i20.ppm
String: i22.ppm
String: i24.ppm
String: i36.ppm
String: i28.ppm
String: i32.ppm
String: i30.ppm
String: i34.ppm
String: i38.ppm
```

```
String: i40.ppm
This is the end of the cluster.
String: i03.ppm
String: i26.ppm
String: i09.ppm
String: i05.ppm
String: i17.ppm
String: i13.ppm
String: i25.ppm
String: i21.ppm
String: i33.ppm
String: i37.ppm
This is the end of the cluster.
String: i07.ppm
String: i19.ppm
String: i02.ppm
String: i29.ppm
String: i11.ppm
String: i35.ppm
String: i27.ppm
This is the end of the cluster.
String: i15.ppm
This is the end of the cluster.
String: i23.ppm
This is the end of the cluster.
String: i31.ppm
This is the end of the cluster.
String: i39.ppm
This is the end of the cluster.
```

Part 4- Creative Step

Note: I mistakenly switched the labels for Image #35 and Image #36 when presenting them to the users, so in the results below, they are switched.

For this part, I had 3 friends complete parts 1 and 2 using cutouts of the images. Their results are displayed below.

To determine the performance of my system, I can compare it to the results from the user subjects. We can assume that the users are pretty good at determining the color and texture of an image. Hence, for each image when we're looking at the 3 most and 3 least similar images, we can add one point for each time at least one of the user's answers for the original image is in the six images my system picked for that original image. We can do the same thing for the 4 most and least similar as well.

Running this on my system, I get an average of 16 points for Part 1, giving me a correct percentage of 40%.

For Part 2, I got an average of 13 points, which gave me a correct percentage of 32.5%.

Although my system wasn't terrible, it could certainly use improvements. Another reason for the low percentages could have been due to the users; 2 out of 3 of the users were my young siblings who probably didn't have a great grasp of texture and even color to an extent.

Player1						
Color Matching						
Original	**Most similar**					**Least similar**
8	7	6	15	27	38	31
38	35	19	24	5	26	12
12	14	2	39	31	37	21
21	23	40	33	31	3	13
13	34	22	14	32	5	18
18	36	33	17	31	37	30
30	29	22	13	6	37	11
11	23	6	5	31	16	32
32	27	24	30	18	6	5
5	6	11	23	13	31	29
6	5	15	23	31	22	32
24	3	4	19	31	18	23
3	16	4	38	36	31	27
36	18	23	33	31	37	30
31	32	27	7	26	14	10
10	1	3	16	11	31	28
28	40	33	25	31	32	16
26	36	18	17	31	37	13
37	16	4	32	23	2	6
15	36	8	17	32	31	37
19	38	16	10	31	6	32
27	32	1	30	26	18	20
20	34	17	25	31	32	6
17	20	18	36	31	30	37
25	28	17	20	37	27	31
29	38	22	30	37	31	6
35	22	38	29	31	6	28
14	13	12	2	31	6	37
9	3	4	1	31	37	6
39	20	22	14	31	36	23
33	36	28	25	30	32	37
2	17	34	20	30	32	27
1	3	9	4	31	6	2
4	10	3	1	6	11	31
7	8	40	15	31	37	32
40	7	8	21	31	27	19
23	6	21	5	31	32	37
16	10	1	4	6	31	29
22	29	2	34	31	6	32
34	20	22	2	37	31	6

Player1						
Texture Matching						
Original	**Most similar**					**Least similar**
36	35	37	30	11	3	1
27	31	17	32	26	1	29
19	32	18	31	12	15	9
7	16	6	8	28	22	29
28	33	24	25	2	29	1
1	11	10	8	15	22	29
29	36	22	37	13	11	1
11	10	1	7	36	26	29
26	30	35	37	31	15	9
15	6	16	2	26	29	31
18	32	20	19	15	9	3
3	4	9	13	32	31	29
4	3	9	13	32	31	29
23	24	18	32	29	1	9
5	6	7	16	29	31	30
33	27	9	28	29	31	35
14	5	25	12	29	31	26
40	37	38	35	1	27	15
31	32	17	27	29	15	36
34	23	22	28	1	9	31
2	5	14	25	29	31	26
6	7	8	16	28	22	29
8	6	7	16	28	22	29
9	3	4	13	28	22	29
10	11	1	7	28	22	29
12	15	5	2	26	29	31
13	3	4	9	26	29	31
16	6	2	15	26	29	31
17	32	31	27	26	29	31
19	20	32	18	15	9	3
20	32	18	19	15	9	3
21	19	20	32	3	4	9
22	23	34	28	1	9	31
24	33	28	25	1	2	29
25	14	5	2	26	29	31
30	37	35	36	3	1	11
32	19	18	31	9	12	15
35	30	26	37	9	31	15
37	26	30	35	6	16	15
38	18	20	32	9	3	15
39	35	30	36	1	3	11

Player2						
Color Matching						
Original	**Most similar**					**Least similar**
1	3	16	4	31	5	2
2	17	34	34	30	32	27
3	16	4	38	36	31	27
4	10	3	1	6	11	31
5	6	11	23	13	31	29
6	5	15	23	31	22	32
7	8	40	15	31	37	32
8	40	5	15	27	38	31
9	3	4	1	31	37	5
10	1	3	16	11	31	28
11	23	15	5	31	16	32
12	14	2	39	31	37	21
13	34	22	14	32	5	18
14	13	12	2	31	5	37
15	36	8	17	32	31	37
16	10	1	4	6	31	29
17	34	18	36	31	30	37
18	36	33	17	32	37	30
19	38	5	10	31	5	32
20	34	17	25	32	32	15
21	23	40	33	31	3	13
22	29	2	34	31	15	32
23	6	21	5	32	32	37
24	3	4	19	31	18	23
25	28	17	20	37	27	31
26	36	18	17	31	37	13
27	32	1	30	26	18	20
28	7	33	25	31	32	16
29	38	22	30	37	31	6
30	29	22	13	6	37	11
31	32	27	7	26	14	10
32	27	24	30	18	15	5
33	36	28	25	30	32	37
34	34	22	2	37	31	6
35	22	38	29	31	6	28
36	18	23	33	31	37	30
37	16	4	32	23	2	6
38	35	19	24	6	26	12
39	34	22	14	31	36	23
40	7	8	21	31	27	19

Player2						
Texture Matching						
Original	**Most similar**					**Least similar**
1	11	10	8	15	22	29
2	5	14	25	29	31	26
3	4	9	13	32	36	36
4	3	4	13	32	36	29
5	6	7	6	29	31	30
6	7	8	16	28	22	29
7	16	6	8	28	36	36
8	16	7	6	33	22	36
9	3	9	13	33	22	29
10	11	1	7	33	36	29
11	10	11	7	36	26	36
12	16	5	2	26	29	31
13	3	4	9	26	29	31
14	5	25	12	29	31	26
15	16	16	2	30	29	31
16	6	2	15	26	36	31
17	32	31	27	26	29	31
18	32	20	19	15	9	3
19	32	18	31	12	15	9
19	20	32	32	15	9	4
20	32	32	19	15	9	3
21	19	20	32	3	4	9
22	23	34	28	1	9	31
23	24	18	32	29	1	9
24	33	28	25	1	2	29
25	14	5	2	26	29	31
26	37	35	37	31	15	9
27	31	17	32	26	1	29
28	33	24	25	2	29	1
29	36	22	37	13	11	1
30	37	35	36	3	1	11
31	19	17	27	29	15	36
32	19	18	31	9	12	15
33	27	9	28	29	31	35
34	23	22	28	1	4	31
35	30	26	37	9	31	15
36	35	37	30	11	3	1
37	26	30	35	6	16	15
38	18	20	32	9	3	15
40	37	18	35	1	27	15

Player3						
Color Matching						
Original	Most similar					Least similar
10	1	3	16	11	31	28
24	3	4	19	31	18	23
9	16	4	1	32	37	6
1	3	16	4	31	6	2
6	5	15	23	32	22	32
5	5	11	23	13	31	29
23	6	21	5	31	32	37
8	7	6	15	27	38	31
40	7	8	21	31	27	19
7	8	40	15	31	37	32
3	16	4	38	36	31	27
4	10	3	1	6	11	31
16	1	1	4	15	31	29
14	13	12	2	31	6	37
12	14	2	39	31	37	21
37	16	4	32	23	2	6
2	17	34	20	30	32	27
36	18	23	33	31	37	30
17	20	18	36	32	30	37
39	34	22	14	31	36	23
34	20	22	2	37	31	5
35	22	38	29	32	6	28
21	23	40	33	31	3	13
11	23	6	5	31	16	32
32	27	24	30	18	5	5
25	28	17	20	37	27	31
30	29	22	13	6	37	11
22	29	2	34	31	6	32
31	31	27	7	26	14	10
27	32	1	30	26	18	20
13	34	22	14	32	5	18
20	34	17	25	31	32	6
38	35	19	24	5	26	12
18	36	33	17	31	37	30
26	18	18	17	31	37	13
15	36	8	17	32	31	37
33	18	28	25	30	32	37
19	38	16	10	31	6	32
29	38	22	30	37	31	6
28	40	33	25	31	32	16

Player3						
Texture Matching						
Original	**Most similar**					**Least similar**
4	3	9	13	32	31	29
9	9	4	3	28	22	29
13	3	4	9	26	29	31
3	4	9	13	32	31	29
14	5	25	12	29	31	26
2	5	14	25	29	31	26
15	6	16	2	26	29	31
5	16	7	16	29	31	30
8	6	7	16	28	22	29
16	6	2	15	26	29	31
6	7	8	16	28	22	29
11	10	1	7	36	26	29
1	11	10	8	15	22	29
10	11	1	7	28	22	29
25	14	5	2	26	29	31
12	15	5	2	26	29	31
7	16	6	8	28	22	29
38	18	20	32	9	3	15
21	19	20	32	3	4	9
32	19	18	31	9	12	15
19	20	32	18	15	9	9
34	23	22	28	1	9	31
22	23	34	28	1	9	31
23	24	18	32	29	1	9
37	26	30	35	6	16	15
33	27	4	28	29	31	35
26	30	35	40	31	15	9
35	30	26	40	9	31	15
27	31	17	32	26	1	29
19	32	18	31	12	15	9
18	19	20	19	15	9	3
31	32	17	27	29	15	36
17	32	31	27	26	29	31
20	19	18	19	15	9	4
28	33	24	25	2	29	1
24	33	28	25	1	2	29
36	35	37	30	11	4	1
39	37	30	36	1	3	11
29	36	22	37	13	11	1
40	37	38	35	1	27	15
30	37	35	36	3	1	11