Virtual Image Recognition

Federick Gonzalez fag2113

March 11, 2020

1 Introduction

The goal of this program is essentially to write and evaluate algorithms which decide the similarities between images on the base of color, texture and shape, and additionally to organize this information in a readable manner. A sequence of 40 images of assorted fruits, vegetables, and other confounding objects all of size $60 \cdot 89$ pixels were analyzed and compared, the results outputted as separate HTML pages for each sort of comparison. Finally, the success of each comparison was measured with relation to a *Crowd.txt* file and a *MyPreferences.txt* file, containing the chosen similar images of both a previous course and the programmer respectively. These were used to fine tune the program to attempt as accurate an evaluation as possible.

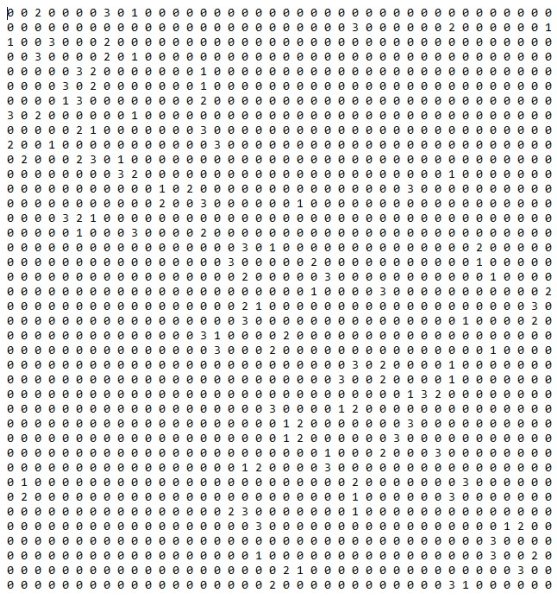

Figure 1: The data in *MyPreferences.txt*

2 Color Distance

2.1 Computation

In order to compare two images, each image was analyzed by creating a 3-Dimensional color histogram, which in Python is essentially a 3-Dimensional Numpy array. The values in this array are pixel values corresponding to color intensities, which each sum up to $60 \times 89 = 5340$ pixels per histogram.

Now that each individual image has been analyzed, comparisons between each image must be made. This is achieved by taking the normalized L1 distance between the separate histograms, using the formula:e

$$L1(Image1, Image2) = \sum_{(r,g,b)} \frac{|Image1(r,g,b) - Image2(r,g,b)|}{2 \cdot rows \cdot cols}$$

Using this formula, each of the 40 images were compared, and each of the 40 images were given 3 images that were deemed most similar to, i.e. three images where $L1(CurrentImage, SimilarImage)$ was closest to 0. These similar images were then checked against Crowd.txt and the system was then fine tuned. In order to test system accuracy, the score of each image was calculated using the following formula:

$$Score(q) = Crowd(q, t_1) + Crowd(q, t_2) + Crowd(q, t_3) \tag{1}$$

To ensure analysis could be completed within an adequate amount of time, the amount of bins for each Red (R), Green (G), or Blue (B) channel in the 3-Dimensional histogram could not be 255. The Number of bins were severely reduced, beginning at $[2, 3, 2]$ for R, G, and B respectively. These values were slowly raised and altered, keeping in mind that G seems to be the color with the most visible differences, B with the least, and R somewhere in the middle.

By raising G, then R, then B at different intervals and checking the results against Crowd.txt, the most accurate value combination seemed to be bins of sizes $[5, 8, 4]$. This conclusion was reached because the total score (The summation of eq. 1 for all images) reached 5372 at this point, the maximum total score for all tested values. Other tested values include $[128, 128, 128]$, $[64, 128, 64]$, $[64, 128, 32]$, $[128, 256, 64]$, and $[8, 16, 6]$, among others.

2.2 Visualization and Performance Evaluation

1) As can be seen in the following sequences of images, for each of the 40 images utilized for this assignment (q), the three most similar images are listed beside it (t), along with their Crowd.txt value ($Crowd(q, t)$). At the end of each row there is listed a score ($Score(q)$). The total score for the Color Distance comparison was 5372.

2) For each row, the number of images selected by the algorithm that were also in turn selected in *MyPreferences.txt* ranged from 0 to 3. In total, the

Figure 2: Image Comparison using Color

images in common between the three best targets for all 40 images and the chosen targets for all images was, in fact, 40.

3) For crowd based accuracy, it seems the first quarter of images seemed to be the most accurate in terms of color. More specifically, images number 08 and 01 seem to be the most accurate out of all 40. This is likely because these two images have very vibrant green coloring, and a large range of green coloring, allowing them to accurately compare to other items that are also of a similar vibrant color scheme with a similar color gradation.

The least accurate area, however, seems to be the third quarter of images, though the least accurate image happens to be 31, as the algorithm deems it as green while it is visibly of a purplish nature. The third quarter of data is

likely the least accurate because there are a variety of colors in almost every image, and varying presence of the black background that could influence color perception. For example, the tomato in picture 23 is clearly red, but it is hard to compare that red to the other red colors in images 5, 6, or 7, due to the presence of a largely visible black background.

The system perfectly matched the personal preferences laid out in *MyPreferences.txt* for the first image only, though it came close in many other cases, such as image 3, 4, and 6.

3 Texture Distance

3.1 Computation

Using the methods native to the OpenCV2 Library, each of the images was first converted to greyscale, then into a Laplacian image, which finds the local changes within an image. Out of this Laplacian image, a one dimensional histogram of values was formed, an as in the Color Distance section, the normalized L1 distance was taken between the separate histograms.

Using this formula, each of the 40 images were compared, and each of the 40 images were given 3 images that were deemed most similar to, i.e. three images where $L1(CurrentImage, SimilarImage)$ was closest to 0. These similar images were then checked against Crowd.txt and the system was then fine tuned. In order to test system accuracy, the score of each image was calculated using the forumla (eq. 1).

The Laplacian transformation changes the possible image values from essentially 0 to ± 5430, therefore the range was chosen to be -5400 to 5400, to make calculation easier. To ensure that accurate comparisons were made, 108001 bins are quite large. However, it was deemed that for the Laplacian images, since the histogram has only one channel, the bin number could be quite large.

After extensive amounts of testing, the value of 10000 bins was settled on, and this is because this seemed the the most accurate value based on the total score of the image. Multiple values were tested on the range from 1000 to 10000 in increments of first 1000, then 500, to see if any other values provided a larger total score, but they did not seem to.

3.1.1 Visualization and Performance Evaluation

1) As can be seen in the following sequences of images, for each of the 40 images utilized for this assignment (q), the three most similar images are listed beside it (t), along with their Crowd.txt value ($Crowd(q,t)$). At the end of each row there is listed a score ($Score(q)$). The total score for the Texture Distance comparison was 2159.

2) For each row, the number of images selected by the algorithm that were also in turn selected in *MyPreferences.txt* ranged from 0 to 3. In total, the images in common between the three best targets for all 40 images and the chosen targets for all images was, in fact, 18.

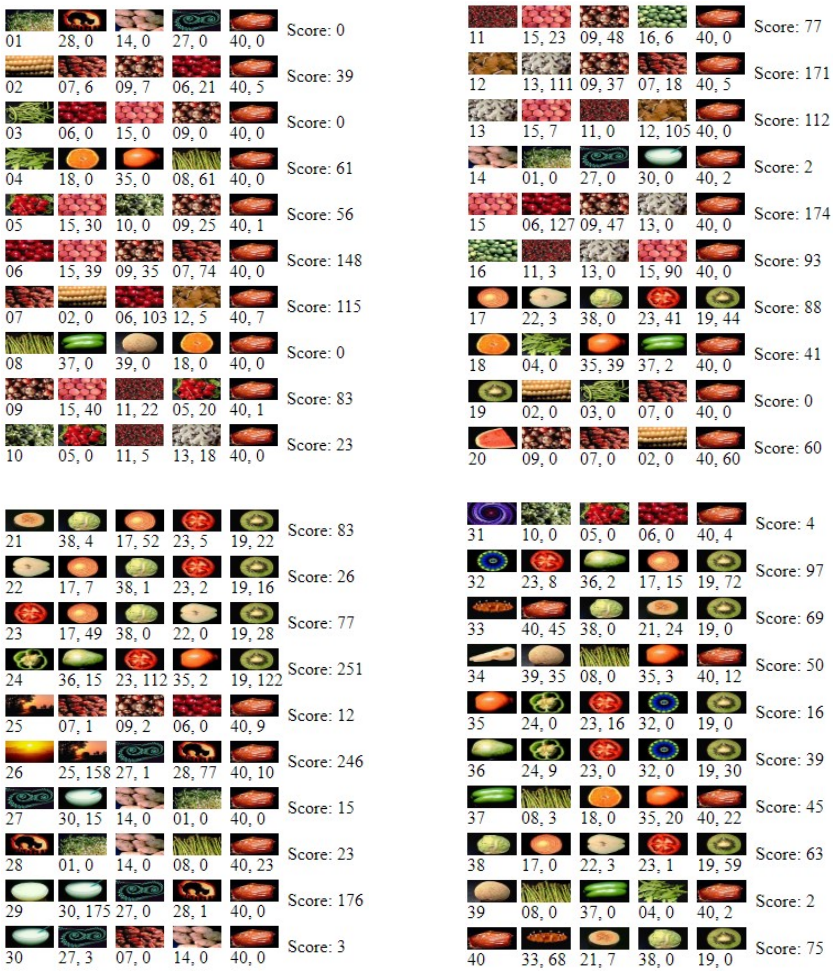

Figure 3: Image Comparison using Texture

3) The image with the largest score here seems to be image 29, with a score of 176. Conversely, the images scored the lowest are 1, 3, 8, and 19, all with a score of 0. This is likely because 1, 3, and 8 are all strangely shaped long green objects oriented in different direction, and the texture algorithm doesn't account for local differences in the same way that humans view texture, this algorithm has no concept of object permanence. 19 is also a strange texture, it is known to be not that different from something like an orange or a tomato, but because the texture detection is based mainly off of small color differences, the black ring of seeds in the center makes it seem edgier than it actually is. It seems that also, human recognition of texture is less important than human recognition of color, as the score is significantly lower than that of color alone.

4 Shape Distance

4.1 Computation

Using the methods native to the OpenCV2 Library, each of the images was converted to grayscale, in order to make differences between black and not black easier to calculate. Rather than making another 3-Dimensional histogram if color and finding a field in which color values could be considered black, as a grayscale image the threshold need only be a single value. A 1 dimensional histogram was made, originally using 255, or a purely black pixel, as the threshold value. Instead of the normalized distance, Normalized overlap was used to calculate the similarity between pixels, the formula for which is as follows:

$$NormOverlap(Image1, Image2) = \sum_{(x,y)} \frac{|Image(N1(x,y) \neq ImageN2(x,y)|}{rows \cdot cols}$$

Using this formula, each of the 40 images were compared, and each of the 40 images were given 3 images that were deemed most similar, i.e. where $NormOverlap(CurrentImage, SimilarImage)$ was closest to 0. These similar images were then checked against Crowd.txt and the system was fine tuned based on the result. In order to test system accuracy, the score of each image was calculated using the formula (1).

Using a grayscale image has the possible values $[0, 255]$, so to ensure accurate comparisons were made, a mask was first made that filtered out all of the black values. Then, within that mask, the number of bins was set to 64, as that seemed to be to most accurate value after much testing.

4.2 Visualization and Performance Evaluation

1) As can be seen in the following sequences of images, for each of the 40 images utilized for this assignment (q), the three most similar images are listed beside it (t), along with their Crowd.txt value ($Crowd(q,t)$). At the end of each row there is listed a score ($Score(q)$). The total score for the Texture Distance comparison was 1416.

2) For each row, the number of images selected by the algorithm that were also in turn selected in *MyPreferences.txt* ranged from 0 to 3. In total, the images in common between the three best targets for all 40 images and the chosen targets for all images was, in fact, 10.

3) The image with the highest score seems to be image 05, with a score of 201, whereas the image with the lowest score, 22, has a score of 0. None of these value ranges are as high as they are in Color Distance, including the total score of all of the Shape Differences. This is likely due to the fact that when people chose preferences, texture was weighted higher than shape, therefore color is weighted much higher than shape.

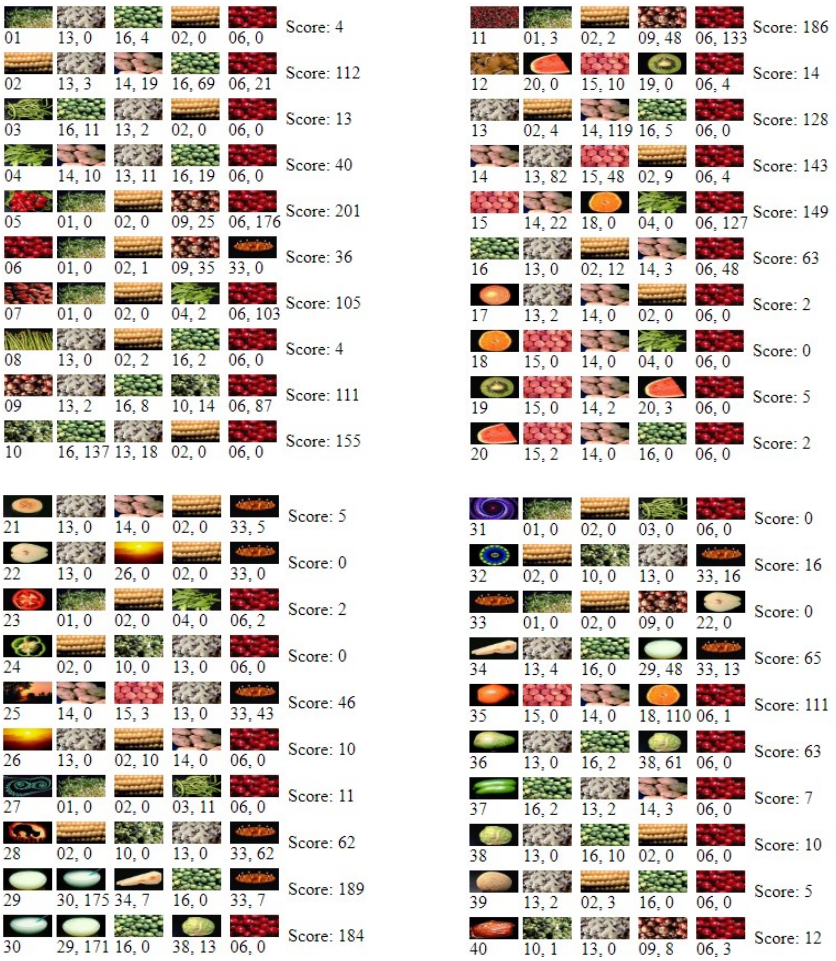

Figure 4: Image Comparison using Shape

5 Overall Distance

The color, shape, and text differences were combined using the formula $P = a \cdot C + b \cdot T + c \cdot S$, where $a + b + c = 1$. The values chosen for a, b, and c were 0.5, 0.45., and 0.05 respectively. While looking at the other comparison methods, it was determined that color was the most useful, shape was somewhere in the middle. Due to this, I began at $[a, b, c] = [0.6, 0.35, 0.05]$, and I manipulated values until I arrived at the highest score value, which was higher than just using shape alone, with a value of 5530.

1) As can be seen in the following sequences of images, for each of the 40 images utilized for this assignment (q), the three most similar images are listed beside it (t), along with their Crowd.txt value ($Crowd(q, t)$). At the end of each

Figure 5: Image Comparison using all three

row there is listed a score ($Score(q)$). The total score for the Texture Distance comparison was 5530.

2) For each row, the number of images selected by the algorithm that were also in turn selected in *MyPreferences.txt* ranged from 0 to 3. In total, the images in common between the three best targets for all 40 images and the chosen targets for all images was, in fact, 39.

3) The image with the highest score seems to be image 08, while the lowest score seems to be image 31. On average, all of the scores are larger than with only color, shape, or texture alone, likely because all three of these are used when visually comparing two objects as a human.

6 Crowd Versus User

6.1

a) The very best possible crowd score is 9853, given all the differences of opinion in Crowd.txt.
b) As a percentage, the final system was 56.12% of this possible score from *Crowd.txt*.
c) As a percentage, the final system was 32.5% of the possible score from $MyPreferences.txt$