Jeremy Woods  
CSS 584  
Assignment 2

**How to run program**

This program has 2 executable parts. One being readImage.exe and the other being CBIR.exe. I have created a batch file that has the ability to run both of these with one click. If you go into the Assignment 2 folder and click on launcher.bat both parts of the program will run and that is all that is need. If you do not wish to do it that way, first launch readImage.exe and then run CBIR.exe. It is done in this order so that readImage.exe can first create the files that CBIR.exe will need. **In order for maximum user friendliness, (i.e. full picture shown in buttons) the program should be run in a maximum size window.**

**Step-by-step use of System**

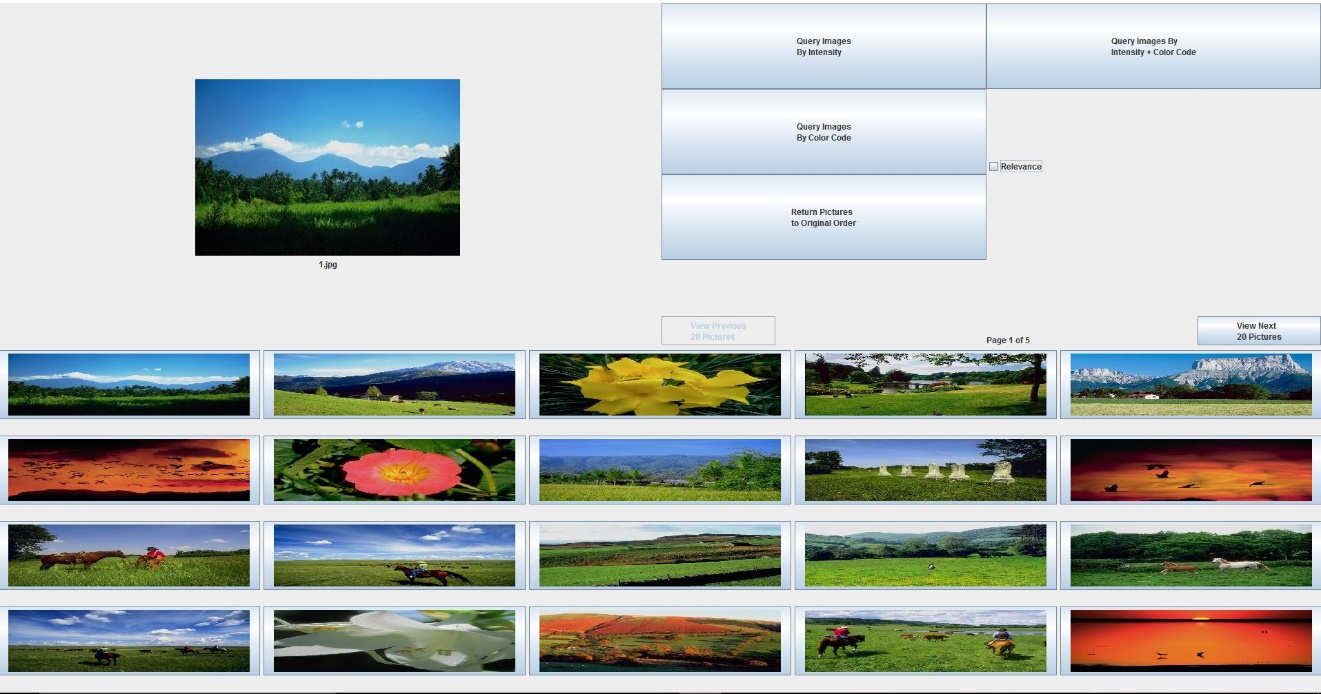
When the system first comes up there will be an empty panel to the upper left, six buttons to the upper right, a checkbox for relevance, and at most 20 pictures on the bottom panel. Here you must either select a picture from the bottom panel or page through the pictures with the next and previous buttons. Pressing any of the query buttons will cause a message to appear in the upper left panel prompting the user to pick a picture for the query. If you select the “Return Pictures to Original Order” button nothing will happen because nothing has been queried yet and as a result the pictures are still in their original order.

After a picture is selected, it will appear in the upper left panel with its name under it. Now you can press any of the query buttons in order to sort all the pictures based on that query. If you select the “Query Images by Intensity” button, the pictures will be sorted in the order or pictures closest to the intensity of the selected picture. If you select the “Query Images by ColorCode” button, the pictures will be sorted in the order or pictures closest to the colorcode of the selected picture. If you select the “Query Images by Intensity + ColorCode” button, the pictures will be sorted in the order by which each pictures features are closest to that of the query image. Along with the “Query Images by Intensity + ColorCode” is the relevant checkbox. If this box is checked, there will be checkboxes that pop up under the first 20 pictures. The user can then check these boxes to adjust the weights for each feature. After some boxes are selected, the user can use the same query again to have the query results adjusted based on relevant images.

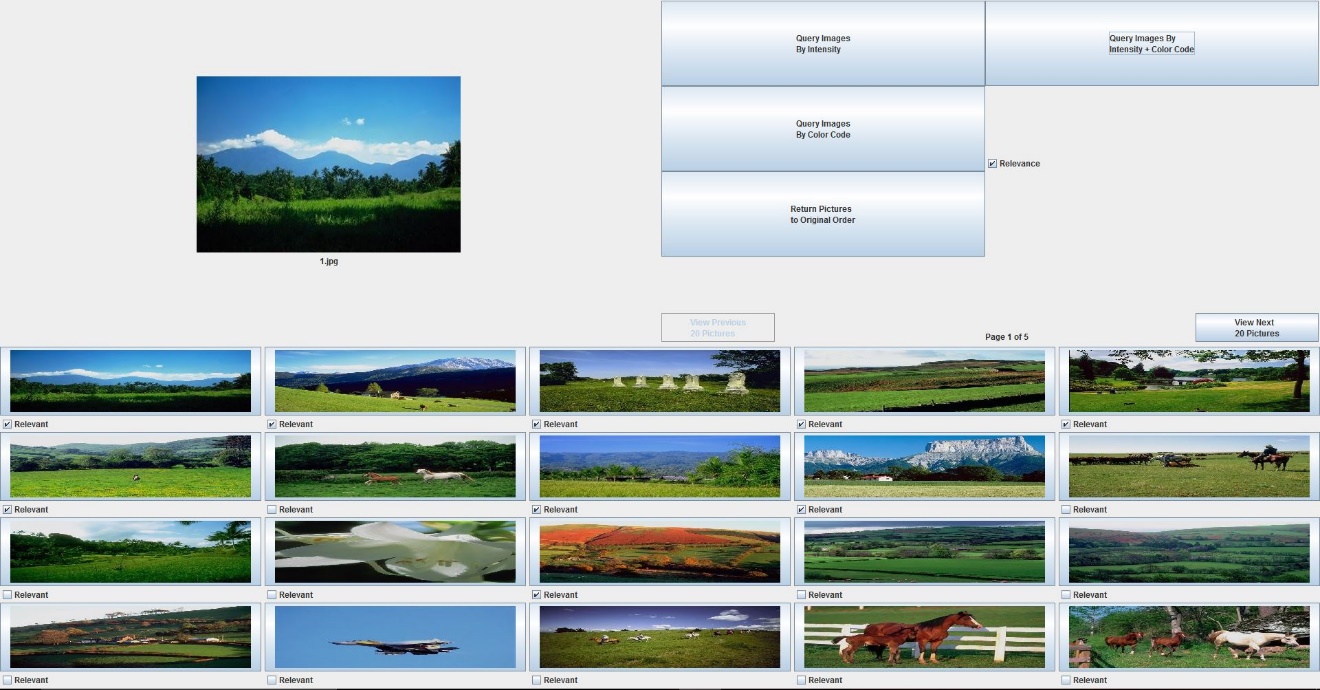
After a query has been done, the closest 20 pictures will be shown in the bottom panel, and as the next button is pressed the list will descend to the pictures the most different query attribute. Also, the “Return Pictures to Original Order” button can be pressed at any time to have the pictures in the bottom panel go back to the order they were in when the application was first started.

**Screen Dump**

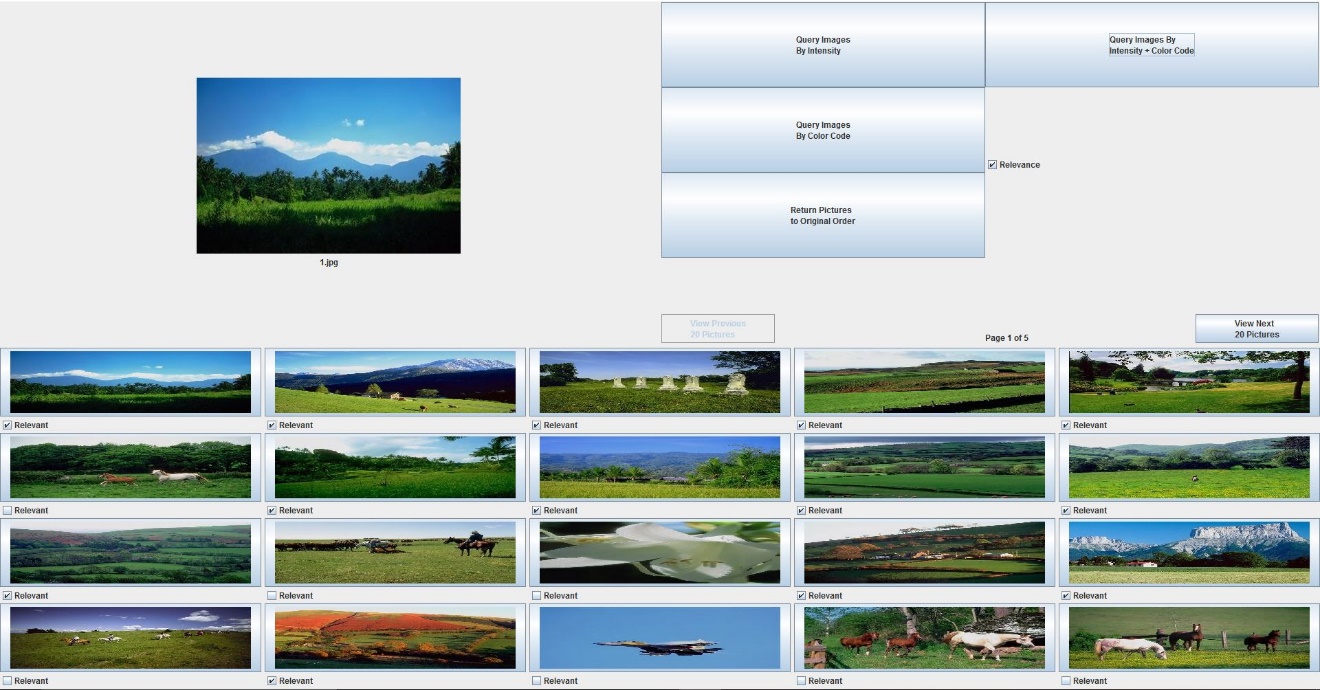
*Figure 1*: Query 1 with no relevance feedback



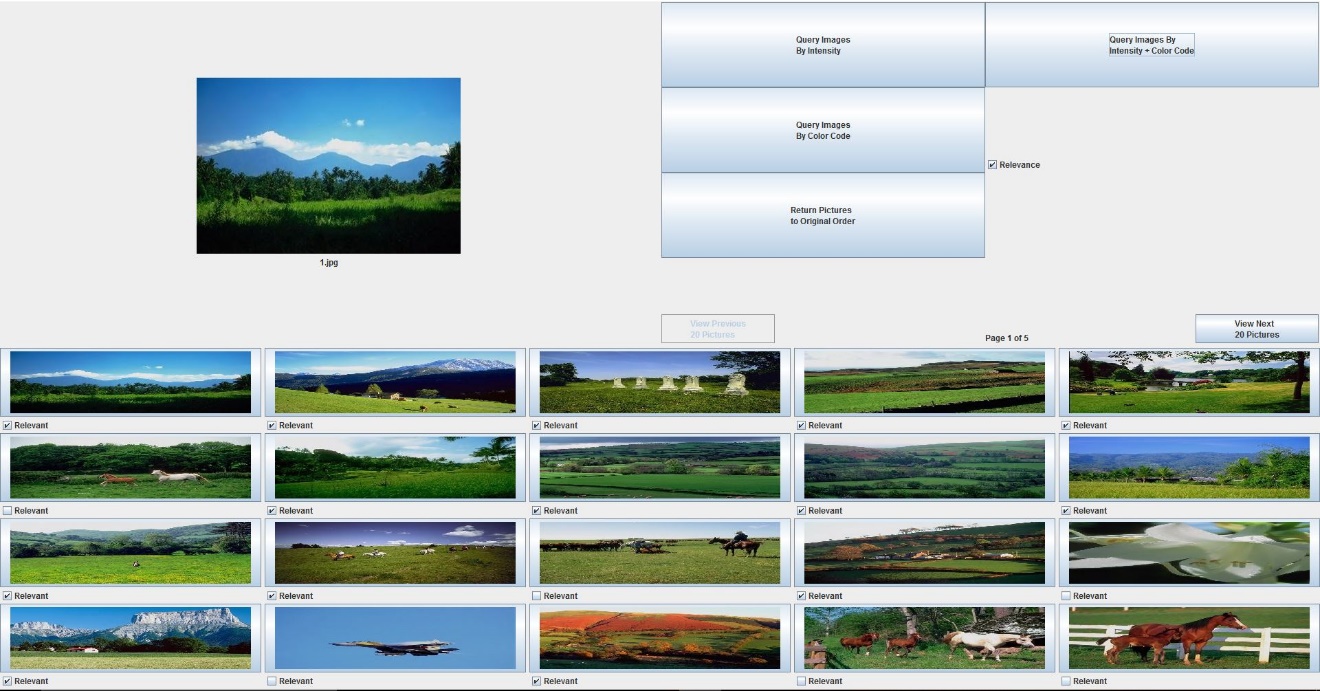
*Figure 2*: Query Iteration 1  
Precision = 9 / 20



*Figure 3*: Query Iteration 2  
 Precision = 13 / 20

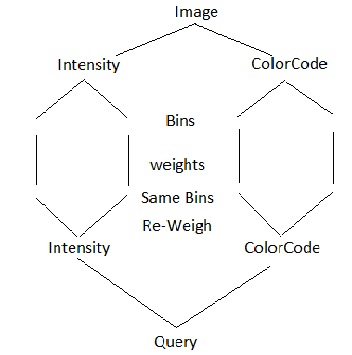


*Figure 4*: Query Iteration 3  
 Precision = 14 / 20



**Proposed vs Simplified RF**

Figure 5: Simplified RF Picture



We used a much simpler version of the relevance feedback feature than what is described in this paper, but the paper had a more generalized concept for the model. This picture kind of shows the approach for our simplified version. The user makes a query and that query is based on features, which has bins with weights that will be adjusted based on the relevant images selected by a user. The weights are then applied to the bins, the bins are compared to one another and as a result we get images from our image pool that are more similar to what was relevant.

**Similarities**

In both the simplified version, as well as the complex one, the weights are initially set to be the same so that every portion of each feature is counted the same. Images similarities to the query are calculated according to a similarity measure and a weight. The similarities/distances of each picture are summed to give us a distance from each picture. Objects are ordered based on their similarity and then the number of most similar that user requested are returned to them. Once the process is over, we collect more data from the user and then begin again with the query process with a new query. The queries are both subjective and they can be updated dynamically through the use of the relevant feedback feature. Both use a linear combination of the lower level features to ensure that feature with more relative importance get more priority when it comes to determining the picture. Both use intra normalization with the same equation that uses the averages of each column as well as the standard deviation to help created a normalized matrix. They also use intra weight updates for selecting the weights, which actually uses the inverse of the calculated standard deviation for a column to get the weight that the column should have for the current query being executed.

**Differences**

The user’s query feedback is distributed among all of the features of the database “evenly” or at least it is pre-determined by us, whereas the complicated version split the feedback based on weight. The features are distributed evenly into histogram bins, not by weight. The overall similarity is not calculated for the simplified version, only each images individual distance from the query. The complex version can make objects as highly relevant, relevant, no-opinion, non-relevant, or highly non-relevant, whereas the simple version is either relevant or not.

I focused mostly on the general RF algorithm and not the normalization and the weight portion, because those offered at least a couple of options as to how you could go about them, and in both cases we selected a method that had been listed.