**MP1 - Metric Learning Write-up**

**Introduction**

In this MP, I worked on completing the Image Retrieval System of a GUI. In particular, I completed two functions: one that deals with retrieval of the top 20 images that were determined to be most ‘similar’ to the chosen image and one that deals with relevance feedback – creating a matrix that aids in finding and returning the aforementioned top 20 images. Both these functions are implemented as call-back functions.

**Methods**

For the image retrieval function, I first computed the centroid (or mean image-vector) of all the images included in the handle.posInds array of indices. To do this, I simply added all the feature vectors together and then divided each row of the resulting sum matrix by the total number of feature vectors added.

The using the relevance feedback matrix (calculated in other aforementioned function) I calculated the Mahalanobis distance between the created centroid and all the images in database. After this, I sorted the top 20 least distances and then found which images they pertained to.

Finally, I returned the indices of the images.

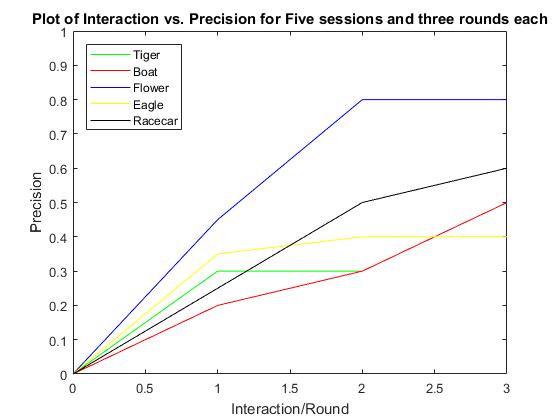
The function then plots all 20 images and then reverts all POS buttons to zero value (not selected).

For the relevance feedback function, I analyzed the already selected positive images (except in round zero when no images have been selected as positive - in this case I simply return an 47x47 identity matrix) and created 47x47 weighting matrix. The way this was done was to first find the variance of the value of each feature across the feature vectors of the positively labelled images. Then, I created a 47x47 diagonal matrix where the values on the diagonal are the reciprocal of the variances calculated for each of the 47 features across all the feature vectors of the postively labelled images plus 0.2222 (for regularization). This matrix is weighting matrix that was desired and so will be returned as a result.

After both functions were implemented, I conducted experiments on the GUI in five sessions. Each session (launching of the GUI) would present a new chosen image and I would then start three rounds of querying. I would calculate the number of positive (relevant) images and then find the precision (number of positive images by total number of images or 20).

**Results**

After the five sessions, I had a total of 15 precision values. I plotted them based on which session they were part of and also provided information on what was shown in that session.



**Discussion**

**What is the general trend of precision versus interaction round? (Or, what should it be?)**

The general trend of the precision plots either a clear increase in precision as the rounds progressed or a plateauing of the results. Ideally, there wouldn’t be any plateauing and the precision would increase without fail for all rounds in all sessions.

**Why do you think certain plots are different than others? (from an image representation perspective)**

Some plots may have an excess of similarities with plots that are not at all related to the subject matter of the plot. For instance, I would consistently have issues with the sunset plot as it would return too many plots relating to gorillas and monkeys. Perhaps even through I choose all plots that pertained to sunsets during the interaction round, there was so many black pixels in those plots (or perhaps another similarity) that images of black colored gorillas were returned. Other plots may only have an excess of similarities across all features with images that pertain to the subject matter of the initial plot.