Seunghyup Alex Oh (916201993)

Problem Set#3

1. Short answer problems [10 points]

1. What exactly does the value recorded in a single dimension of a SIFT keypoint descriptor signify?

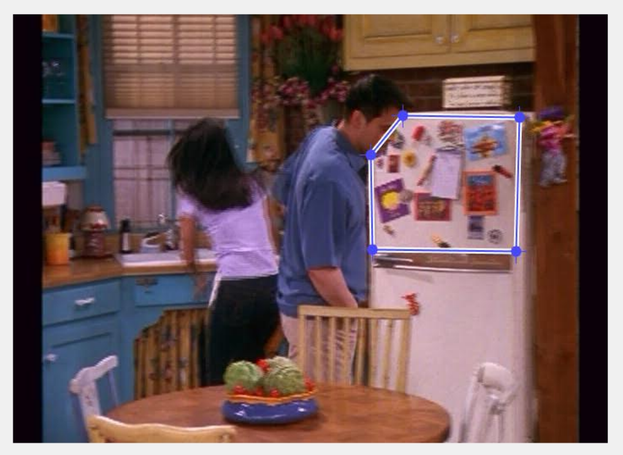
A SIFT descriptor is a 3D spatial histogram of the image gradients in characterizing the appearance of a keypoint which uses the histograms to bin pixels within sub-patches according to their orientation. Namely, the values recorded in a single dimension of the SIFT keypoint descriptor which are part of the histograms are the magnitude of bins of the gradient orientations of the descriptor. They specify each sub-patch in the keypoint descriptor.

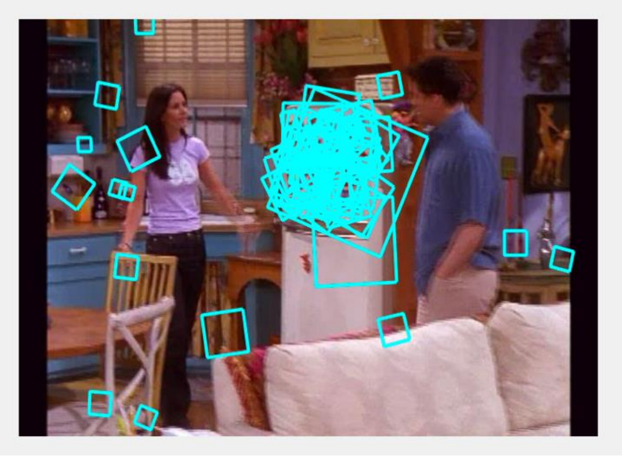
2. When performing interest point detection with the Laplacian of Gaussian, how would results differ if we were to (a) take any positions that are local maxima in scale-space, or (b) take any positions whose filter response exceeds a threshold? Specifically, what is the impact on repeatability or distinctiveness of the resulting interest points?

A repeatability refers the same feature being found in several images despite geometric and photometric transformations. And a distinctiveness refers the particular feature which is different from other features and easy to recognize. (a) – if we take any positions that are local maxima in scale-space, we will have different values of the scale of the interest point for each run. Thus, it will have greater repeatability but less distinctiveness on the resulting interest points. (b) – On the other hand, if we take any positions whose filter response exceeds a threshold, as the result will truly depend on the value of threshold, the local maxima won't be passed, and the result will be an empty set for the interest point. Thus, it will have greater distinctiveness but less repeatability on the resulting interesting points.

2. Programming: Video search with bag of visual words [90 points]

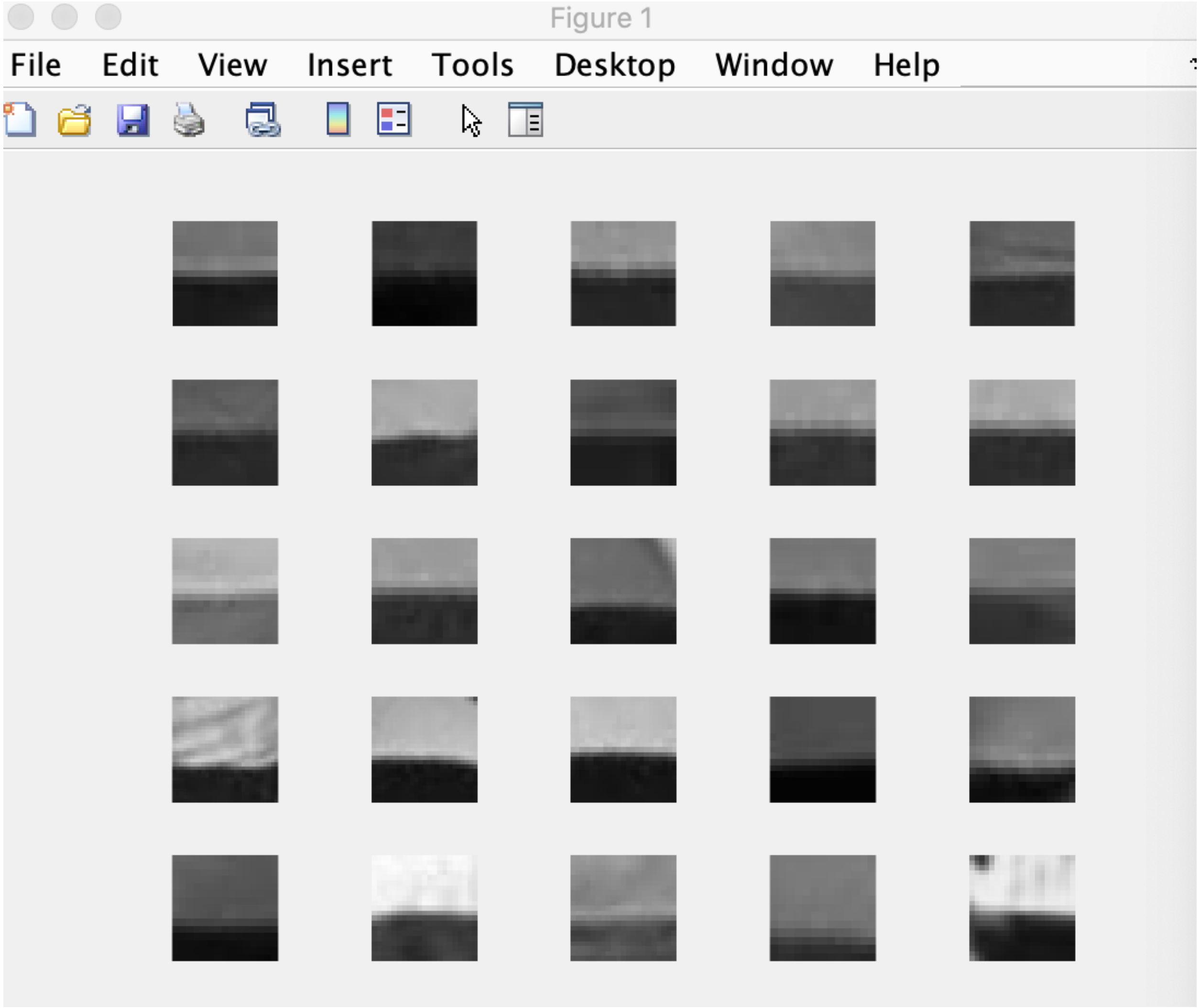
1. Raw descriptor matching [20 pts]:

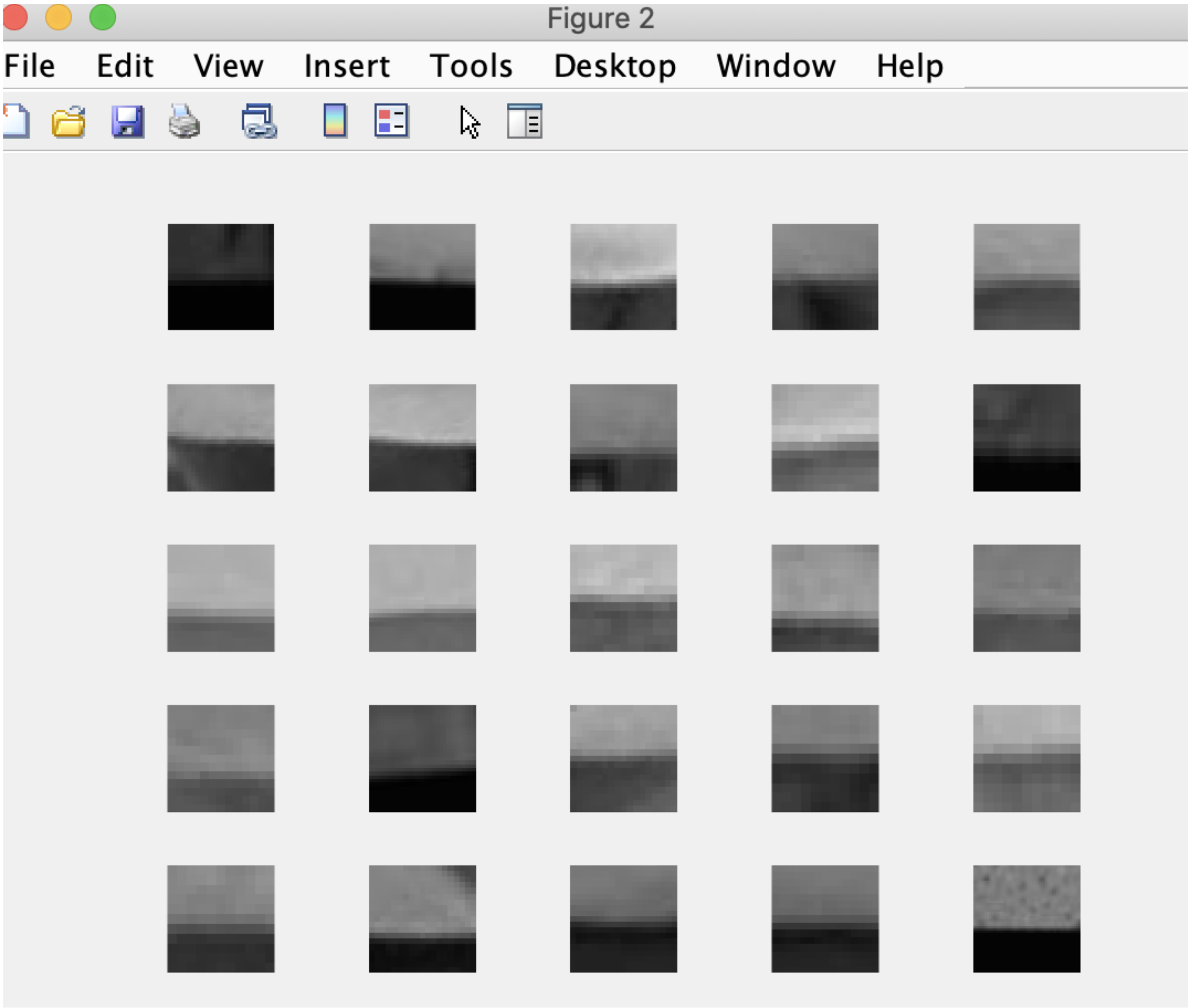




The script, raw_descriptor_matches.m, first uses selectRegion.m to allow a user to select a region of interest from the first image from twoFrameData.mat. Then the script matches descriptors in the user selected region to corresponding descriptors of the second image based on Euclidean distance in SIFT space. I could've set the threshold to less than 0.8 (which I currently set it as) to make less errors for the top half of the fridge something between 0.4 and 0.5 but I wanted the result to match the given image from the instruction.

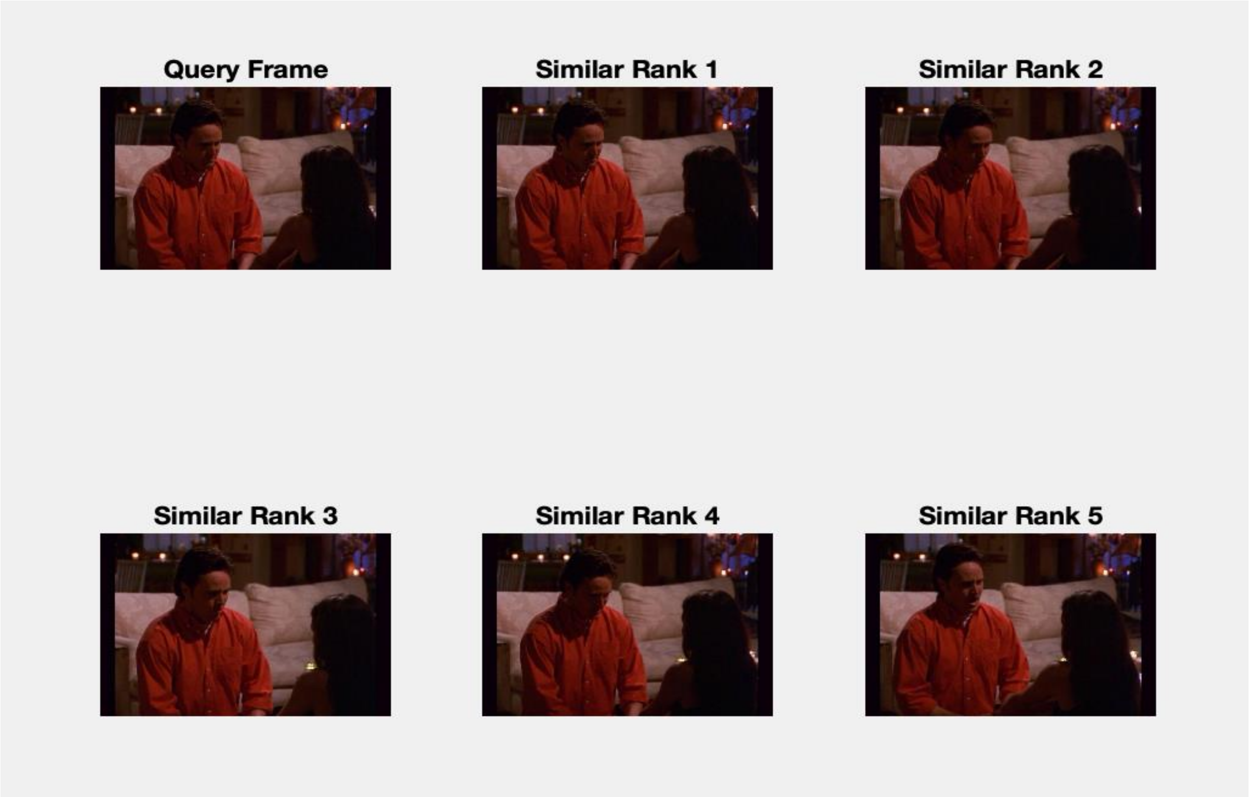
2. Visualizing the vocabulary [25 pts]:

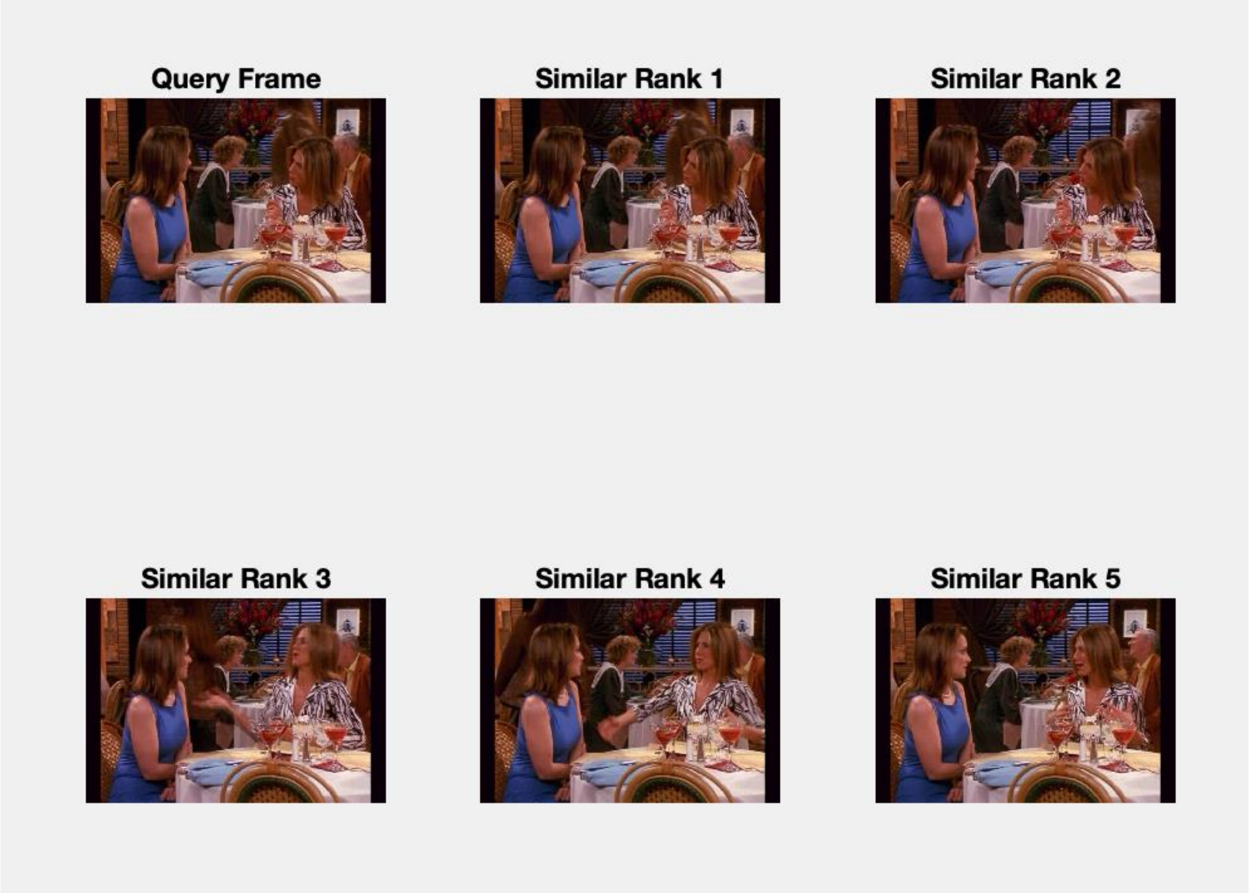


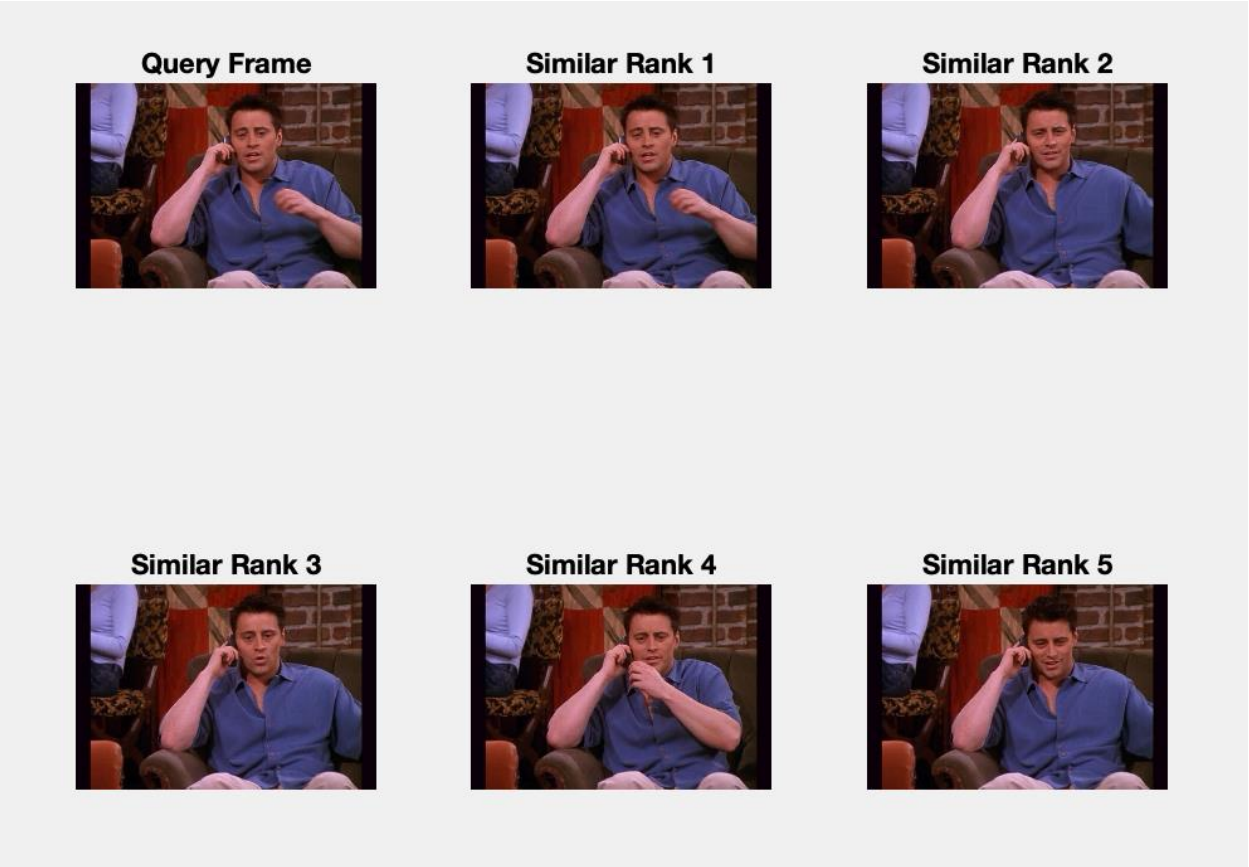


The script, visualize_vocabulary.m, first chooses two words which are the most commonly used and distinct enough to illustrate what the different words are capturing. Then the script samples 200 random sampled descriptors (enough to display 25 patches per word – initially used 500 but took too long about more than 3-5 minutes so changed it to 200 for all of our sake) descriptors per every 20 frames from frames directory. Given with k = 1500, kmeansML.m is applied and saved the visual words in kMeans.mat as kMeans. 25 patches for each of the both chosen visual words are clustered is displayed. As you can see from the result, the most common feature drawn from the two most words used is edge from light color top to dark color below it.

3. Full frame queries [25 pts]:


Query Frame
Similar Rank 1
Similar Rank 2
Similar Rank 3
Similar Rank 4
Similar Rank 5


Query Frame
Similar Rank 1
Similar Rank 2
Similar Rank 3
Similar Rank 4
Similar Rank 5

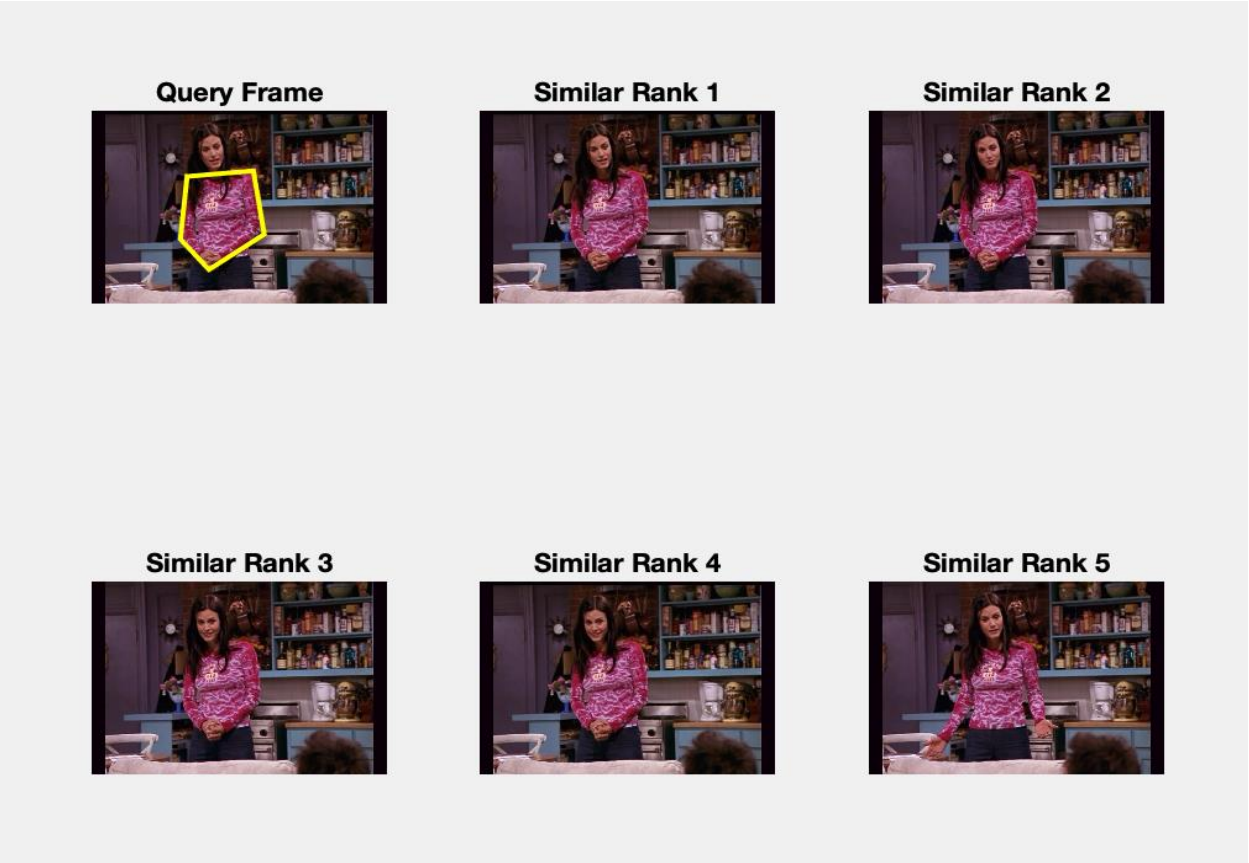


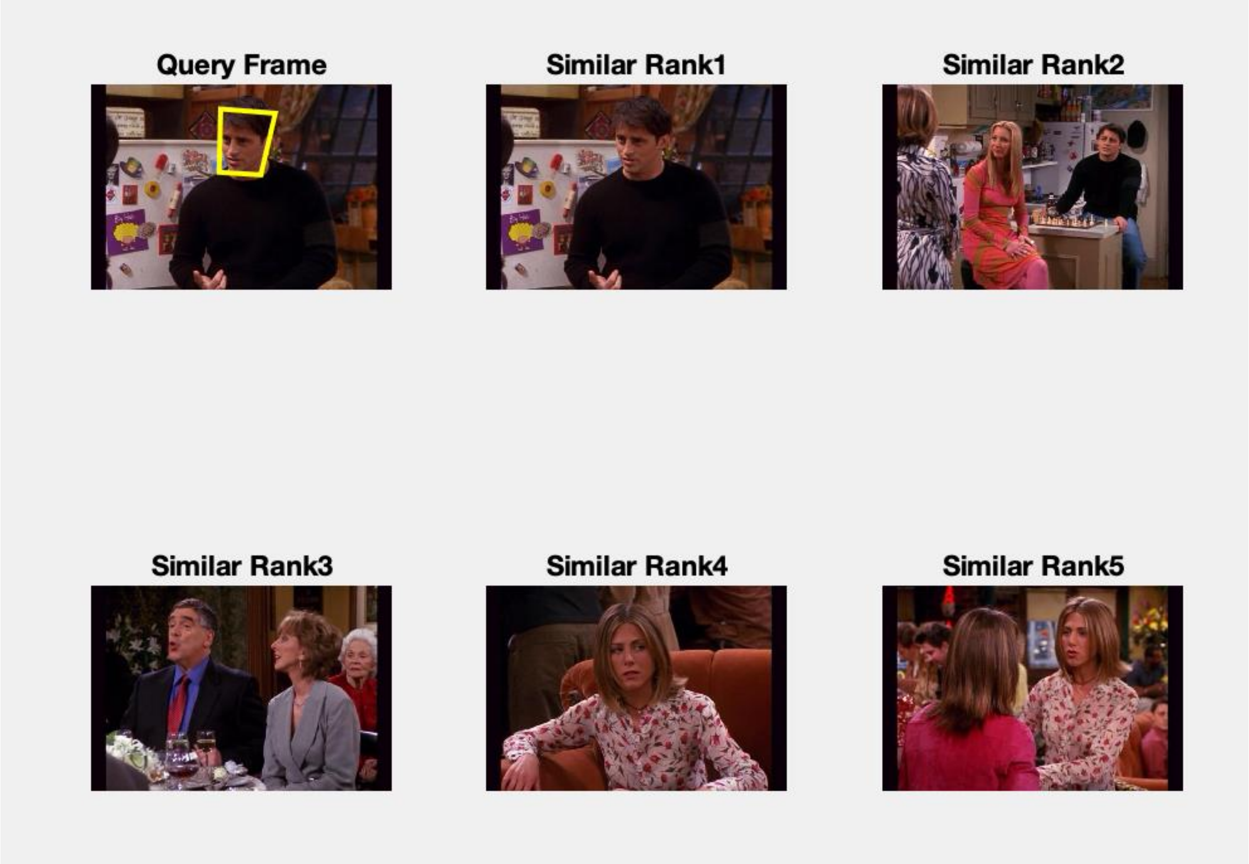
The script, full_frame_queries.m, I chose three different frames from the entire video dataset to serve as query images and before running the script. I created the histogram with the precalculated data (Kmeans.mat) with using makeHistogram.m, getHistogram.m. And then the histogram data is saved in histograms.mat and loaded into the script. First, I get to choose three images from the frame directory. The similarity between the three query frames and the rest frames from the directory is calculated with rankingCheck.m and ranked with sortRanking.m. And then displayed the query images along with top 5 similar images to it.

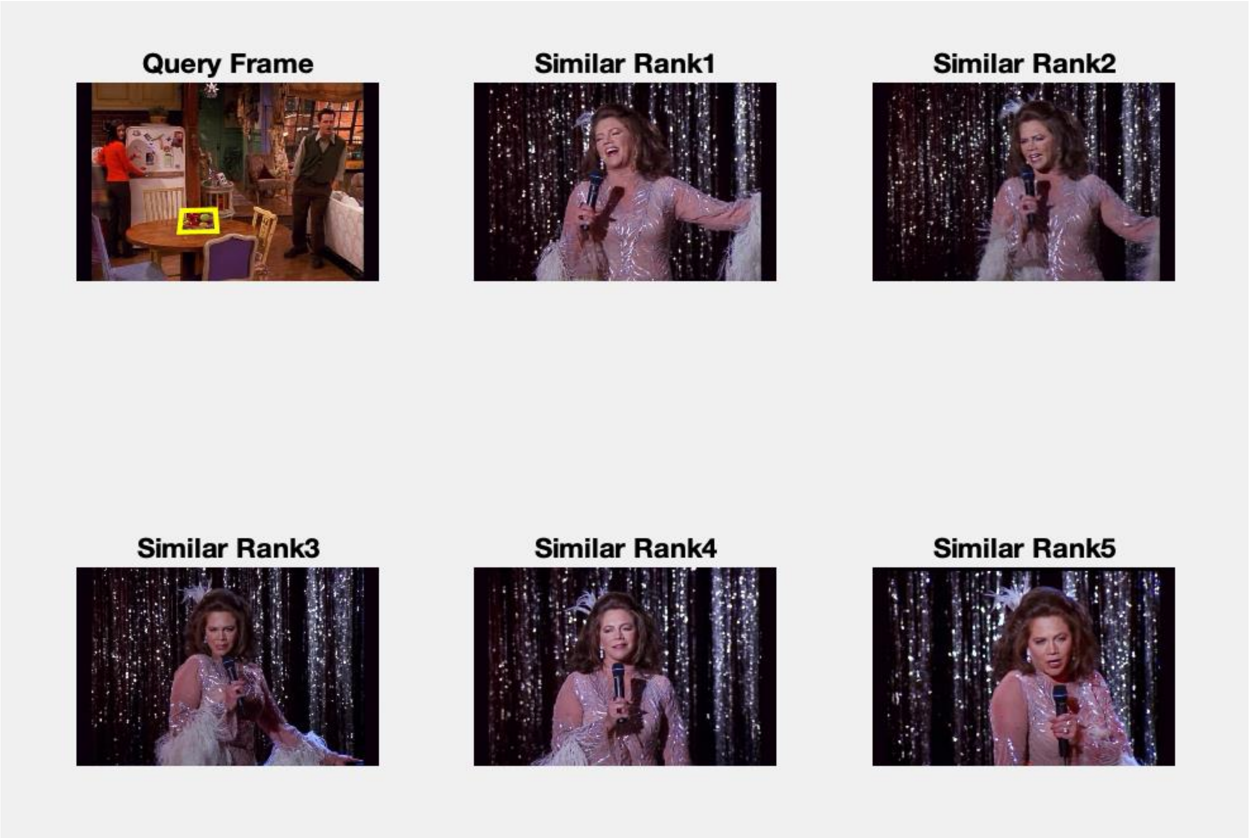
As you can see from the image above, the background, character's action, outfits, etc. from the query frame and the rest similar images are pretty much similar.

```
friends_0000000060.jpeg, friends_0000000285.jpeg, friends_0000002729.jpeg
```

4. Region queries [20 pts]:


Query Frame
Similar Rank 1
Similar Rank 2
Similar Rank 3
Similar Rank 4
Similar Rank 5


Query Frame
Similar Rank1
Similar Rank2
Similar Rank3
Similar Rank4
Similar Rank5



The script, region_queries.m, first let users to specify the region of his interest on the query image. And the script uses selectedRegion.m to specify which region the user has specified the region of the interest. The selectedRegion.m gets the descriptors from it and create histograms and then process the rest as Full frame queries, sort the rank and display the similarity rank along with the query image specified with the particular region of the interest.

The first set is successful case which the selected region of interest was the female character's pink shirt and as it has the unique pattern and color, it has successfully accomplished finding the matching image. The second set found the most similar M frames but has different objects (failure case) amidst of successful case. The set chose the male character's face and the first and second image includes the best matching face, but the rest three images contain different faces. The third set is total failure case which the selected region of interest was the apple on round table, but the matching results displays a woman singing in glaring dress.