Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

**DOCUMENTATION:**

Q1. Related files:

- getImageFeatures.m:
    - This function takes in a single image and a preference for feature generation (SIFT or DSIFT). Returns a n x 128 matrix containing the SIFT descriptors (1 x 128) per n detected features.
- getImagesFeatures.m:
    - Takes a folder containing the training images and preference for feature generation (SIFT or DSIFT). Function will iterate through all images and find all features, storing the descriptors in a single matrix.

Q2. Related files:

- getVladModel.m
    - Takes features of the training set, as well as the specified number of clusters and desired reduced dimensions of the features. Because VLAD requires "data-to-cluster" assignments, this function generates a "codebook" or list of centroids using kmeans.
- getFisherVectorModel.m
    - Takes features of the training set, as well as the specified number of clusters and desired reduced dimensions of the features. Uses the reduced features to train a GMM model. Obtains the mean, covariance, and prior and saves it to the model.

Q3. Related files:

- getVladAggregation.m
    - Takes VLAD model (i.e. kmeans codebook) and features and uses them to generate a kd-tree. The indices of the nearest centroid to each vector in the features is generated using the kdtree. An assignment matrix is then created to assign each descriptor to a cluster. The results are then encoded using VLAD.
- getFisherVectorModel.m
    - Takes GMM model and features and encodes them using Fisher Vector.
- NOTE: Both models recalculate the PCA and reduces the features, as was done when the models were generated. This was done because the features needed to be projected to the desired lower dimension to match the VLAD model dimension before aggregation.

Q4. Related files:

Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

- getDistances.m
    - The purpose of this function is to get the distance between the images in the pairs of the ground truth set. After the Euclidean distance is calculated between the features of the images, the distance is flattened by using the mean. This allows for the result to be passed to vl_roc.

NOTE: I have generated the ROC for all 24 possible feature/aggregation models. However, being utterly new to MATLAB and incompetent with figure formatting, I also generated the required 10 using a Jupyter notebook in an attempt to make the ROC more legible. The Jupyter notebooks are best viewed using my GitHub repo. Furthermore, I ended up only using 25 matching and 25 non-matching pairs for my ground truth, as it was taking impractically long to run on my machine.

- DSIFT_FV_Models.mlx
- DSIFT_VLAD_Models.mlx
- SIFT_FV_Models.mlx
- SIFT_VLAD_Models.mlx
- FV_Models.ipynb
- VLAD_Models.ipynb

Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

Homework-2: In this assignment you will practice DenseSIFT and SIFT feature extractions from images, and then testing out VLAD and Fisher Vector aggregation schemes to generate a uniform dimension feature representation for images. Then for a given set of images and their pair wise matching/non-matching info from the CDVS data set,

https://umkc.box.com/s/43fq7oumycl08ise6g8p10f0fsn21t29


Images are in the cdvs_thumbnail folder, while matching image pairs are in the file:*fnames_all-mp_fid-nmp_fid-n_mp-n_nmp.mat*. You will test the performance of different combination of features and aggregation schemes to see which one gives us the best performance in TPR-FPR ROC.

[Q1, 20pts] Compute Image Features, create a matlab/python function that compute n x d features by calling vl_feat DSIFT and SIFT functions (notice that vl_feat also has Python version), implementing the following function:

```
% im - input images, let us make them all grayscale only, so it is a h x w matrix
```

Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

```
%    opt.type = { 'sift', 'dsft'} for sift and densesift
%    f - n x d matrix containing n features of d dimension
function [f]=getImageFeatures(im, opt)
```

[Q2, 20pts] Compute VLAD and Fisher Vector models of image features, for this purpose you need to first compute a Kmeans model for DenseSIFT and SIFT. Use the CDVS data set given as both training and testing for convenience (not the right way in research though, should use a different data set, say FLICKR MIR, or ImageNet), implementing the following functions:

```
% f - n x d matrix containing training features from say 100 images.
% k - VLAD kmeans model number of cluster
% kd - desired dimension of the feature
% vlad_km - VLAD kmeans model
% A - PCA projection for dimension reduction
function [vlad_km, A]=getVladModel(f, kd, k)
% PCA dimension reduction of the feature
[A,s,lat]=princomp(f);
f0 = f*A(:,1:kd); % this is the feature with desired d-dimensions
.....
```

Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

```
% f - n x d matrix containing n features from say 100 images.
% k - number of GMM components
% kd - desired lower dimension of the feature
% fv_gmm - FisherVector GMM model:
%                fv_gmm.m - mean, fv_gmm.cov - variance, fv_gmm.p - prior
% A - PCA for dimension reduction
function [fv_gmm, A]=getFisherVectorModel(f, kd, k)
[A,s,lat]=princomp(f);
f0 = f*A(:,1:kd); % this is the feature with desired d-dimensions
.....
```

%hint from last year's gmm training with a set of dimensions and number of clusters

Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

```
%%%%%%%%%%%%%%%%%%%%%%%%
% train SIFT PCA, gmm,
%%%%%%%%%%%%%%%%%%%%%%%%
doTrainGMM=0;
if doTrainGMM
    % load SIFT training data
    load data/hw-2-data.mat;
    [A, s, lat]=princomp(double(sift_gmm_km_train));
    figure(21); grid on; hold on; plot(lat, '.-');

    kd=[16, 24]; nc = [16, 32, 64, 96];
    t0=cputime;
    for j=1:length(kd)
        for k=1:length(nc)
            fprintf('\n t=%1.2f: GMM: kd(%d)=%d, nc(%d)=%d: ', cputime-t0, j, kd(j),
k, nc(k));
            % training data reduce dimension via PCA
            x = double(sift_gmm_km_train)*A(:,1:kd(j));
            % gmm model:
            [gmm(j,k).m, gmm(j,k).cov, gmm(j, k).p]=v1_gmm(x', nc(k),
'MaxNumIterations', 30);
        end
    end
    % save PCA and GMM model:
    save data/hw2-A-gmm-kd-16-24-nc-16-32-64-96.mat A gmm;
else
    load data/hw2-A-gmm-kd-16-24-nc-16-32-64-96.mat A gmm;
end
```

[Q3, 20pts] Compute VLAD and Fisher Vector Aggregation of Images, from the given VLAD and FV models, implementing the following functions. Notice that the feature nxd f need to be projected to the desired lower dimension via, f0=f*A(:,1:kd), to match the VLAD model dimension before calling this function.

Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

```
% f - n x d matrix containing a feature from an image by calling f=getImageFeature(im,..
% vlad_km - VLAD kmeans model
function [vlad]=getVladAggregation(vlad_km, f)
```

```
% f - n x d matrix containing a feature from an image by calling f=getImageFeature(im,..
% fv_gmm - GMM Model from features, has m, cov, and p.
function [fv]=getFisherVectorAggregation(fv_gmm, f)
```

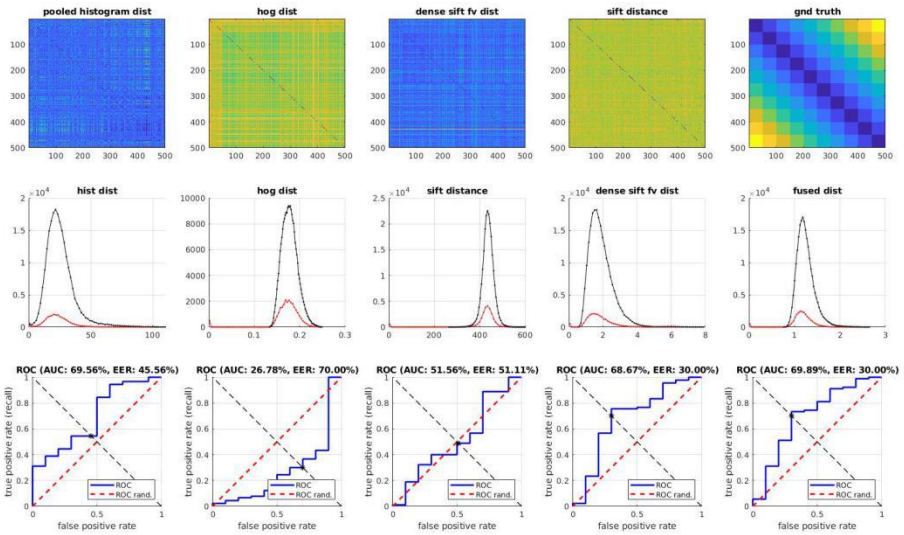
Student Name:__Shelby Mohar_____, Student ID: ____16151180_____

[Q4, 20pts] Now benchmarking the TPR-FPR performance of various feature and aggregation scheme performance against the mini CDVS data set. For the SIFT and DenseSIFT features, let us have kd=[24, 48], nc=[32, 64, 96]. So for each image, we will have 2 x 6 x2 = 24 different feature + aggregation representations. For the total of N images in the mini CDVS dataset, we have M=N*(N-1)/2 total image pairs, and the matching pairs ground truth are given, we only care about the first 100 matching pairs and first 100 non-matching pairs in the fid.mat, which has two variables mp and nmp. Each has a row of two image filenames to their associated images, e.g, mp(1,:): mp_2.jpg and mp1_2.jpg are two matching pairs:


And nmp(1,:) contains file names for the following non-matching pairs


Let us use the Euclidean distance to on those features to compute the TPR-FPR ROCs and find out which one have the best performance. Last year's plots are attached below for example, you only need to plot the last row for 10 feature-aggregation combinations.

Student Name:__Shelby Mohar_____, Student ID: ____16151180_____




[Q5. bonus 30pts]: Compute SIFT and DenseSIFT for the NWPU Aerial image data set, and show the VLAD and FV aggregation results. Notice that you can choose your own PCA dimension and K for kmeans and GMM. To make this easier, only validate against the first 15 classes and 20 images per class. So you will have a 300x300 pdist2() matrix and those sharing the same semantic label will be considered a TP pair, while others TN pair, analyze which combination (2 feature x 12 aggregation) gives the best performance.