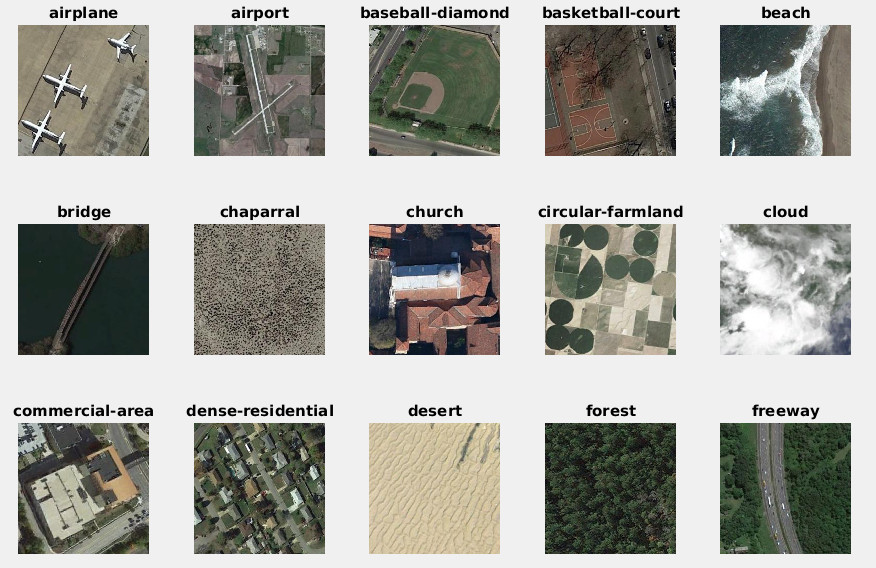
1. Compute Histogram for image recognition (50pts)

Data set: the NWPU aerial data set contains approximately 45 categories of 700 images for each category in 256x256 RGB format. The data set can be downloaded from:

<https://umkc.box.com/s/fxvzh5qq2tiob6eklfxfwn89kg3e1io1>



For HW-1, let us just take the first 15 classes (shown above) {airplane, airport, bridge, ..., freeway} from the full data set, and 40 images per class and form a data set of N=600 images with 15 labels. Objectives are,

1. compute a HSV Kmeans model with total number of entry K=[64, 128] implementing in matlab (or python) and show your code and results [15pts]

|  |
| --- |
| function [color\_table]=getHSVColorTable(im\_path, K)    ....  return; |

1. for each image compute a color histogram, implement the following function in matlab or python with both an Euclidean distance and KL distance (also known as cross entropy) metrics (optional for 10pts extra credits) [15pts]

|  |
| --- |
| function [h]=getHSVHist(im, color\_table, opt)  im = rgb2hsv(im); [ht,wid,~]=size(im);  If opt == ‘euc’ % euclidean distance  dist = pdist2(color\_table, reshape(im, [ht\*wid, 3]));  elseif opt =’kld’ % KL distance  end  ....  return; |

1. Image Recognition Experiment: Use this HSV histogram feature to represent images, you will have a NxK data matrix, for N=15\*40, and K =[64, 128]. Split the data set into training and *leave 1 out testing*, and conduct classification using knnclassify(). [20pts]

|  |
| --- |
| % load hsv features N x K  load hsv\_hist64.mat; % 400x64  size(hsv\_hist)  test\_set=[1:10]; train\_set=[11:40];  n\_class=15; n\_img=40;  for k=1:n\_class  offs=(k-1)\*n\_img;  % test: offs+ test\_set, training: all samples minus test set  rec\_label = knnclassify(hsv\_hist(offs+test\_set,:), hsv\_hist(setdiff(offs+test\_set, [1:n\_class\*n\_img]),:), k\*ones(1, length(train\_set)));  % your code here to analyze accuracy  .......  end |

1. (50pts) Homograph estimation: use the SVD to solve the over-fitting Ah=0 as covered in class. Use the sample code as template: <http://www.vlfeat.org/applications/sift-mosaic-code.html,> compute homograph for the following two images, as well as two images taken by yourself:

|  |  |
| --- | --- |
| sift-1 | sift-2 |

First compute SIFT and SIFT matches, then select good matches to solve homography as an over-determined linear equation via SVD. Show your code and solution details, as well as the final homography matrix. Use matlab imtransform to wrap the image according to the homography and verify the accuracy.

1. How many SIFT points pairs you found ? Select the best M=8 matching pairs, and use that to compute your Matrix A, for Ah=0 homography solution: [15pts]
2. Show the SVD of A and what are the zero singular values ? [10 pts]
3. Show the null space solution to Ah=0, i.e,non-zero h that makes Ah=0 [10pts]
4. For all matched SIFT points, how many are in agreement with this homography ? [15pts]