Hengyue Liu
4107-2966-75
hengyuel@usc.edu
CSCI574 Homework #5
November 20, 2015

Problem: Bag of Features Object Recognition

I Abstract and Motivation

The idea is introduced by Csurka *et al* with the ECCV paper[i]. In the paper, the method *bag of keypoints* is generated using K-means clustered SIFT features with feature reduction by PCA. So the feature for each natural image is transformed to the histogram of bag of features. Thus, the image can be recognized with classifiers by comparing the histograms of training images. In this paper, Naïve Bayes and SVM classifiers are applied and compared. However, in this assignment, one can use the k-nearest neighbor classifier to deal with small size of dataset say 100 training images as well as 50 test images.

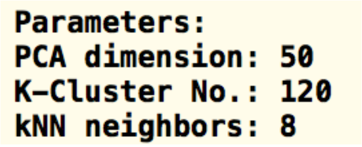
II Approach and Procedures

1. Read all training images (100 in total, 20 for each category), compute the SIFT features for each image stored as n*128 matrix.
2. Conduct PCA feature dimension reduction for each n*128 feature matrix. In this step, one can choose the number of features to reduce, suppose 20 features are needed, then the feature matrix is transformed to n*20.
3. First two steps generate PCA-SIFT features for each image. Combine all 100 matrices into one larger N*20 matrix, then apply k-means clustering for rows of the matrix. One can choose k arbitrarily, if choose k=100, then each row of the matrix is labeled with a particular label within 0~99. Meanwhile, k mean vectors of each label center are generated as the codebook for assigning codewords for each image. By the way, in k-mean clustering, the number of trial times is set as 50 times.
4. For each image (training or testing), compute the histogram of codewords.
5. For each histogram of test image, find the k-nearest neighbor histograms of training images. In this step, the vote is weighted by the distance calculated between test histogram and training histogram of that neighbor. The number of neighbors can be modified, better large than 10.
6. For all 50 test images, make the classification predictions, and show the correct rate.

III Experimental Results

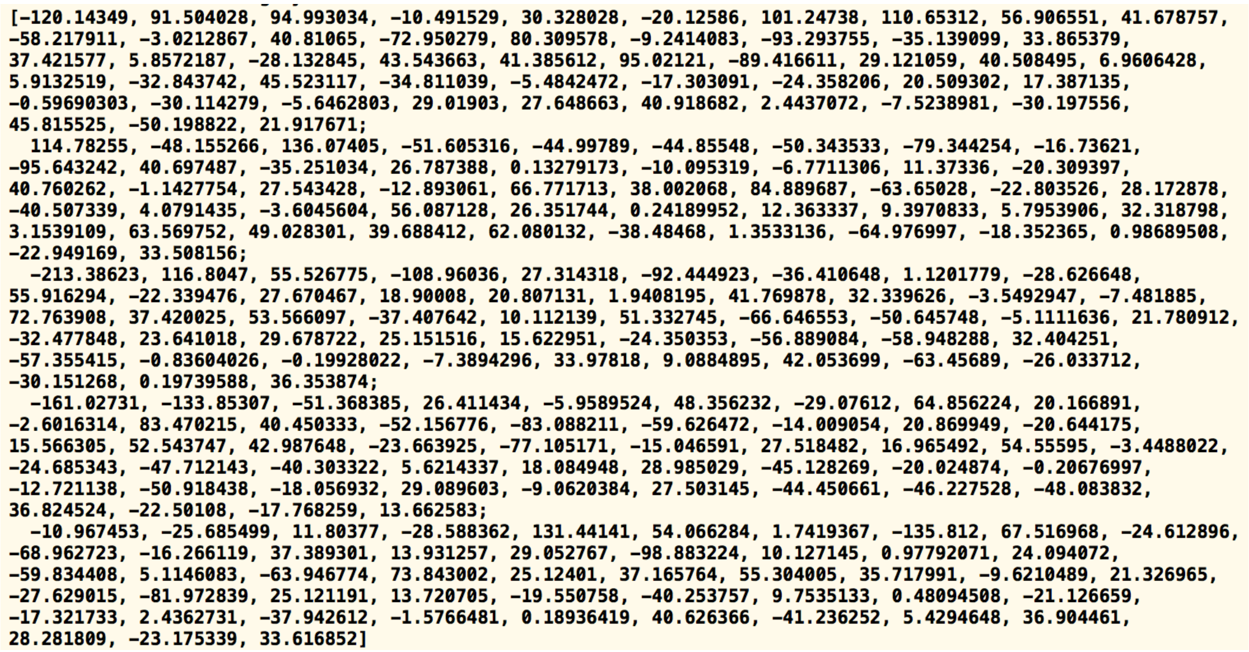
Shown below are the example results for some important steps. There are 5 categories, with each one labeled with {0}-car side, {1}-butterfly, {2}-face, {3}-watch and {4}-water lilly respectively.

The parameters used in this example is:



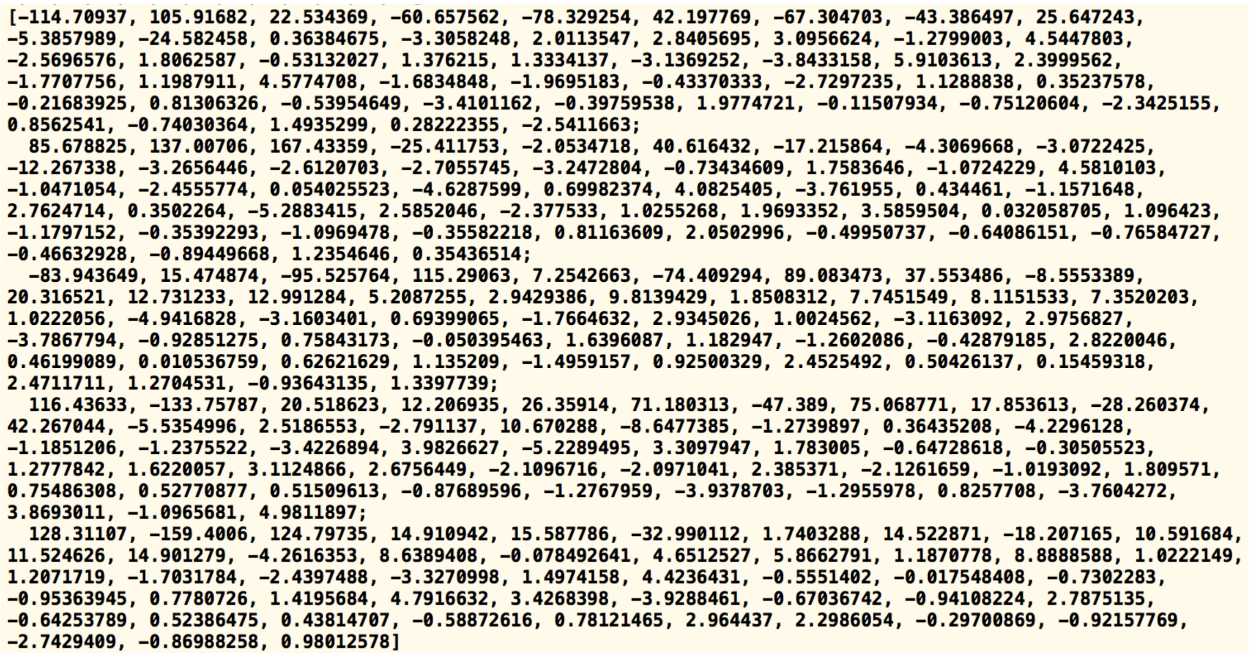
```
Parameters:
PCA dimension: 50
K-Cluster No.: 120
kNN neighbors: 8
```

Figure 1 Parameters for this example



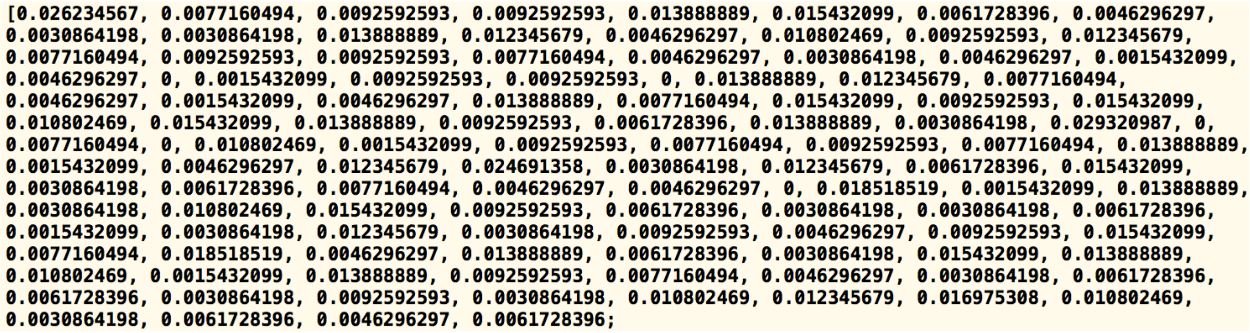
```
[-120.14349, 91.504028, 94.993034, -10.491529, 30.328028, -20.12586, 101.24738, 110.65312, 56.906551, 41.678757,
-58.217911, -3.0212867, 40.81065, -72.950279, 80.309578, -9.2414083, -93.293755, -35.139099, 33.865379,
37.421577, 5.8572187, -28.132845, 43.543663, 41.385612, 95.02121, -89.416611, 29.121059, 40.508495, 6.9606428,
5.9132519, -32.843742, 45.523117, -34.811039, -5.4842472, -17.303091, -24.358206, 20.509302, 17.387135,
-0.59690303, -30.114279, -5.6462803, 29.01903, 27.648663, 40.918682, 2.4437072, -7.5238981, -30.197556,
45.815525, -50.198822, 21.917671;
  114.78255, -48.155266, 136.07405, -51.605316, -44.99789, -44.85548, -50.343533, -79.344254, -16.73621,
-95.643242, 40.697487, -35.251034, 26.787388, 0.13279173, -10.095319, -6.7711306, 11.37336, -20.309397,
40.760262, -1.1427754, 27.543428, -12.893061, 66.771713, 38.002068, 84.889687, -63.65028, -22.803526, 28.172878,
-40.507339, 4.0791435, -3.6045604, 56.087128, 26.351744, 0.24189952, 12.363337, 9.3970833, 5.7953906, 32.318798,
3.1539109, 63.569752, 49.028301, 39.688412, 62.080132, -38.48468, 1.3533136, -64.976997, -18.352365, 0.98689508,
-22.949169, 33.508156;
  -213.38623, 116.8047, 55.526775, -108.96036, 27.314318, -92.444923, -36.410648, 1.1201779, -28.626648,
55.916294, -22.339476, 27.670467, 18.90008, 20.807131, 1.9408195, 41.769878, 32.339626, -3.5492947, -7.481885,
72.763908, 37.420025, 53.566097, -37.407642, 10.112139, 51.332745, -66.646553, -50.645748, -5.1111636, 21.780912,
-32.477848, 23.641018, 29.678722, 25.151516, 15.622951, -24.350353, -56.889084, -58.948288, 32.404251,
-57.355415, -0.83604026, -0.19928022, -7.3894296, 33.97818, 9.0884895, 42.053699, -63.45689, -26.033712,
-30.151268, 0.19739588, 36.353874;
  -161.02731, -133.85307, -51.368385, 26.411434, -5.9589524, 48.356232, -29.07612, 64.856224, 20.166891,
-2.6016314, 83.470215, 40.450333, -52.156776, -83.088211, -59.626472, -14.009054, 20.869949, -20.644175,
15.566305, 52.543747, 42.987648, -23.663925, -77.105171, -15.046591, 27.518482, 16.965492, 54.55595, -3.4488022,
-24.685343, -47.712143, -40.303322, 5.6214337, 18.084948, 28.985029, -45.128269, -20.024874, -0.20676997,
-12.721138, -50.918438, -18.056932, 29.089603, -9.0620384, 27.503145, -44.450661, -46.227528, -48.083832,
36.824524, -22.50108, -17.768259, 13.662583;
  -10.967453, -25.685499, 11.80377, -28.588362, 131.44141, 54.066284, 1.7419367, -135.812, 67.516968, -24.612896,
-68.962723, -16.266119, 37.389301, 13.931257, 29.052767, -98.883224, 10.127145, 0.97792071, 24.094072,
-59.834408, 5.1146083, -63.946774, 73.843002, 25.12401, 37.165764, 55.304005, 35.717991, -9.6210489, 21.326965,
-27.629015, -81.972839, 25.121191, 13.720705, -19.550758, -40.253757, 9.7535133, 0.48094508, -21.126659,
-17.321733, 2.4362731, -37.942612, -1.5766481, 0.18936419, 40.626366, -41.236252, 5.4294648, 36.904461,
28.281809, -23.175339, 33.616852]
```

Figure 2 First 5 PCA-SIFT feature vectors for car_side_image0001.jpg



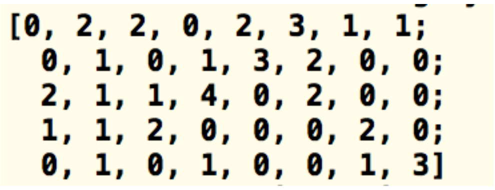
```
[-114.70937, 105.91682, 22.534369, -60.657562, -78.329254, 42.197769, -67.304703, -43.386497, 25.647243,
-5.3857989, -24.582458, 0.36384675, -3.3058248, 2.0113547, 2.8405695, 3.0956624, -1.2799003, 4.5447803,
-2.5696576, 1.8062587, -0.53132027, 1.376215, 1.3334137, -3.1369252, -3.8433158, 5.9103613, 2.3999562,
-1.7707756, 1.1987911, 4.5774708, -1.6834848, -1.9695183, -0.43370333, -2.7297235, 1.1288838, 0.35237578,
-0.21683925, 0.81306326, -0.53954649, -3.4101162, -0.39759538, 1.9774721, -0.11507934, -0.75120604, -2.3425155,
0.8562541, -0.74030364, 1.4935299, 0.28222355, -2.5411663;
  85.678825, 137.00706, 167.43359, -25.411753, -2.0534718, 40.616432, -17.215864, -4.3069668, -3.0722425,
-12.267338, -3.2656446, -2.6120703, -2.7055745, -3.2472804, -0.73434609, 1.7583646, -1.0724229, 4.5810103,
-1.0471054, -2.4555774, 0.054025523, -4.6287599, 0.69982374, 4.0825405, -3.761955, 0.434461, -1.1571648,
2.7624714, 0.3502264, -5.2883415, 2.5852046, -2.377533, 1.0255268, 1.9693352, 3.5859504, 0.032058705, 1.096423,
-1.1797152, -0.35392293, -1.0969478, -0.35582218, 0.81163609, 2.0502996, -0.49950737, -0.64086151, -0.76584727,
-0.46632928, -0.89449668, 1.2354646, 0.35436514;
  -83.943649, 15.474874, -95.525764, 115.29063, 7.2542663, -74.409294, 89.083473, 37.553486, -8.5553389,
20.316521, 12.731233, 12.991284, 5.2087255, 2.9429386, 9.8139429, 1.8508312, 7.7451549, 8.1151533, 7.3520203,
1.0222056, -4.9416828, -3.1603401, 0.69399065, -1.7664632, 2.9345026, 1.0024562, -3.1163092, 2.9756827,
-3.7867794, -0.92851275, 0.75843173, -0.050395463, 1.6396087, 1.182947, -1.2602086, -0.42879185, 2.8220046,
0.46199089, 0.010536759, 0.62621629, 1.135209, -1.4959157, 0.92500329, 2.4525492, 0.50426137, 0.15459318,
2.4711711, 1.2704531, -0.93643135, 1.3397739;
  116.43633, -133.75787, 20.518623, 12.206935, 26.35914, 71.180313, -47.389, 75.068771, 17.853613, -28.260374,
42.267044, -5.5354996, 2.5186553, -2.791137, 10.670288, -8.6477385, -1.2739897, 0.36435208, -4.2296128,
-1.1851206, -1.2375522, -3.4226894, 3.9826627, -5.2289495, 3.3097947, 1.783005, -0.64728618, -0.30505523,
1.2777842, 1.6220057, 3.1124866, 2.6756449, -2.1096716, -2.0971041, 2.385371, -2.1261659, -1.0193092, 1.809571,
0.75486308, 0.52770877, 0.51509613, -0.87689596, -1.2767959, -3.9378703, -1.2955978, 0.8257708, -3.7604272,
3.8693011, -1.0965681, 4.9811897;
  128.31107, -159.4006, 124.79735, 14.910942, 15.587786, -32.990112, 1.7403288, 14.522871, -18.207165, 10.591684,
11.524626, 14.901279, -4.2616353, 8.6389408, -0.078492641, 4.6512527, 5.8662791, 1.1870778, 8.8888588, 1.0222149,
1.2071719, -1.7031784, -2.4397488, -3.3270998, 1.4974158, 4.4236431, -0.5551402, -0.017548408, -0.7302283,
-0.95363945, 0.7780726, 1.4195684, 4.7916632, 3.4268398, -3.9288461, -0.67036742, -0.94108224, 2.7875135,
-0.64253789, 0.52386475, 0.43814707, -0.58872616, 0.78121465, 2.964437, 2.2986054, -0.29700869, -0.92157769,
-2.7429409, -0.86988258, 0.98012578]
```

Figure 3 First 5 mean vectors of codebook



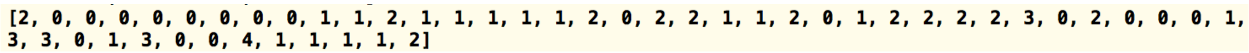
```
[0.026234567, 0.0077160494, 0.0092592593, 0.0092592593, 0.013888889, 0.015432099, 0.0061728396, 0.0046296297,
0.0030864198, 0.0030864198, 0.013888889, 0.012345679, 0.0046296297, 0.010802469, 0.0092592593, 0.012345679,
0.0077160494, 0.0092592593, 0.0092592593, 0.0077160494, 0.0046296297, 0.0030864198, 0.0046296297, 0.0015432099,
0.0046296297, 0, 0.0015432099, 0.0092592593, 0.0092592593, 0, 0.013888889, 0.012345679, 0.0077160494,
0.0046296297, 0.0015432099, 0.0046296297, 0.013888889, 0.0077160494, 0.015432099, 0.0092592593, 0.015432099,
0.010802469, 0.015432099, 0.013888889, 0.0092592593, 0.0061728396, 0.013888889, 0.0030864198, 0.029320987, 0,
0.0077160494, 0, 0.010802469, 0.0015432099, 0.0092592593, 0.0077160494, 0.0092592593, 0.0077160494, 0.013888889,
0.0015432099, 0.0046296297, 0.012345679, 0.024691358, 0.0030864198, 0.012345679, 0.0061728396, 0.015432099,
0.0030864198, 0.0061728396, 0.0077160494, 0.0046296297, 0.0046296297, 0, 0.018518519, 0.0015432099, 0.013888889,
0.0030864198, 0.010802469, 0.015432099, 0.0092592593, 0.0061728396, 0.0030864198, 0.0030864198, 0.0061728396,
0.0015432099, 0.0030864198, 0.012345679, 0.0030864198, 0.0092592593, 0.0046296297, 0.0092592593, 0.015432099,
0.0077160494, 0.018518519, 0.0046296297, 0.013888889, 0.0061728396, 0.0030864198, 0.015432099, 0.013888889,
0.010802469, 0.0015432099, 0.013888889, 0.0092592593, 0.0077160494, 0.0046296297, 0.0030864198, 0.0061728396,
0.0061728396, 0.0030864198, 0.0092592593, 0.0030864198, 0.010802469, 0.012345679, 0.016975308, 0.010802469,
0.0030864198, 0.0061728396, 0.0046296297, 0.0061728396;
```

Figure 4 Histogram of codewords for car_side_image0001.jpg



```
[0, 2, 2, 0, 2, 3, 1, 1;
  0, 1, 0, 1, 3, 2, 0, 0;
  2, 1, 1, 4, 0, 2, 0, 0;
  1, 1, 2, 0, 0, 0, 2, 0;
  0, 1, 0, 1, 0, 0, 1, 3]
```

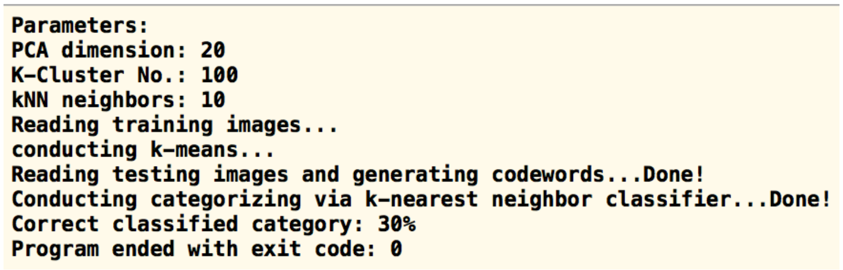
Figure 5 8-nearest neighbors for first 5 test images



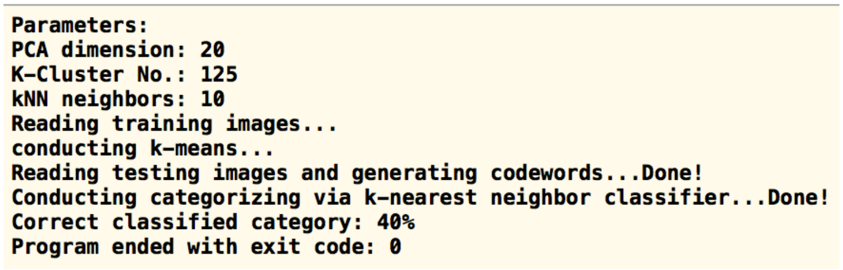
```
[2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1, 1, 1, 1, 2, 0, 2, 2, 1, 1, 2, 0, 1, 2, 2, 2, 2, 3, 0, 2, 0, 0, 0, 1,
3, 3, 0, 1, 3, 0, 0, 4, 1, 1, 1, 1, 2]
```

Figure 6 Prediction results for 50 test images

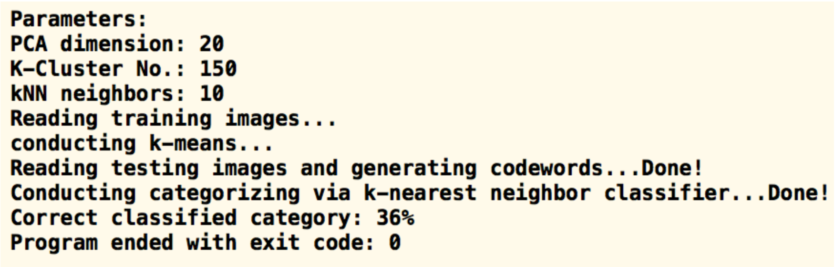
Shown below are the prediction results for several parameter combinations.



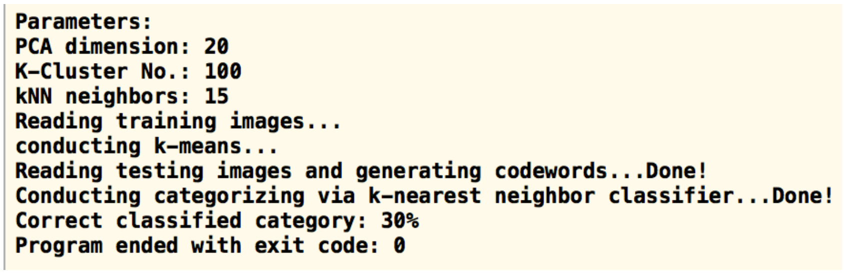
```
Parameters:
PCA dimension: 20
K-Cluster No.: 100
kNN neighbors: 10
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 30%
Program ended with exit code: 0
```



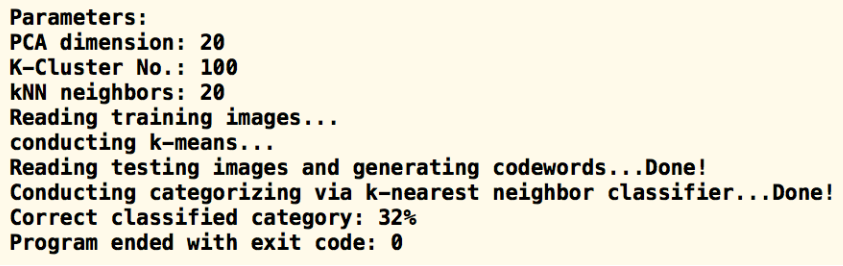
```
Parameters:
PCA dimension: 20
K-Cluster No.: 125
kNN neighbors: 10
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 40%
Program ended with exit code: 0
```



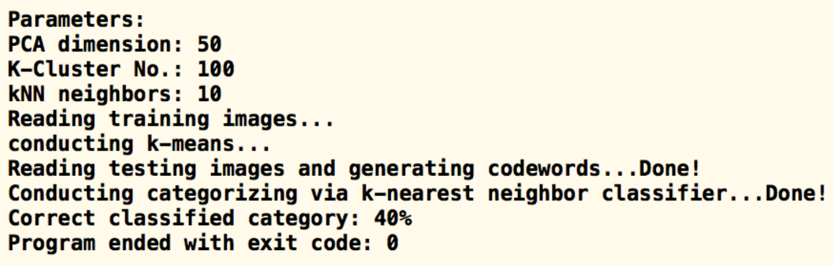
```
Parameters:
PCA dimension: 20
K-Cluster No.: 150
kNN neighbors: 10
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 36%
Program ended with exit code: 0
```



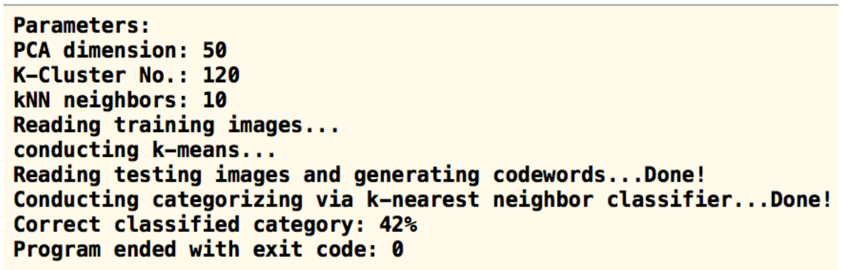
```
Parameters:
PCA dimension: 20
K-Cluster No.: 100
kNN neighbors: 15
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 30%
Program ended with exit code: 0
```



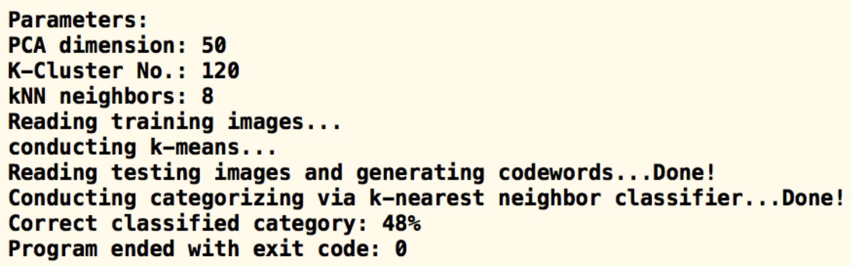
```
Parameters:
PCA dimension: 20
K-Cluster No.: 100
kNN neighbors: 20
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 32%
Program ended with exit code: 0
```



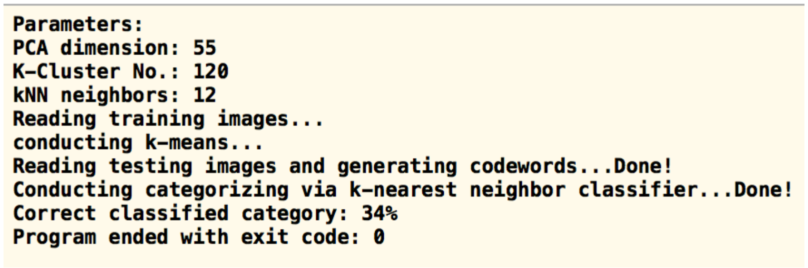
```
Parameters:
PCA dimension: 50
K-Cluster No.: 100
kNN neighbors: 10
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 40%
Program ended with exit code: 0
```



```
Parameters:
PCA dimension: 50
K-Cluster No.: 120
kNN neighbors: 10
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 42%
Program ended with exit code: 0
```

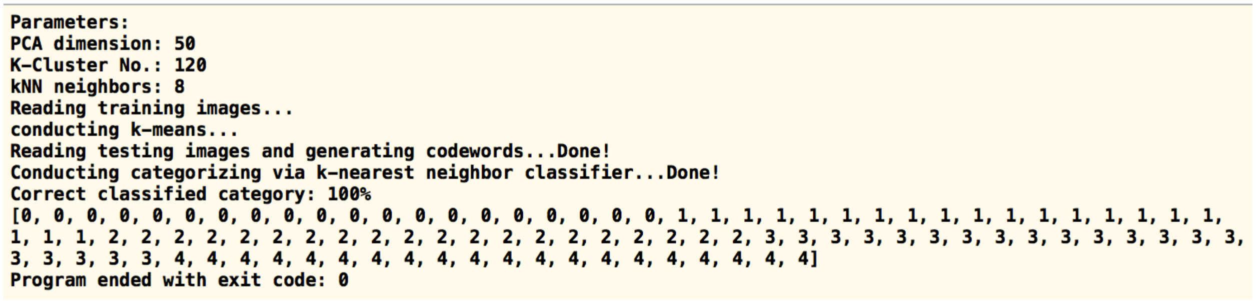


```
Parameters:
PCA dimension: 50
K-Cluster No.: 120
kNN neighbors: 8
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 48%
Program ended with exit code: 0
```



```
Parameters:
PCA dimension: 55
K-Cluster No.: 120
kNN neighbors: 12
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 34%
Program ended with exit code: 0
```

Use train images as test images, the result:



```
Parameters:
PCA dimension: 50
K-Cluster No.: 120
kNN neighbors: 8
Reading training images...
conducting k-means...
Reading testing images and generating codewords...Done!
Conducting categorizing via k-nearest neighbor classifier...Done!
Correct classified category: 100%
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]
Program ended with exit code: 0
```

IV Discussion

If use train images as test images, with most combinations of parameters, the correct rate is 100%. This shows the implementation should be correct.

From the classification results provided above, a 48% correct rate can be achieved for classifying 5 categories with 100 images and 50 test images, by setting feature dimension=50, clusters=120, kNN=8. This rate is the highest one for several trials for different parameters. From the observation, increasing each parameter individually will increase the correct rate. However, there are peaks for each parameter, if the parameters are too large, the predication will be less accurate, even with correct rate lower than 30%. Hence, the parameters should be modified manually several times to achieve better performance, which is the shortcoming of this implementation.

Based on the results, car-side, butterfly and faces can be classified more correctly than the other two categories. The prediction result contains small portion of label 3 and 4. This should not be the problem of the program implementation, and one reason is the k-nearest-neighbor classifier is linear classifier, and these two categories' feature can be non-linear and all mixed together with other categories' feature. On the other hand, better results can be achieved by applying other classifiers like probability approach and SVM.

[i] Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004, May). Visual categorization with bags of keypoints. In Workshop on statistical learning in computer vision, ECCV (Vol. 1, No. 1-22, pp. 1-2).