Image Classification

Name: Wei-Chong Chen Department: ECE Student ID: S01276647

1 Bag of Features Classification with SIFT Descriptors

1.1 Algorithm Implementation

1. Use SIFT to find features (and their descriptors) in all of the training set images. Do this carefully (see Matlab's struct data-type), so that you can easily identify which features belonged to which training class.

First, I use struct to assign different folders based on the training classes into structure. And use vl_sift to extract the features. After getting the features, I assign them into struct again to make life easy.

2. Cluster all the SIFT feature descriptors you found using a distance metric of your choice (be sure to include your chosen metric in your report along with your rationale).

Here I find VLFeat packages include k-means package, therefore, I use its own package instead. The way I choose the parameters are basically considering faster requirement. Thus, Elkan and plusplus are chosen.

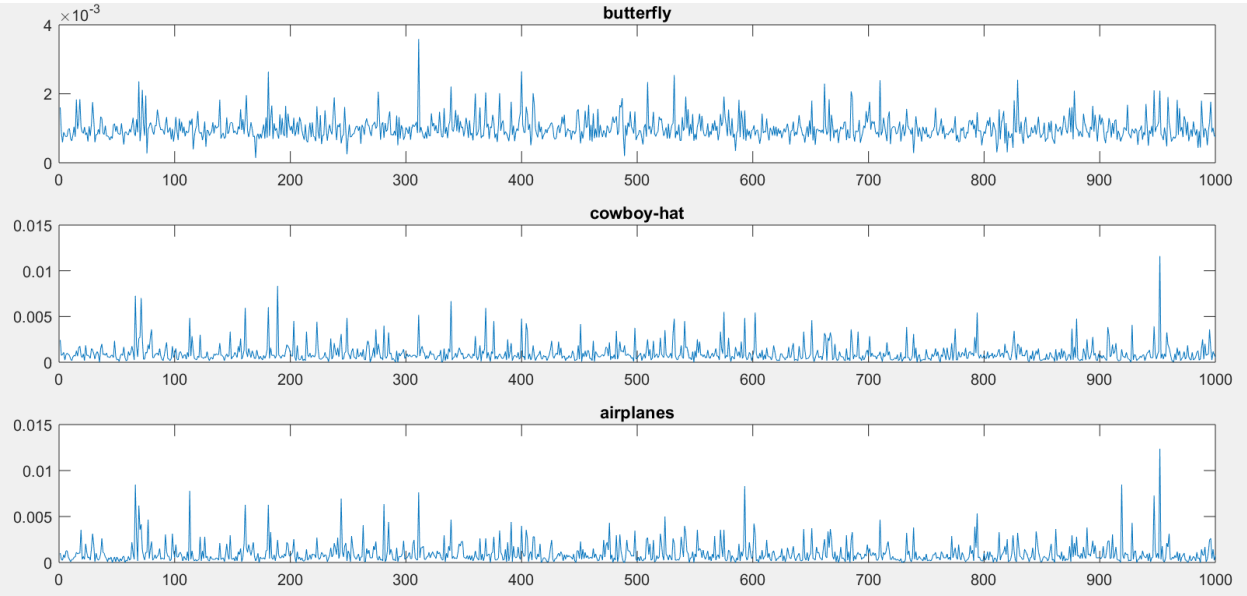
```
vl_kmeans(single(Dname),N, 'algorithm', 'elkan', 'Initialization', 'plusplus');
```

3.

I use the suggested knnsearch package to get a distance metric. With the distance metric, I first calculate the histogram of distance with 15 divisions. Later, I remove the last two divisions to eliminate the outliers.

```
knnsearch(cluster{1,1}',Dname1','distance','euclidean');
[ftshist, binpos] = hist(Dist,15);
index(Dist > binpos(end-1)) = [];
```

After thresholding, I keep the corresponding value to the histogram and normalize it.



4.-7.
Follow the similar steps with the same parameters and settings to test the TestDataset.

1.2 Technical Write-up: Results and Discussion

```
0.3000    0.3000    0.4000
     0    1.0000         0
0.1250         0    0.8750
```

The above matrix is a snapshot from matlab command window which I re-arranged it in the following table.
It's not surprised to find that the result of hat is not good, because the pattern of hat is not recognizable in many ways. And because hat and airplane share some similar peaks, the error will occur from time to time.

	Predicted class			
Actual class	**Classes**	**Hat**	**Butterfly**	**Airplane**
	Hat	**30%**	**30%**	**40%**
	Butterfly	**0**	**100%**	**0**
	Airplane	**12.5%**	**0**	**87.5%**

Since the algorithm is time consuming with high accuracy. The way to improve the algorithm is to find a better algorithm which can deal with N=1000 more efficiently.

1.3 Competition
The mean accuracy is 72.5% according to the table.

2 Can you fix it? Yes you can!
2.2 Technical Write-up: Results and Discussion
As shown in https://pdfs.semanticscholar.org/b00c/19f4c596f6b99c34ec6c612adcc61d4e6b53.pdf table 1, SURF can outperform SIFT and PCA-SIFT since SURF takes much less time than the other two while conserve high accuracy compared to PCA-SIFT. And from https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf, ORB provides less time with high accuracy compared to SIFT yet defeated by SURF.
Therefore, I use Speeded Up Robust Features (SURF) to make the featuring more efficiently. I use SURF to replace the function of SIFT. There are three number of classifications that I use, N=1000, 3000, and 5000. And I find that although the number of classifications increases, the accuracy does not. This observation is because as the number of classification becomes too large, the classifier has to take care too many unnecessary parts, inducing low accuracy.
The averages are shown in the table.

	N=1000	N=3000	N=5000
Accuracy	22.62%	25.20%	24.91%

The detailed accuracy table is in Excel file "Table_25items."

VF-SIFT can provide less time yet with sacrifice of accuracy. As for Vocabulary Tree Clustering, with the aid of k-means, the clustering accuracy is high while consuming more time.

The 12x12 table is in Excel file "Table_12items." And I use N=3000, the mean accuracy is 24.29%.

2.3 Competition
12x12 : 24.29%
25x25: 25.20%

Grad Credits: Support Vector Machines for Image Classification

3.1 Reading

This paper mainly discusses the SVM technique from basic idea to high dimensionality formula. First, it considers only separable case where is a fundamental yet easier case. Second, it changes the problem to non-separable case where introduces penalty to the case. And then, it goes to a more general case which uses symmetric function K or kernel to take care of the formula. To cope with the problem from n-dimensional signal, it uses n transition histogram (one for each axe). Or if we want the result to be more sophisticated, we should compute the transition histogram of the norm of the gradient. The paper also mentioned that to reduce dimension, PCA is a good way to think about.

The experimental results show that the proposed SVM (RBF) can provide high accuracy. One interesting finding is that as sigma goes to infinity, the performance of proposed SVM will converge to linear one, and either the formula or the experimental result shows the same thing. And if the kernel is based on chi-square function, the performance will outperform RBF most of the time, especially on 64 bins histogram. It also shows that with the aid of PCA, the dimension can be decreased while having better accuracy when number of principal components grows except for the polynomial kernel case. And the result shows that transition histogram can improve the performance which corresponding to the findings. In 5.2.2, we can find that KNN performs well with K=1, nonetheless, chi-square based SVM outperforms it totally.

The paper not only addresses the basic idea of SVM but also proposes a good kernel based SVM, resulting in better performance. I think it should be good to replace PCA to another metric since from the discussion of Problem 2, we can find that PCA is not a perfectly efficient metric.

3.2 Train SVM on 3 class dataset

I first use the package provided by Matlab to do the classification training.

```
path = imageDatastore('./Assignment06_data_reduced/TrainingDataset/', 'IncludeSubfolders', true, '
classifier = trainImageCategoryClassifier(path,bagOfFeatures(path));
```

And use evaluate function provided by Matlab to test them. To better collect the data, I put folder TestDataset_1, TestDataset_2, and TestDataset_3 into one folder.

```
pathtest = imageDatastore('./Assignment06_data_reduced/TEST/', 'IncludeSubfolders', true, 'labels
evaluate(classifier,pathtest);
```

And the result is shown below:

```
                PREDICTED
KNOWN    | 024    051    251
--------------------------------
024      | 1.00   0.00   0.00
051      | 0.40   0.50   0.10
251      | 0.00   0.00   1.00

* Average Accuracy is 0.83.
```

3.3 Competition

Mean accuracy : 0.83