Homework Assignment 5, CS696, Applied Computer Vision

Hsuan Yu, Liu 823327369

**My algorithm and decision,**

At beginning, the flower images are not same sizes, so I cropped those images into 500x450 images from centroid. Moreover, I developed the classifiers into three respectively scripts, main\_HOC\_SVM.m, main\_HOG\_SVM.m, and main\_SIFT\_SVM.m.

Three types of features:

1. Histogram of Color: (In main\_HOC\_SVM.m)

An image has RGB layers, so I computed three histograms for each layer and normalized each histogram to unit. Also, I used 30 bins for each layer. Thus, each image has 90 features.

2. Histogram of Gradient: (In main\_HOG\_SVM.m)

Before extracting features, I converted color to grayscale. After converted, I used MATLAB built-in function, extractHOGFeatures, to get HOG features.

3. SIFT-like: (In main\_SIFT\_SVM.m)

It contains three other functions, get\_interest\_points, get\_features, and buildBagOfVisualWord. First two functions are the same functions in last assignment, Assignment 4. That is because each image gets more than one features, so I have to combine those features. Thus, I applied Bag of Visual Word model to transform the features.

At beginning, I would get all the descriptors of train data and use K mean to cluster in the number of features for SVM training. After got categories, using a center descriptor in each category represents the category. Secondly, after got a codewords, used it to compute the histogram from codewords. The histogram is a 1x(Number of words) vector, and it’s a features for SVM. I used 500 to be the number of words.

Some setting of SIFT:

1. features size = [16, 16]

2. K= 0.06

2. Interesting Point: A point whose Harris corner score is larger than 0.005\*(The highest score)

**Results of three type of features:**

I record 3 times experiments for each of type.

1. Histogram of Color:

1st time: Bin size=30

Accuracy: 0. 588889

Failure examples:

GT=3 Predict=2 GT=1 Predict=3 GT=2 Predict=3

2nd time: Bin size=30

Accuracy: 0. 666667

Failure examples:

GT=2 Predict=3 GT=3 Predict=2 GT=1 Predict=2

3rd time: Bin size=50

Accuracy: 0.633333

Failure examples:

GT=1 Predict=3 GT=3 Predict=2 GT=2 Predict=3

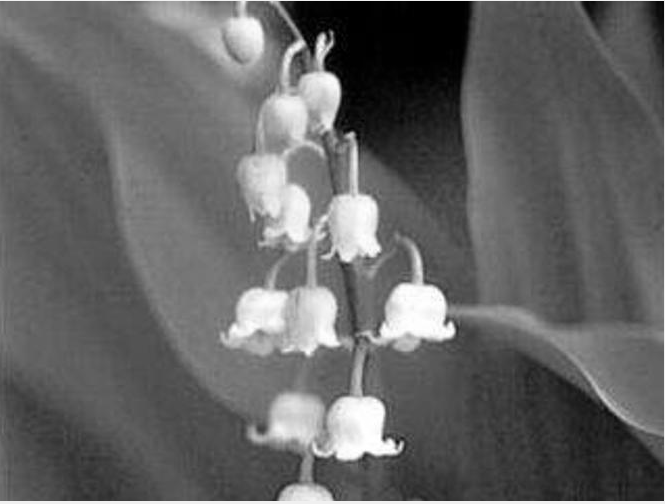
2. Histogram of Gradient:

1st time:

Accuracy: 0.7

Failure examples:

GT=3 Predict=1 GT=2 Predict=3 GT=3 Predict=2

2nd time:

Accuracy: 0. 688889

Failure examples:

GT=2 Predict=1 GT=3 Predict=1 GT=1 Predict=2

3rd time

Accuracy: 0. 766667

Failure examples:

GT=2 Predict=3 GT=1 Predict=2 GT=3 Predict=1

3. SIFT-like:

1st time:

Accuracy: 0. 344444

Failure examples:

GT=1 Predict=3 GT=1 Predict=3 GT=2 Predict=3

2nd time:

Accuracy: 0. 455556

Failure examples:

GT=1 Predict=3 GT=2 Predict=1 GT=2 Predict=3

3rd time

Accuracy: 0. 511111

Failure examples:

GT=2 Predict=3 GT=3 Predict=1 GT=1 Predict=3

**  **

**Conclusion:**

According to my experiments, the highest is using histogram of gradient. The accuracy can approach to 77%. The lowest accuracy is using SIFT-like. Its accuracy is between 30%~50%. Moreover, the SIFT-like required more computing time than the others, because it has to build a codework. If I take more interesting points, the computing time increases as well. In my code, a total features are around 60K ~ 80K. Another parameter which affects cost is number of words. If the number is too small, it might not have enough features for training. I tested 300 and 500. The 500 is better than 300.

HOC are affected by color, so the images containing similar color may classify into the same category. Therefore, flower2 and flower3 are blurred. If we focus on only one flower, the accuracy will be lower.

HOG is the best type feature of those, but there are the greatest number of features in one image.

Each image has more than 20K features.