Summary

1) Software Used: -Python with lib such as numpy ,opencv,sklearn and scipy

2) Feature Extraction Process (*) :- I choose to use SIFT with the Dense interest points. use of Dense SIFT, which is prove to give better result because lots of information is been preserved. Other option are Harris corner detector, SURF (faster but not good quality result)..etc.

3) Similarity/Distance Measures (*) :- K-mean clustering algorithm(minibatch for efficency).Euclidean distance.

3) Classifier (if you have used any standard classifier):- SVM (linearSVM with pyramid match kernel)

4) References (if any) :- http://cs.brown.edu/courses/cs143/results/proj3/psangkloy/ , https://www.youtube.com/watch?v=iGZpJZhqEME

Algorithm :(Bag-Of-Features)

1) Feature extraction: - I choose to use SIFT with SIFT interest point ,not the Dense SIFT. We can also use Dense SIFT which is prove to give better result because lots of information is been preserved. Other option are Harris corner detector, SURF (faster but not good quality result)..etc. Latter I also experimented with Dense Shift .

2)Computing image descriptor: - The SIFT descriptor proposed by Lowe (1999, 2004) can be seen as a position-dependent histogram of local gradient directions around the interest point. The SIFT descriptor is invariant to translations, rotations and scaling transformations in the image domain and robust to moderate perspective transformations and illumination variations. Experimentally, the SIFT descriptor has been proven to be very useful in practice for image matching and object recognition under real-world conditions. The different type of descriptor are OpponentSIFT (which consider RGB channel where as SIFT use grayscale),PCA SIFT (by reducing dimensionality).

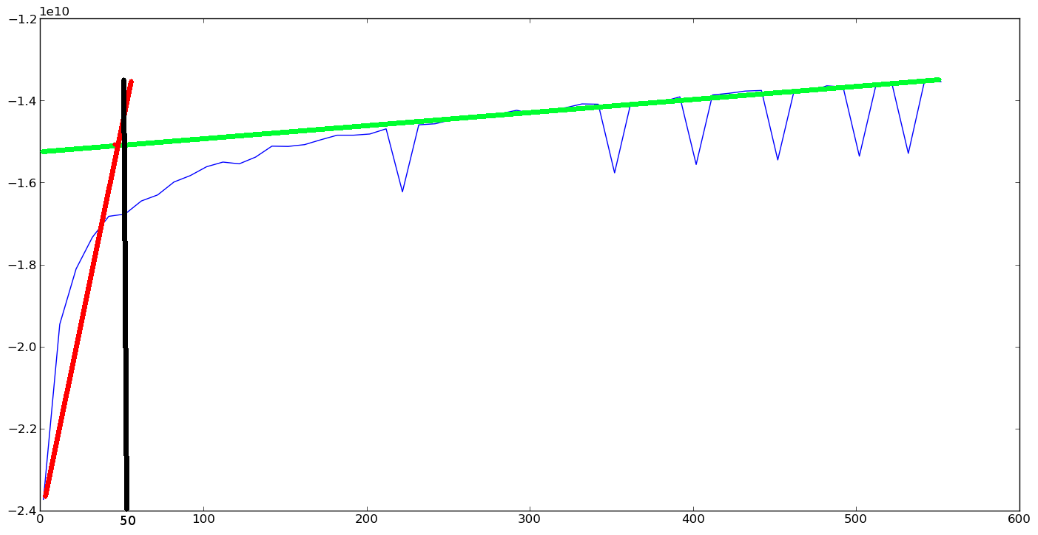
3)Visual words or Visual Vocabulary: - I took the descriptor from all the images and performed K-Mean Clustering (mini batch k-mean which is faster) .There centroid or codebook is called as Visual words .

4)Then find the histogram or distribution of each image descriptor over the visual word. Then store this histogram with label .This histogram bins form the feature vector.

5)Classification: - I used LinerSVM or SVM classifier which is efficient and provide good result. We could have used nearest neighbor classifier which does not perform well ..etc.

Performance Improvement

1)choice of K value for clustering using K-mean. I computed score for each cluster(k=10,20....in step of 10). Used Elbow method for finding number of clusters. So I found K~ 50 .This gave a accuracy of ~60% on old validation set.



2)I used the same parameters in order to easily compare the accuracy. improvement I made is by adding spatial features. From Lazebnik et al 2006, the spatial information can be preserved instead of thrown away with bag of words representation. I used L = 2. since it seems to have the best accuracy for strong features according to the experiment in the paper.
I got accuracy of ~80%.

3)I'll just increase the vocab size from 50 to 400 (k=400) with pyramid match kernel, and now the accuracy increase to ~86%. With a cost of run time a LOT more.

4)I used Dense feature. This is much memory consuming task but better result. I observed that the accuracy was 88% with cluster size of 50 (k=50) and used pyramid match kernel.

5)I used Dense feature. This is much memory consuming task but better result. I observed that the accuracy was 90% with cluster size of 400 (k=400) and used pyramid match kernel.

```
             precision    recall  f1-score   support

          1       0.93      0.86      0.89       200
          2       0.90      0.91      0.91       200
          3       0.86      0.84      0.85       200
          4       0.86      0.97      0.91       200
          5       0.95      0.91      0.93       200

avg / total       0.90      0.90      0.90      1000
```

from above observation I conclude that we need to make trade of between memory,speed and accuracy .

Experiment Result

Some of the output show, with different command to see above observation

for observation [1] as above

(note that the Validation folder should contain label for testing with file name labels_valid.txt , also change the line 81 in learn.py to computeHistrogram as shown in comment and change the line 64 in test.py to computeHistrogram as shown in comment)

```
$python learn.py -d ./Train/ -n 50
```

output

```
-----------------------------------------------------
label are:['Images']
-----------------------------------------------------
Generation SIFT words
-----------------------------------------------------
No masks avilable
-----------------------------------------------------
the number of words:144947
-----------------------------------------------------
Generation vocabulary or visual words using clustering
-----------------------------------------------------
number of cluster=50
-----------------------------------------------------
Coumputing Histrogram from centroids
-----------------------------------------------------
The number of feature:50
-----------------------------------------------------
write the histogram.out
```

```
------------------------------------------------------
Then to test on dataset using test script
```

```
$python test.py -d ./Validation/ -c ./Train/CENTROID.file
```

```
output

./Train/CENTROID.file
------------------------------------------------------
Generation SIFT words
------------------------------------------------------
No masks avilable
------------------------------------------------------
Using trained vocabulary or visual words
------------------------------------------------------
Coumputing Histrogram from centroids
------------------------------------------------------
write the histogram.out
------------------------------------------------------
```

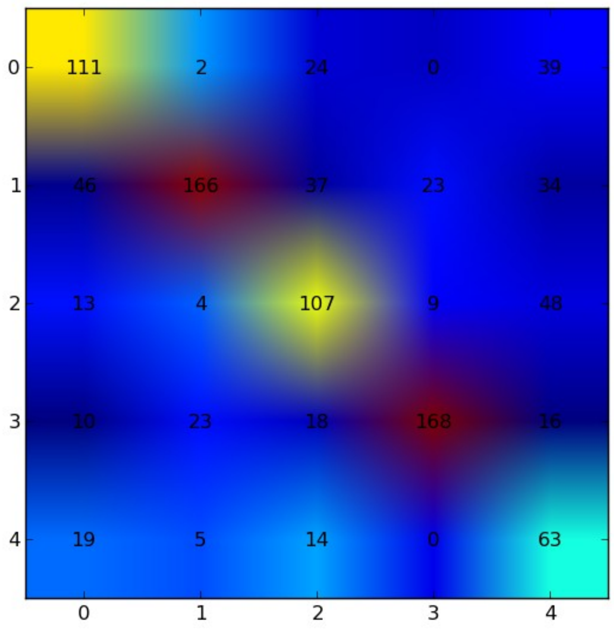
Then to classify

```
$python classify.py
```

```
             precision    recall  f1-score   support

          1       0.63      0.56      0.59       199
          2       0.54      0.83      0.66       200
          3       0.59      0.54      0.56       200
          4       0.71      0.84      0.77       200
          5       0.62      0.32      0.42       200

avg / total       0.62      0.62      0.60       999
```



for observation [2] as above
(note that the Validation folder should contain label for testing with file name labels_valid.txt , also change the line 81 in learn.py to computeHistrogramByLevel as shown in comment and change the line 64 in test.py to computeHistrogramByLevel as shown in comment)

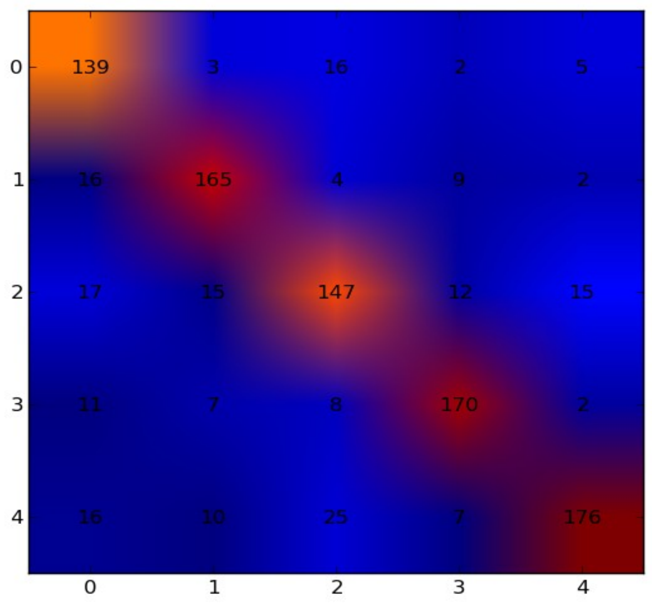
```
$python learn.py -d ./Train/ -n 50
----------------------------------------------------
label are:['Images']
----------------------------------------------------
Generation SIFT words
----------------------------------------------------
No masks avilable
----------------------------------------------------
the number of words:144947
----------------------------------------------------
Generation vocabulary or visual words using clustering
----------------------------------------------------
number of cluster=50
----------------------------------------------------
Coumputing Histrogram from centroids
----------------------------------------------------
The number of feature:1050
----------------------------------------------------
write the histogram.out
----------------------------------------------------
```

```
$python test.py -d ./Validation/ -c ./Train/CENTROID.file
----------------------------------------------------
Generation SIFT words
----------------------------------------------------
No masks avilable
----------------------------------------------------
Using trained vocabulary or visual words
----------------------------------------------------
Coumputing Histrogram from centroids
----------------------------------------------------
write the histogram.out
----------------------------------------------------
```

```
$python classify.py
    precision    recall  f1-score   support

          1       0.84      0.70      0.76       199
          2       0.84      0.82      0.83       200
          3       0.71      0.73      0.72       200
          4       0.86      0.85      0.85       200
          5       0.75      0.88      0.81       200

avg / total       0.80      0.80      0.80       999
```



for observation [3] as above

(note that the Validation folder should contain label for testing with file name labels_valid.txt , also change the line 81 in learn.py to computeHistrogramByLevel as shown in comment and change the line 64 in test.py to computeHistrogramByLevel as shown in comment)

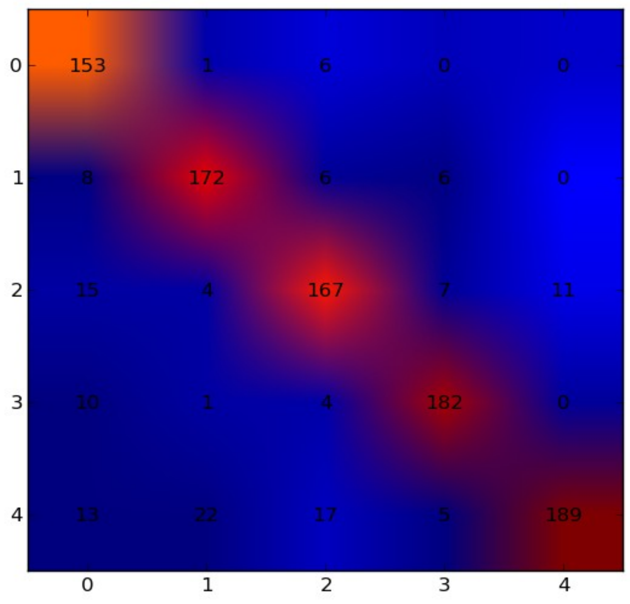
```
$python learn.py -d ./Train/ -n 400
-----------------------------------------------------
label are:['Images']
-----------------------------------------------------
Generation SIFT words
-----------------------------------------------------
No masks avilable
-----------------------------------------------------
the number of words:144947
-----------------------------------------------------
Generation vocabulary or visual words using clustering
-----------------------------------------------------
number of cluster=400
-----------------------------------------------------
Coumputing Histrogram from centroids
-----------------------------------------------------
The number of feature:8400
-----------------------------------------------------
write the histogram.out
-----------------------------------------------------
```

```
$python test.py -d ./Validation/ -c ./Train/CENTROID.file
-----------------------------------------------------
Generation SIFT words
-----------------------------------------------------
No masks avilable
-----------------------------------------------------
Using trained vocabulary or visual words
-----------------------------------------------------
Coumputing Histrogram from centroids
-----------------------------------------------------
write the histogram.out
-----------------------------------------------------
```

```
$python classify.py
precision    recall  f1-score   support

        1       0.96      0.77      0.85       199
        2       0.90      0.86      0.88       200
        3       0.82      0.83      0.83       200
        4       0.92      0.91      0.92       200
        5       0.77      0.94      0.85       200

avg / total       0.87      0.86      0.86       999
```



!!!WARNING!!! Note that to compute dense SIFT memory consumend is lot and time consuming so please test it on high configuration system.

for observation [4] as above

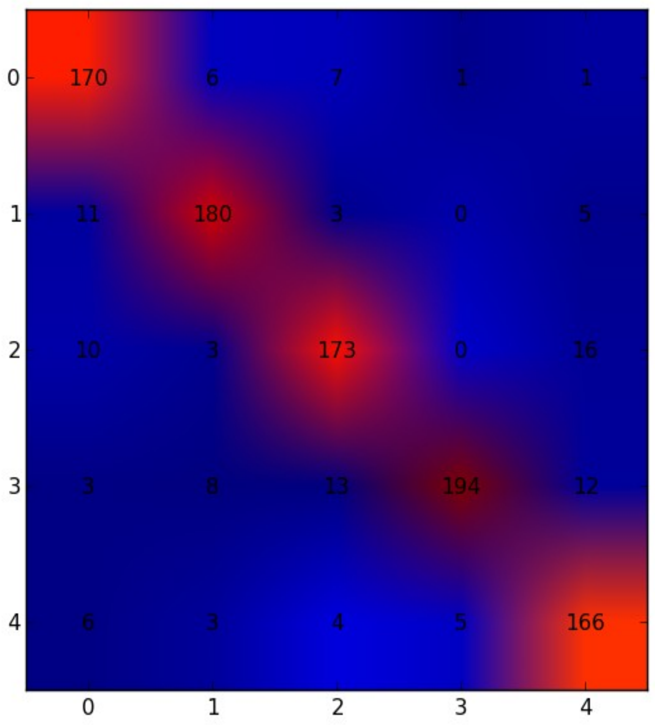
```
$python learn.py -d ./Train/ -n 50 Dense
------------------------------------------------------
label are:['Images']
------------------------------------------------------
Generation SIFT words
------------------------------------------------------
No masks avilable
------------------------------------------------------
the number of words:800000
------------------------------------------------------
Generation vocabulary or visual words using clustering
------------------------------------------------------
number of cluster=50
------------------------------------------------------
Coumputing Histrogram from centroids
------------------------------------------------------
The number of feature:1050
------------------------------------------------------
write the histogram.out
------------------------------------------------------
```

```
$python test.py -d ./Validation/ -c ./Train/CENTROID.file Dense
------------------------------------------------------
Generation SIFT words
------------------------------------------------------
No masks avilable
------------------------------------------------------
Using trained vocabulary or visual words
------------------------------------------------------
Coumputing Histrogram from centroids
------------------------------------------------------
write the histogram.out
------------------------------------------------------
```

```
$python classify.py
precision    recall  f1-score   support

        1      0.92      0.85      0.88       200
        2      0.90      0.90      0.90       200
        3      0.86      0.86      0.86       200
        4      0.84      0.97      0.90       200
        5      0.90      0.83      0.86       200

avg / total       0.89      0.88      0.88      1000
```



for observation [5] as above

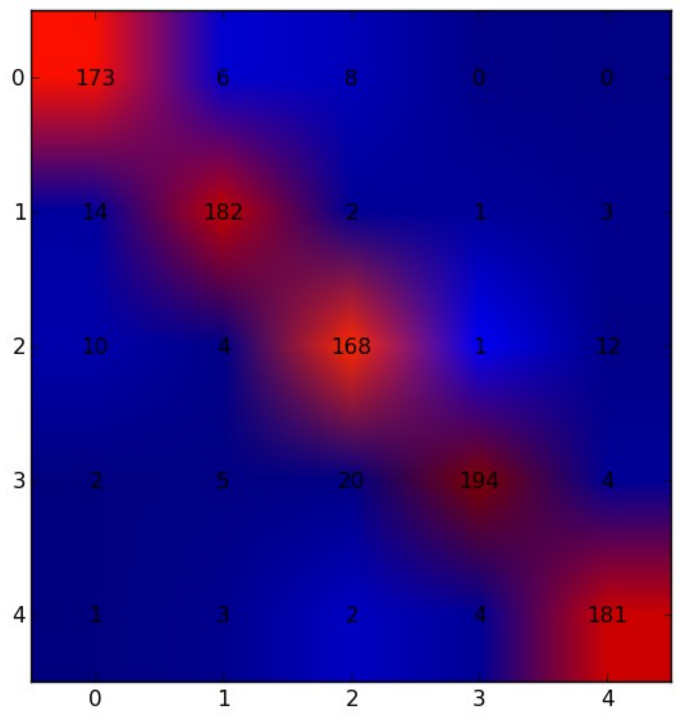
```
$pthon learn.py -d ./Train/ -n 400 Dense
-----------------------------------------------------
label are:['Images']
-----------------------------------------------------
Generation SIFT words
-----------------------------------------------------
No masks avilable
-----------------------------------------------------
the number of words:800000
-----------------------------------------------------
Generation vocabulary or visual words using clustering
-----------------------------------------------------
number of cluster=400
-----------------------------------------------------
Coumputing Histrogram from centroids
-----------------------------------------------------
The number of feature:8400
-----------------------------------------------------
write the histogram.out
-----------------------------------------------------
```

```
$python test.py -d ./Validation/ -c ./Train/CENTROID.file Dense
------------------------------------------------------
Generation SIFT words
------------------------------------------------------
Using trained vocabulary or visual words
------------------------------------------------------
Coumputing Histrogram from centroids
------------------------------------------------------
write the histogram.out
------------------------------------------------------
```

```
$python classify.py
precision    recall  f1-score   support

          1     0.93      0.86      0.89      200
          2     0.90      0.91      0.91      200
          3     0.86      0.84      0.85      200
          4     0.86      0.97      0.91      200
          5     0.95      0.91      0.93      200

avg / total       0.90      0.90      0.90      1000
```



Prediction Algorithm

Inputs:

- L, a learner (training algorithm for binary classifiers)
- samples X
- labels y where $y_i \in \{1, \dots K\}$ is the label for the sample X_i

Output:

- a list of classifiers f_k for $k \in \{1, \dots K\}$

Procedure:

- For each k in $\{1 \dots K\}$:
    - Construct a new label vector $y_i' = 1$ where $y_i = k$, 0 (or -1) elsewhere
    - Apply L to X, y' to obtain f_k

Making decisions proceeds by applying all classifiers to an unseen sample x and predicting the label k for which the corresponding classifier reports the highest confidence score:

$$\hat{y} = \arg \max_{k \in 1 \dots K} f_k(x)$$

Why do you think your algorithm got the accuracy that it did on the validation data?

1) My algorithm failed to filter common background noise.

2)The validation dataset had image take during night which are not found in training dataset.

3)There was addition of confusion information within image. eg a leg with sleeper etc.

Scope of Improvement

1)By developing SIFT which is invarient of light or improving the image the fix probelm of illumination.

2)using partially location dependent feature.

3)some way of background substraction to remove unwanted SIFT feature.eg by using IDF score or stopwords.

To Run the test on test dataset.

```
$pthon learn.py -d ./Train/ -n 400 Dense

$python testOnTestD.py -d ./Test/ -c ./Train/CENTROID.file Dense

$python classifyOnTest.py
```