CS172 Computer Vision I:
Image Classification with BoW, SPM and SVM

Yang Hongdi
2019533234
yanghd@shanghaitech.edu.cn

Abstract

This project contains two models using BoW + SVM and BoW + SPM + SVM to do Image Classification. For BoW (Bag of Words), SIFT descriptors are used for feature extraction. K-means is used to generate codebook. Feature quantization and average pooling is performed to generate the histogram. SPM is implemented by the author. Linear SVM is choosed to do the classification. All training data and test data are based on Caltech256 dataset.

1. Introduction

In this project, Python is selected to be the language in coding. For SIFT descriptors extraction, opencvpython library is used. For K-means, to reduce program running time, sklearn.cluster.MiniBatchKMeans is used. For SVM, only linear SVM is performed in this project, and sklearn.svm.LinearSVC is used. SPM is implemented by the author. In Caltech256 dataset, 257-clutter is ignored as the data inside is strange.

2. Algorithm Description

2.1. SIFT feature extraction

SIFT descriptor has already been introduced in previous projects, so in this project it will not be detaily explained. It can be treated as some 128-dimension vectors to describe the features of the image.

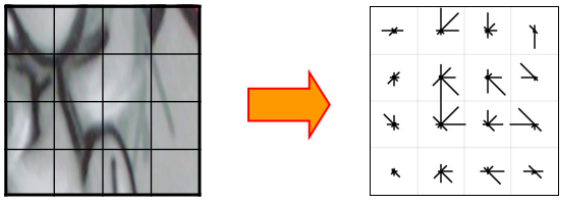

Figure 1. SIFT 128-d descriptor

2.2. K-means

K-means clustering is a simple and elegant approach for partitioning a data set into K distinct, nonoverlapping clusters. For each cluster, we want to minimize sum of squared Euclidean distances between points x_i and their nearest cluster centers m_k.

$$D(X, M) = \sum_{cluster\ k} \sum_{point\ i\ in\ clusterk} (x_i - m_k)^2$$

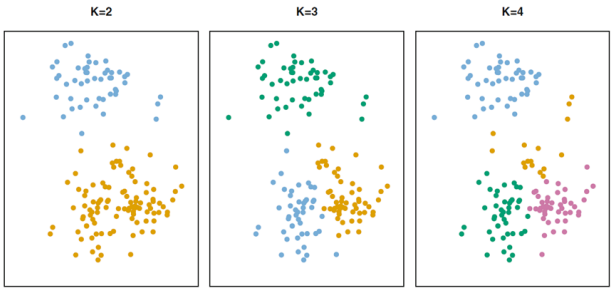

Figure 2. same data with different k

To implement the algorithm, we have 2 steps :

- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each data point to the nearest center

• Recompute each cluster center as the mean of all points assigned to it

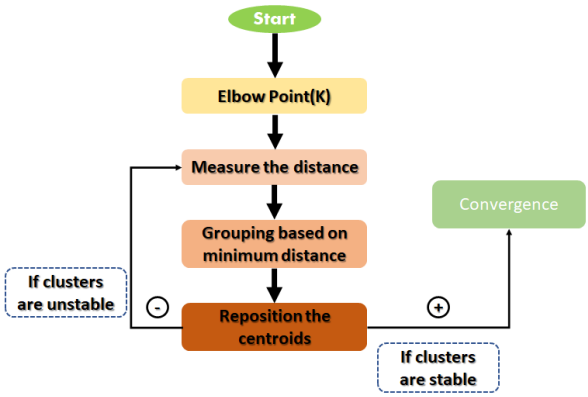

Figure 3. K-means Algorithm Implementation

2.3. Bag of Words

In computer vision, a bag of visual words is a vector of occurrence counts of a vocabulary of local image features. In this project, visual words is represent by SIFT descriptors, so each word is a 128-d vector.
To implement Bag of Words Model, we mainly have 4 steps :

1. Extract local features (by SIFT described in 2.1)
2. Learn "visual vocabulary" (by K-means described in 2.2)
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of "visual words"

After all the steps, we will have a histogram for each image representing the frequencies of its "visual words". The number of "visual words" will be defined by the number of clusters we choose in K-means.

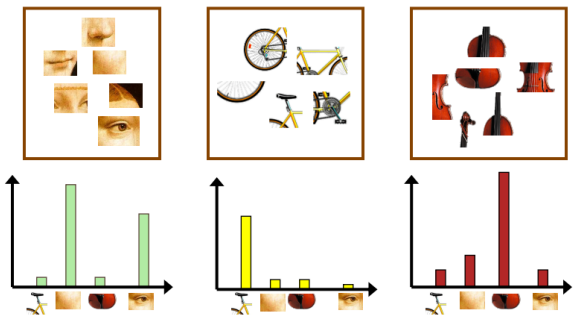

Figure 4. final histogram of BoW

2.4. Support Vector Machine

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.
In SVM, we find hyperplane that maximizes the margin between the positive and negative examples. i.e for separable data, we find

$$\min_{w,b} \frac{1}{2}||w||^2 \quad subject\ to\ y_i(w \cdot x_i + b) \geq 1$$

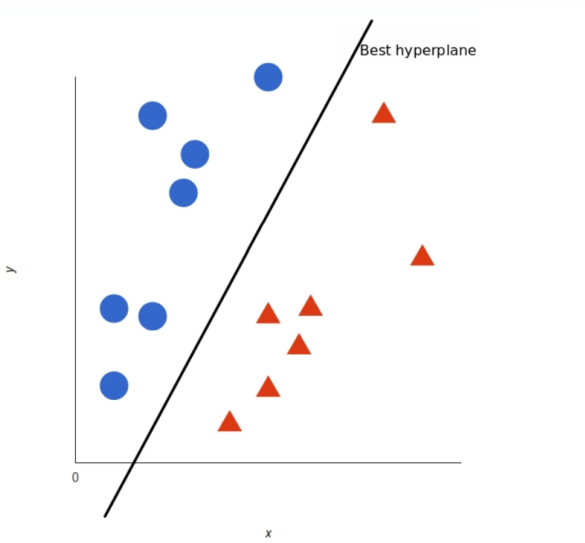

Figure 5. separable data with svm

while for non-separable data, we can calculate

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(w \cdot x_i + b))$$

or just use nonlinear SVM to map the data to a higher dimensional space.
Due to machine hardware restriction and time limitation, only linear kernel is performed in this project.

2.5. Spatial Pyramid Matching

For SPM, The general idea is to divide the image into a small number of cells, and concatenate the histogram of each of these cells to the histogram of the original image, with a suitable weight. In this project, the weight is choosed to be $2^{current\ level\ -level\ num}$

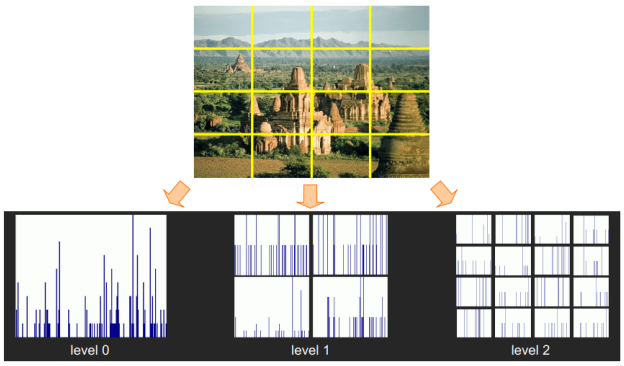

Figure 6. a 2-level spm

So if SPM is performed, we will train SVM using the concatenated histogram instead of a single histogram predicted by BoW. As more features are performed in SPM model, it is beileved to produce a better result than simple BoW, however, with more execution time. Due to machine hardware restriction and time limitation, only 1-level spm is performed in this project.

3. Problems Encountered

3.1. Time

At first, sklearn.cluster.KMeans is directly used with cluster num = 1000 and max iter = 300. However, with these parameters the program execution time can be exetremly long, calculating samples with training size = 15, class size = 256 would take about 2 hours. So sklearn.cluster.MiniBatchKMeans is used, and the KMeans calculating time can be reduced to about half an hour.

3.2. Repeatablity

For each test run, we need to extract features and calculate the Kmeans, which would take a quite long time. So to reduce repeat works, pickle library in Python is used to save the images, labels, descriptors and Kmeans result on disk. And we can directly load them in later test runs.

3.3. Feature detection Failure

In some images, the feature detection would fail. If feature detection failure happens, we would ignore this image and not add it to any dataset. However, in SPM, there will be cases that features can be detected in some areas, while no keypoints will be detected in other areas. To solve this, uniform sample is performed in SPM. Instead of detecting keypoints by SIFT, keypoints are choosed uniformly. This would avoid feature detection failure, but the program execution would also be much longer.

4. Code Running

In file folder ImageClassification, run
python BoW.py
python SPM.py

5. Result Analysis

5.1. Start with small number of classes

First, we choose a small number of classes to test the accuracy of our models.
we choose first 16 classes to test. The parameters will be :

- num of clusters : 4 x class size = 64
- Kmeans max iter time : 300
- LinearSVC : C = 0.01 (tolerance of failure) max iter time : 300

The result is :

Table 1. BoW Result of 16 classes

train size	15	30	45	60
accuracy	0.257	0.275	0.283	0.286

Table 2. SPM result of 16 classes

train size	15	30	45	60
accuracy	0.248	0.261	0.303	0.327

With train size increasing, the accuracy is improved. And as train size and class size is low, the SPM will have less accurate result at first. But as train size increasing, the SPM will produce better result than BoW.

5.2. Train with 256 classes

After testing with small size of classes, we then train and test with 256 classes(class 257-clutter is ignored). We start with parameters:

- num of clusters : 4 x class size = 1024
- Kmeans max iter time : 100
- LinearSVC : C = 0.01 (tolerance of failure) max iter time : 300

The result is :

Table 3. BoW Result of 256 classes

train size	15	30	45	60
accuracy	0.0428	0.0575	0.0662	0.0748

Table 4. SPM Result of 256 classes

train size	15	30	45	60
accuracy	0.0656	0.0906	0.1059	0.1133

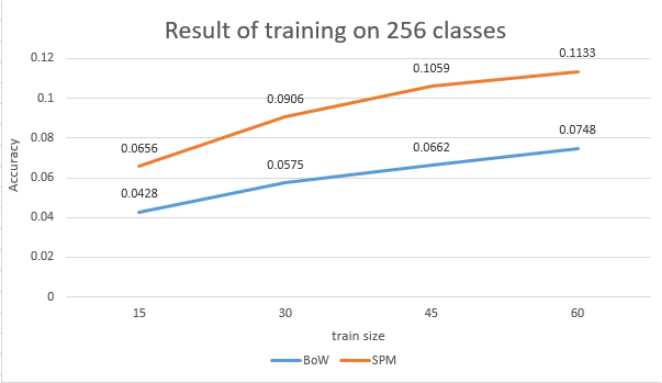

Figure 7. Result of training on 256 classes

According to the graph, we can find that SPM model has a significant better accuracy than BoW. And for both models, the accuracy increases as train size increases, but the increasing rate is decreasing.

5.3. Experiment with parameters

As the result accuracy is quite low with parameters in previous section, more parameters are tried.
First, Kmeans max iter time is increased to 300.

Table 5. BoW Result of 256 classes

train size	15	30	45	60
accuracy	0.0413	0.0567	0.0667	0.0722

Table 6. SPM Result of 256 classes

train size	15	30	45	60
accuracy	0.0653	0.0912	0.1057	0.1132

We can find that the result is almost the same with max iter num = 100.
Then we try :

- num of clusters : 8 x class size = 2048
- Kmeans max iter time : 300
- LinearSVC : C = 0.01 (tolerance of failure) max iter time : 1000

The Result is :

Table 7. BoW Result of 256 classes

train size	15	30	45	60
accuracy	0.0406	0.0516	0.0572	0.0648

Table 8. SPM Result of 256 classes

train size	15	30	45	60
accuracy	0.0624	0.0808	0.0957	0.1027

We find that with more clusters, the accuracy actually decreases. It is believed that more clusters causes overfitting.

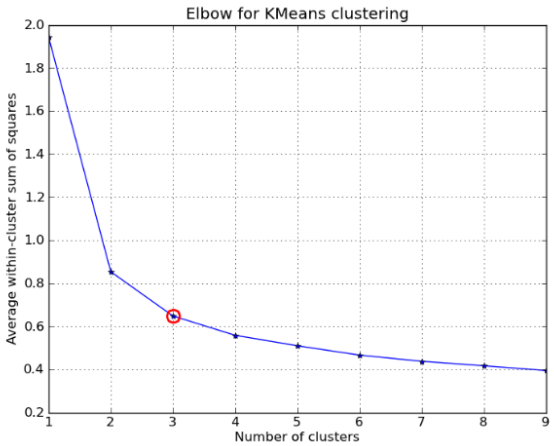

Figure 8. Elbow for Kmeans clustering

As the figure shows, with num of clusters increases, the within-cluster sum of squares actually dose not change much. But with more clusters, some non-important features may be considered as a "visual word" and thus cause overfitting, the accuracy decreases.

6. Thoughts about Low Accuracy

When testing with small size of classes, the accuracy can reach about 30%, which is acceptable. However, testing with 256 classes, the accuracy can only reach 11%, which may be too low. Considering 256 classes, some classes might be similar, thus cause the svm data to be non-separable. And in the dataset, there exist some images that are quite strange or can't even detect features. So changing parameters will not be able to improve the accuracy a lot, non-linear svm and higher level of SPM should be considered.

7. References

1. https://towardsmachinelearning.org/k-means/
2. https://en.wikipedia.org/wiki/Bag-of-words_model
3. https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

4. https://slazebni.cs.illinois.edu/publications/pyramid_chapter.pdf