Algorithm and Data Structure Coursework: K-Means Feature for Image Retrieval

Qiwei Feng
2011011250, IIIS-10
Tsinghua University
gdfqw93@163.com

Pufan He
2011011307, IIIS-10
Tsinghua University
hpfdf@126.com

ABSTRACT

This project implements a similar image search algorithm (image retrieval) based on multiclass classification and K-Means feature. Our training phase includes image resizing, image patch extraction, patch sampling, PCA whitening, K-Means for patches, feature extraction and multiclass SVM. We use 218-dimension K-Means and RGB, HSV color moment. The training phase takes no greater than one hour in time, 8GB in memory. Finally we obtained 69.82% accuracy on test data classification.

We have made our work open, and the full project codes can be found at `https://github.com/caiwaifung/lastcourse`.

Keywords

Image Retrieval, Image Classification, SVM, Whitening, K-Means

1. INTRODUCTION

2. IMPLEMENTATION

We built a system to extracting features from images as well as training model and answering queries of finding related images. The whole system is written in MATLAB.

The system contains feature extracting part and SVM training and testing part. In the feature extracting part, we use both features from K-Means method [3], and the RGB and HSV color moments. We use 200-dimension K-Means features and 18-dimension color moment features (9 for RGB and 9 for HSV). The K-Means method contains patch extracting, patch whitening, patch sampling and K-Means clustering, and feature extrating. The color moment method makes use of 3 common color moments for each channel of RGB and HSV colors. At last, we put the features into SVM and make the classification possible. The related image search process is done by simply finding the closest feature in the same catelogy.

The following subsections includes the details of our algorithms.

2.1 Patch Extracting and Sampling

The key idea of K-Means featuring is to find the most common patches in all images, and build features based on that common patches (called centroids). Let w be the size of the centroids (set to 6 in our implementation). For an l-by-m image, we will consider all its sub-images (patches) of size w-by-w; there are totally $(l-w+1)(m-w+1)$ such patches. Each patch can be represented into a vector of length $3w^2 = p$, so one image can be represented into a n-by-p matrix, where $n = (l-w+1)(m-w+1)$. We represent all images into this matrix form. This process is called **patch extracting**.

We need a large amount of sampling patches for the K-Means clustering process. We simply random selected $t \approx 1000000$ patches from all the images' matrices. The sampled patches is a matrix P of size t-by-p.

2.2 Whitening

Directly compute the square distance between two $6 \times 6 \times 3$ patches is not very reasonable. We use the idea of whitening from [3] to transform our patches into normalized 108-length real vector, so that when consider the transformed patch as a random variable, the mean of every entry is 0 and the variance of every entry is 1. Whitening could reduce possible bias in patch component, and returns more robust K-Means result.

This can be done by a PCA transform:

$$X_{\text{wh}} = \text{diag}\left(1/\sqrt{\text{diag}(S)+\epsilon}\right) \times U^T \times (X-\mu), \qquad (1)$$

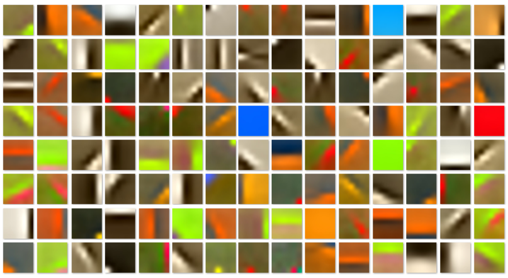
where the covariance matrix SVD decomposite into U, S, V, and μ is the mean X. We use the built-in SVD function in matlab to achieve high efficiency.

2.3 K-Means Clustering

We want $k = 200$ centroids of all sampled patches. We do the K-Means clustering for those patches, and take the final k "means". The final centroids can be represented by a matrix C of size k-by-p.

Without Whitening

With Whitening



2.4 Feature Extracting

Now consider an l-by-m image. We know that it has n patches, so it's an matrix G of size n-by-p. We first calculate the distance between each patch and each centroids, resulting in a distance matrix $D \in \mathbb{R}^{n \times k}$ where D_{ij} is the distance from the i-th patch and the j-th centroid. Then, we normalize each row of D by subtracting its mean and dividing its standard variance. We negate each elements of the normalized D, elimate all negative elements by setting them to zeros, and get the new matrix D^*. Here, D^*_{ij} being large means that the i-th

2.5 Color Moment

2.6 Multiclass SVM

2.7 Related Image Search

3. EXPERIMENTS

3.1 Data Set

Class labels $1 \le C \le 10$:

1. Bird.
2. Insect.
3. Butterfly.
4. Waterwheel.
5. Construction.
6. Piano.
7. Airplane.
8. Wine.
9. Woman.
10. Flower.

3.2 Without Whitening

3.3 With Whitening

3.4 Final Test

The distribution of predicted labels:

```
1-1: 41 |################ 71.93 %
1-2:  3 |#
1-3:  1 |
1-4:  3 |#
1-5:  0 |
1-6:  1 |
1-7:  2 |
1-8:  2 |
1-9:  1 |
1-10: 3 |#

2-1:  7 |##
2-2: 28 |########### 49.12 %
2-3:  8 |###
2-4:  1 |
2-5:  0 |
2-6:  0 |
2-7:  2 |
2-8:  4 |#
2-9:  3 |#
2-10: 4 |#

3-1:  0 |
3-2:  5 |##
3-3: 46 |################## 85.19 %
3-4:  1 |
3-5:  0 |
3-6:  1 |
3-7:  0 |
3-8:  0 |
3-9:  1 |
3-10: 0 |

4-1:  0 |
4-2:  1 |
4-3:  3 |#
4-4: 38 |############### 73.08 %
4-5:  4 |#
4-6:  2 |
4-7:  0 |
4-8:  2 |
4-9:  2 |
4-10: 0 |

5-1:  0 |
5-2:  0 |
5-3:  0 |
5-4:  7 |##
5-5: 40 |################ 68.97 %
5-6:  2 |
5-7:  1 |
5-8:  5 |##
5-9:  3 |#
5-10: 0 |
```

```
6-1:  1 |
6-2:  2 |
6-3:  0 |
6-4:  6 |##
6-5:  1 |
6-6: 38 |############### 58.46 %
6-7:  0 |
6-8:  7 |##
6-9: 10 |####
6-10: 0 |

7-1:  3 |#
7-2:  1 |
7-3:  1 |
7-4:  1 |
7-5:  1 |
7-6:  2 |
7-7: 43 |################# 81.13 %
7-8:  0 |
7-9:  0 |
7-10: 1 |

8-1:  2 |
8-2:  1 |
8-3:  0 |
8-4:  2 |
8-5:  5 |##
8-6:  4 |#
8-7:  0 |
8-8: 53 |##################### 74.65 %
8-9:  3 |#
8-10: 1 |

9-1:  3 |#
9-2:  1 |
9-3:  0 |
9-4:  0 |
9-5:  0 |
9-6:  5 |##
9-7:  1 |
9-8:  4 |#
9-9: 35 |############## 71.43 %
9-10: 0 |

10-1: 3 |#
10-2: 7 |##
10-3: 2 |
10-4: 4 |#
10-5: 1 |
10-6: 1 |
10-7: 1 |
10-8: 5 |##
10-9: 7 |##
10-10:66|########################## 68.04 %
```

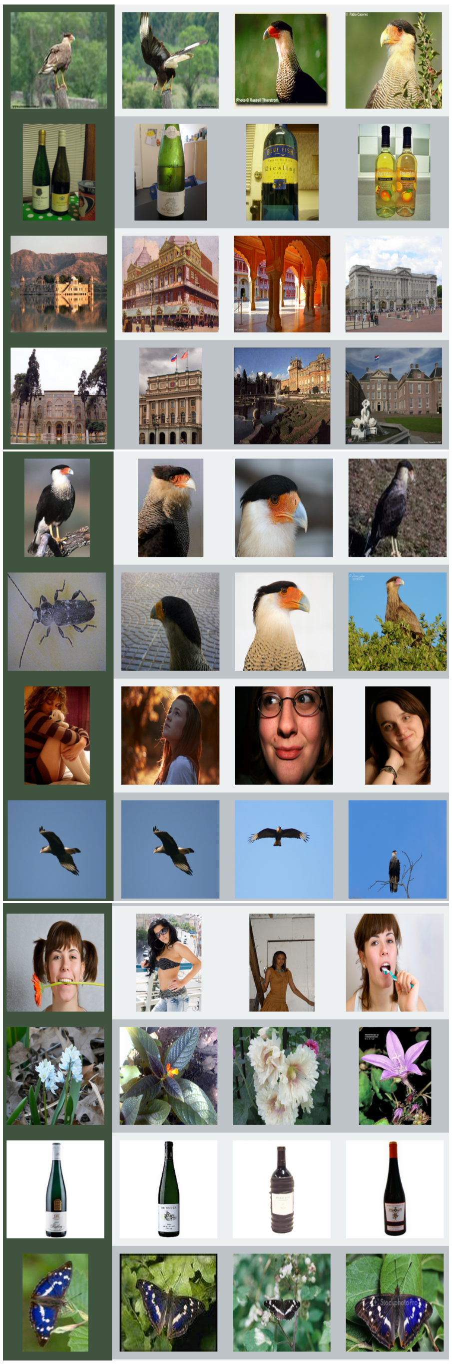
Final Accuracy of 10-classification:

```
Accuracy = 69.8206% (428/613) (classification)
```

On training set:

```
Accuracy = 88.96% (4448/5000) (classification)
```

Query image | 3 closest images in predicted class



Please see `a.html` under `result.zip` for a more detailed demo.

4. CONCLUSION AND FUTURE WORK

We have implemented a full workflow of image retrieval problem. Our program is integrated in one matlab module, and almost all parameters can be adjusted. We implemented our own K-Means algorithm, and visualized our K-Means result on patch clustering into K images. The centroids we got prove to meet clear patterns, which is similar to other successful convolutionary computer vision systems. Our training phase and feature extraction process are highly optimized, so that training 5000 images takes less than one hour, and classifying 2000 test images takes less than one minutes. Our final accuracy rate 69.82% is also competitive in all current image 10-classification algorithms using the same level of computing resources. The closest retrieval demo brings very reasonable results as well.

However there are still many wrong predictions that are trivial for human. If we use deeper machine learning model such as CNN, we may further improve our accuracy. We can also generate small noises, random rotation, flipping, and many other tricks to enrich the dataset for larger machine learning framework. So if time permitting, we will try those algorithms.

5. REFERENCES

[1] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[2] A. Coates and A. Y. Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer, 2012.

[3] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[4] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[5] D. Lu and Q. Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[6] J. A. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[7] A. Vailaya, M. A. Figueiredo, A. K. Jain, and H.-J. Zhang. Image classification for content-based indexing. *Image Processing, IEEE Transactions on*, 10(1):117–130, 2001.