LETTER

Codebook Learning for Image Recognition Based on Parallel Key SIFT Analysis

Feng YANG[†,††a)], ***Member***, **Zheng MA**[†], ***and*** **Mei XIE**[†††], ***Nonmembers***

SUMMARY The quality of codebook is very important in visual image classification. In order to boost the classification performance, a scheme of codebook generation for scene image recognition based on parallel key SIFT analysis (PKSA) is presented in this paper. The method iteratively applies classical k-means clustering algorithm and similarity analysis to evaluate key SIFT descriptors (KSDs) from the input images, and generates the codebook by a relaxed k-means algorithm according to the set of KSDs. With the purpose of evaluating the performance of the PKSA scheme, the image feature vector is calculated by sparse code with Spatial Pyramid Matching (ScSPM) after the codebook is constructed. The PKSA-based ScSPM method is tested and compared on three public scene image datasets. The experimental results show the proposed scheme of PKSA can significantly save computational time and enhance categorization rate.

key words: *codebook learning, image classification, parallel key SIFT analysis, ScSPM*

1. Introduction

In recent years, the bag-of-features (BOF) method and the spatial pyramid matching (SPM) model have been extremely popular in image classification. Over last decade, extensive research works have been done based on these two models and many algorithms have emerged, such as ScSPM model in [1] used sparse coding (SC) to obtain nonlinear codes and classified with linear classifiers, LCC mechanism in [2] constrained the sparse coding to be local, and LLC algorithm in [3] relaxed the sparse coding constraint to locality and calculated feature vector by linear coding. Generally, most of these models consist of the following four steps. Firstly, local features (*e.g.* SIFT) are extracted from each image as representations of the image regions. Secondly, a codebook is learned according to some approaches, such as k-means clustering, sparse coding, vocabulary tree and hamming embedding. Thirdly, image representation vector is formed by encoding each local feature in the codebook. Finally, image representation vector is classified by linear or non-linear classifiers.

As can be seen from these four stages, the step of codebook learning effects the quality of the final image representation and is also very important for the image classification [4]. Traditionally, the codebook was generated by unsupervised clustering manner, *e.g.* k-means, and worked well in visual classification tasks and visual regression tasks. However, this kind of method have two drawbacks: codeword uncertainty and codeword plausibility. Several algorithms have been proposed to improve the performance, such as kernel codebooks learning proposed by Gemert *et al.* [5], over-complete codebook learning proposed by JJ.Wang *et al.* [3] and J. Mairal *et al.* [6], and small-sized codebook learning by Z. Jiang *et al.* [4].

In this paper, we propose a scheme of codebook generation for image recognition based on parallel key SIFT analysis (PKSA). The main idea of this scheme is to iteratively evaluate the parallel key SIFT descriptors (KSDs) from original SIFT descriptors by using k-means clustering algorithm and similarity analysis, and then the codebook is learned by a relaxed clustering algorithm with the set of KSDs. The performance of classification is improved by PKSA algorithm and dual clustering method.

The rest of the paper is organized as follows: the framework of our proposed scheme of codebook learning based on parallel key SIFT analysis (PKSA) is introduced in detail in Sect. 2 and linear ScSPM model is reviewed in Sect. 3; Sect. 4 presents the experimental setup and results, and finally conclusions is provided in Sect. 5.

2. Scheme of PKSA Based Codebook Learning

Suppose the number of SIFT descriptors extracted from each image is N and the dimensionality is D. The set of SIFT descriptors of the m-th image can be denoted by I_m and $I_m = [x_1, x_2, \ldots, x_j, \ldots, x_N] \in R^{D\times N}$. Therefore, the training dataset with n images can be marked as I, where $I = (I_1, I_2, \ldots, I_m, \ldots, I_n)$. Let V be the codebook with M bases, $V = [v_1, v_2, \ldots, v_M] \in R^{D\times M}$.

2.1 Traditional Codebook Learning Model

Traditional codebook learning method uses classical k-means clustering algorithm to solve the following minimization formulation.

$$\arg\min_{B,V} \|X - VB\|_F^2 \qquad s.t. B_{ij} \in \{0, 1\} \tag{1}$$

Where X is the set of SIFT descriptors I, $V = [v_1, v_2, \ldots, v_M]$

Manuscript received July 27, 2016.
Manuscript revised December 4, 2016.
Manuscript publicized January 10, 2017.

[†]The authors are with the School of Communication and Information Engineering, University of Electronic Science and Technology of China, P.R.China.
[††]The author is with the School of Information and Engineering, Wenzhou Medical University, P.R.China.
[†††]The author is with School of Electronic Engineering, University of Electronic Science and Technology of China, P.R.China.
a) E-mail: yangfeng_34@163.com

DOI: 10.1587/transinf.2016EDL8167

is the learned codebook with M entries determined by cluster centers, $\| \bullet \|_F$ is Frobenius norm, $B = [b_1, b_2, \ldots, b_N]$ and b_i is M-dimensional binary vector.

The constraint of classical k-means algorithm is too strict and each local feature in the codebook is assigned to just one visual word. A relaxed binary condition named non-negative matrix factorization (NMF) [7] is introduced to solve the minimization problem in Eq. (2). The NMF model is a special constrained sparse coding algorithm with the aim of alleviating the information loss of the sparse coding plus max pooling.

$$\arg\min_{B,V} \|X - VB\|_F^2 \qquad s.t.\ \|b_i\|_{l^1} = 1, b_i \geq 0, \forall i \tag{2}$$

2.2 Codebook Learning Based on PKSA

With the aim of improving the quality of the codebook and reducing the computational time, the strategy of parallel key SIFT analysis (PKSA) is introduced in this paper, which makes use of the combination of k-means clustering and NMF algorithm.

The basic idea of PKSA is selecting the key SIFTs (a subset of SIFT descritors) from each image with more representative characteristics but much fewer numbers by iteratively using k-means algorithm and similarity analysis, and learning the codebook with the selected collection of key SIFTs from training dataset by clustering algorithm of NMF. Therefore, representative and useful features in the image are selected so that the time for codebook learning is greatly reduced and the learned codebook V is likely to represent all of the images well.

According to the pseudocode in Alg.1, the strategy of PKSA can be divided into two steps: parallel key SIFT selection (PKSS) and codewords clustering. The former aims at selecting the key SIFT descriptors (KSDs) from each image by iteratively deleting the redundant SIFT descriptors (with smaller S values comparing with the similarity adjust factor σ) so as to increase the representativeness of the selected key SIFTs, while the later learns the codebook by NMF algorithm.

The process of PKSS is quite simple, including an iterative parallel identification of K KSDs and an elimination of non-key ones. The KSDs are identified according to the center of k-means clustering and the non-key ones are selected by similarity analysis.

$$S_{ij} = \|x_i - x_j\|_{l^1} \tag{3}$$

$$\sigma = w * \frac{1}{N} \sum_{j=1}^{N-K} S_{ij} \tag{4}$$

Based on all of the KSDs selected from the training dataset I in Z, $Z = (I_{12}, I_{22}, \ldots, I_{m2}, \ldots I_{n2})$, the codebook V is efficiently generated by using codewords clustering algorithm of NMF with a relaxed binary condition as defined in Eq. (2), where $V = [v_1, v_2, \ldots, v_M]$ is codebook to be evaluated.

Algorithm 1 The pseudocode for Parallel Key SIFT Analysis

Input: The SIFT descriptors of training dataset $I = (I_1, I_2, \ldots, I_m, \ldots, I_n)$
Output: The learned codebook V

```
//1. parallel key SIFT selection
for m from 1 to n, I_m is the SIFT descriptors of the m-th image, I_m = [x_1, x_2, ..., x_N], do
    //1.1 initialization
    Candidate key SIFT descriptors (CKSD) set Y1 ← I_m
    Size of CKSD set N' ← N
    //1.2 iteration
    while N' ≥ K do
        //1.2.1 k-means clustering
        Separate N' CKSDs into K clusters by k-means algorithm
        for i from 1 to K do
            //1.2.2 define KSDs
            Define the center of the i-th cluster as x'_i
            for j from 1 to N' do
                Calculate the distance between x'_i and x_j
            end for
            Define the CKSD x_i with minimun distance from x'_i as KSD
            //1.2.3 calculate similarity S
            for j from 1 to N' do
                Calculate similarity S_ij between KSD x_i and non-KSD x_j
according to Eq. (3)
            end for
            //1.2.4 calculate σ
            Evaluate the similarity adjust factor σ according to Eq. (4)
            //1.2.5 update the CKSD set by comparing each similarity S_ij
with the similarity adjust factor σ
            for j from 1 to N' do
                if (S_ij < σ) and (j ≠ i) then
                    Delete x_j from Y1
                end if
            end for
            //1.2.6 update
            Save KSD x_i in set Y2
            Update N'
        end for
    end while
    I_m2 ← Y2
end for
//2. codewords clustering
Generate the codebook V by NMF algorithm according to Eq. (2) based
on the KSD set Z = (I_12, I_22, ..., I_m2, ... I_n2)
Return V
```

The proposed method can increase the representativeness of the codewords and reduce the dimensionality of the codebook significantly so as to increase the classification rate. Suppose the number of SIFT descriptors in original training dataset is N_{before} and the number after PKSA is N_{after}. Since N_{after} is much less than N_{before}, about 5%-10% of N_{before}, the size of codebook V can be greatly reduced. Therefore, the time of codebook learning can be significantly reduced.

3. Linear ScSPM Model for Classification

In order to shown the effectiveness of the scheme, the final feature vectors are calculated and the images are classified by using linear ScSPM model in [1] with the aim to solve the following optimization problem.

$$\arg\min_{C} \sum_{i=1}^{N} \|x_i - Vc_i\|^2 + \lambda \|c_i\|_{l^1} \quad (5)$$

where $C = [c_1, c_2, \ldots, c_N]$ is the cluster membership indicator, $\lambda\|c_i\|_{l^1}$ is the sparsity regularization term with the purpose of achieving a unique solution and much less quantization error.

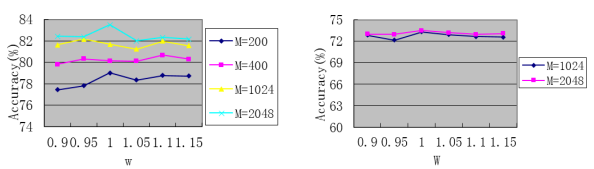
4. Experimental Results

In the experimental section, we report the performance of our PKSA algorithm on three public datasets: fifteen scenes dataset, Caltech-101 dataset and Caltech-256 dataset. Our experiments use only a single SIFT descriptor of each image, with dimension of 128, extracted from patches of 16*16 pixels and densely sampled by a step of 8 pixels. The parallel parameter K in our experiments is preseted to be 2. With the aim of achieving reliable results, all experimental results are repeated 10 times by randomly selecting training and testing data according to the common benchmarking procedures. The average recognition rate of every class is calculated for each run and the mean and standard deviation of the recognition rates are recorded as the final results. All the experiments were implemented using Matlab on a PC with 3.30GHz Intel(R) Xeon(R) CPU and 32G memory.

4.1 Fifteen Scenes Dataset

The first dataset is fifteen scenes dataset with 4485 images in 15 categories. The average size of each image is 300*250 pixels and the image number in each category ranges from 200 to 400. The number of training images is 100 per class and the detailed experimental results of this database is illustrated in Fig. 1 (a) with different sizes of codebook (M): 200, 400, 1024 and 2048. As shown in Fig. 1 (a), the classification rates are increased with larger value of M and almost stable with different values of weight parameter w.

The comparison results of different methods in classification accuracy are shown in Table 1. All methods are under the same set of training on 100 images per class and testing on the rest. Our scheme outperforms LSS by more than 11% and ScSPM by more than 3%. It also can be concluded from Table 1 that not all of deep neural networks work very well on this kind of small scale datasets for the limitation of the number of training images. For example, although the method DDSFL+Caffe achieves impressive improvement, LDANet and DLANet only obtain a slight improvements and PCANet even a bitter lower than our proposed method.



(a) Fifteen scenes (b) Caltech-101

Fig. 1 Performance of the proposed algorithm on Fifteen scenes and Caltech-101 with different M

4.2 Caltech-101 Dataset

Our second dataset is Caltech-101, which contains 9144 images of 101 categories with significant variance in shape, such as brain, airplane, bass, anchor and so on. The image resolution is 300*300 pixels and the number of images in each class is quite different, varies from 31 to 800. According to the common experimental setup for Caltech-101, we trained on 30 images per class and tested on the rest. And the final performance is measured by calculating average accuracy of 101 classes and one background class. The performance of different sizes of codebook (M), 1024 and 2048, are compared in Fig. 1 (b).

The comparison results of different methods in classification accuracy are shown in Table 2 with the codebook trained on 1024 bases. We tested the PKSA algorithms on 5, 10, 15, 20, 25 and 30 training images per class, respectively. Our scheme outperforms ScSPM more than 0.7% and even better than LLC on all test results. The deep neural networks algorithm of DeCAF outperforms our method with 30 training images per class, while the classification of PCANet is less than our algorithm both in 15 and 30 training images per class.

4.3 Caltech-256 Dataset

The last dataset of experiments is Caltech-256 consisting of 30,607 images in 256 categories with much higher variability in object size, pose and location. There are more than 80 images in each category with image size less than 300*300 pixels. The proposed algorithm is performed with the codebook of 1024 bases on 15, 30, 45 and 60 training images per class, respectively. The experimental results of the proposed

Table 1 Comparisons with different methods on fifteen scenes dataset.

Algorithms	Accuracy (%)
ScSPM [1]	80.28±0.93
KSPM [1]	76.73±0.65
OB [8]	80.9
LSS [9]	72.20±0.20
PCANet [10]	82.73±0.40
LDANet [10]	84.75±0.69
DLANet [11]	85.13±0.38
DDSFL [12]	84.42
DDSFL+Caffe [12]	92.81
Ours	83.27±0.39

Table 2 Experimental results on Caltech-101 dataset

Training images	5	10	15	20	25	30
ScSPM [1]	-	-	67.0	-	-	73.2
LLC [3]	51.15	59.77	65.43	67.74	70.16	73.44
D-KSVD [13]	49.6	59.5	65.1	68.6	71.1	73.0
K-SVD [14]	49.8	59.8	65.2	68.7	71.0	73.2
DeCAF [15]	-	-	-	-	-	86.91
PCANet [10]	-	-	61.46	-	-	68.56
Ours	52.5	61.4	67.8	69.1	71.6	73.8

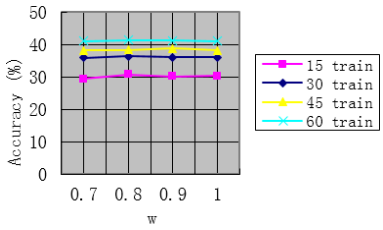
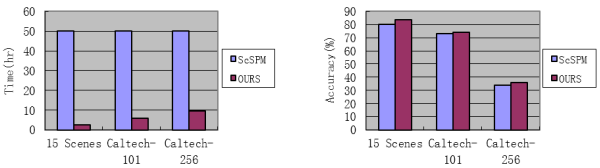


Fig. 2 Classification accuracy with different training images on Caltech-256.

Table 3 Experimental results on Caltech-256 dataset

Training images	15	30	45	60
LScSPM [16]	30.00±0.14	35.74±0.10	38.54±0.36	-
SIFT LLC [17]	25.06±0.07	31.22±0.24	34.92±0.39	37.22±0.35
ScSPM [1]	27.73±0.51	34.02±0.35	37.46±0.55	40.14±0.36
Ours	30.66±0.22	35.68±0.32	38.73±0.28	41.59±0.06



(a) Comparison on time for codebook learning over three datasets.

(b) Comparison on classification accuracy over three datasets.

Fig. 3 Experimental comparisons between ScSPM model and ours

scheme with different values of training images are shown in Fig. 2 and the comparison results with other models are listed in Table 3.

4.4 Performance Analysis over Three Datasets

The performances of computational time for codebook learning and classification accuracy over three datasets are compared with ScSPM in Fig. 3 (a) and Fig. 3 (b), respectively. Different from more than 50 hours of codebook learning method in ScSPM model, the proposed PKSA scheme takes much less time even in the database of Caltech-256 and achieves approximately 1.5%∼3% enhancement in classification.

5. Conclusion

In this paper, we propose a scheme of codebook learning for image recognition based on parallel key SIFT analysis (PKSA). The method iteratively uses k-means clustering algorithm and similarity analysis to evaluate key SIFT descriptors and filter out others, and generates the codebook by a relaxed clustering algorithm of NMF according to the selected set of KSDs. We perform experiments on 3 widely used image databases based on ScSPM model. And experimental results show that our algorithm reduces the computational time for codebook learning significantly while obtains higher categorization accuracy.

References

[1] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1794–1801, IEEE, 2009.
[2] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," Proc. 2009 Conference on Advances in Neural Information Processing Systems, pp.2223–2231, 2009.
[3] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3360–3367, IEEE, 2010.
[4] Z. Jiang, Z. Lin, and L.S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1697–1704, IEEE, 2011.
[5] J.C. Van Gemert, J.-M. Geusebroek, C.J. Veenman, and A.W.M. Smeulders, "Kernel codebooks for scene categorization," 10th European Conference on Computer Vision (ECCV 2008), vol.5304, pp.696–709, Springer Verlag, 2008.
[6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," J. Machine Learning Research, vol.11, pp.19–60, 2010.
[7] C. Lang, S. Feng, B. Cheng, B. Ni, and S. Yan, "A unified supervised codebook learning framework for classification," Neurocomputing, vol.77, no.1, pp.281–288, 2012.
[8] L.J. Li, H. Su, E.P. Xing, and L. Fei-Fei, "Object bank: A high-level image representation for scene classification and semantic feature sparsification," 2010 Conference on Neural Information Processing Systems (NIPS 2010), pp.1378–1386, 2010.
[9] N. Rasiwasia and N. Vasconcelos, "Scene classification with low-dimensional semantic spaces and weak supervision," 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1–6, IEEE, 2008.
[10] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?," IEEE Trans. Image Process., vol.24, no.12, pp.5017–5032, 2015.
[11] Z. Feng, L. Jin, D. Tao, and S. Huang, "Dlanet: A manifold-learning-based discriminative feature learning network for scene classification," Neurocomputing, vol.157, pp.11–21, 2015.
[12] Z. Zuo, G. Wang, B. Shuai, L. Zhao, and Q. Yang, "Exemplar based deep discriminative and shareable feature learning for scene image classification," Pattern Recognition, vol.48, no.10, pp.3004–3015, 2015.
[13] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.2691–2698, IEEE, 2010.
[14] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," IEEE Trans. on Signal Processing, vol.54, no.11, pp.4311–4322, 2006.
[15] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," Computer Science, vol.50, pp.815–830, 2013.
[16] L. Yang, R. Jin, R. Sukthankar, and F. Jurie, "Unifying discriminative visual codebook generation with classifier training for object category recognition," 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1253–1260, IEEE, 2008.
[17] J. Chen, Q. Li, and Q. Peng, "Csift based locality-constrained linear coding for image classification," Pattern Analysis and Applications, vol.18, no.2, pp.441–450, 2015.