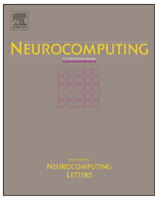
Neurocomputing 144 (2014) 509–515

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Hierarchical feature coding for image classification

Jingyu Liu *, Yongzhen Huang, Liang Wang, Shu Wu

Institute of Automation, Chinese Academy of Sciences (CASIA), National Laboratory of Pattern Recognition (NLPR), Beijing 100190, China

ARTICLE INFO

Article history:
Received 28 January 2014
Received in revised form
4 April 2014
Accepted 22 April 2014
Communicated by Qingshan Liu
Available online 24 May 2014

Keywords:
Image classification
Hierarchical encoding
Higher level representation

ABSTRACT

Feature coding and pooling are two critical stages in the widely used Bag-of-Features (BOF) framework in image classification. After coding, each local feature formulates its representation by the visual codewords. However, the two-dimensional feature-code layout is transformed to a one-dimensional codeword representation after pooling. The property for each local feature is ignored and the whole representation is tightly coupled. To resolve this problem, we propose a hierarchical feature coding approach which regards each feature-code representation as a high level feature. Codeword learning, coding and pooling are also applied to these new features, and thus a high level representation of the image is obtained. Experiments on different datasets validate our analysis and demonstrate that the new representation is more discriminative than that in the previous BOF framework. Moreover, we show that various kinds of traditional feature coding algorithms can be easily embedded into our framework to achieve better performance.

1. Introduction

Image classification is a fundamental vision problem which is to classify images to the specified one or more categories. It has a wide range of applications in image retrieval [1–3], web analysis [4–6], etc. This is a very challenging task due to the variability of illumination, scales, rotation, viewpoints and occlusion. Inspired by the bag of words (BOW) model [7] in document analysis, the bag of features (BOF) model [8] has been demonstrated successful for image classification. In the BOF model, an image is modeled as an unordered composition of visual features which are encoded by a group of visual codewords. After that, features' responses on each codeword are pooled to one single value, and the image is finally described as a codebook histogram.

Coding and pooling are two critical procedures of the traditional BOF model. Many efforts have been dedicated to develop effective encoding and pooling algorithms. Though many algorithms have been proposed, the inherent characteristics of coding and pooling stay unchanged. Our proposed hierarchical framework is inspired by the essential drawbacks of coding and pooling, as can be summarized in the following two aspects:

1. The nature of coding is to partition the continuous feature space to discrete visual words. Different coding strategies are employed to assign each feature to its surrounding visual words. Inspired by Huang et al. [9], we interpret coding as a process of constructing connections. Features and visual words can be deemed as vertexes in the feature space. After coding, an undirected and weighted edge will bridge each local feature and their surrounding visual words. A more weighted edge characterizes an accurate approximation of features, whereas a less weighted edge indicates the ambiguity of visual words. Therefore, we believe such connections yield some valuable information, which yet, are not fully utilized in the traditional framework.
2. After coding, the traditional BOF framework will enter the next stage, pooling. The nature of pooling is to accumulate local features to a global appearance-based representation. For each local feature, the weighted connections with its surrounding visual words are obliterated in the process of pooling. Therefore the abundant and more subtle information of each local feature are abandoned in the process of pooling. Figs. 1 and 2 illustrate the phenomenon. Fig. 1 shows average pooling, where different appearances result in the same visual word histogram after pooling. As a result, two images from different categories might be wrongly classified into the same one. Fig. 2 shows max pooling, where only the largest response (0.5) is preserved. Though close enough, other values (0.49) are ignored.

Current studies on feature coding combined with feature pooling naturally result in the drawback of the traditional BOF framework. As analyzed above, the pooling operation ignores the connections of each local feature and their surrounding visual words. To address this, we deem the connections between features and visual words as a kind of "higher level" features (here,

* Corresponding author. Tel.: +86 1381 022 6465.
E-mail address: jyl_999@163.com (J. Liu).

http://dx.doi.org/10.1016/j.neucom.2014.04.022
0925-2312/© 2014 Elsevier B.V. All rights reserved.

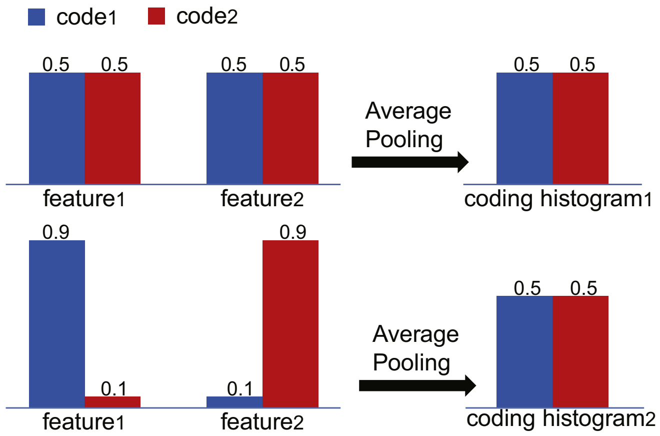


Fig. 1. Different feature appearances formulate the same visual word histogram after average pooling. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

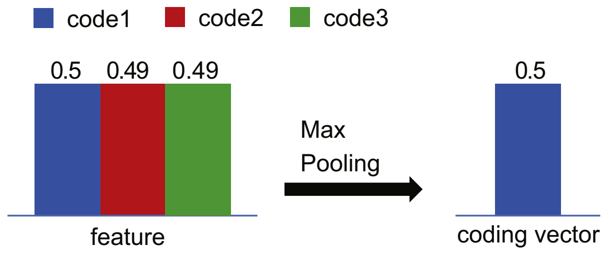


Fig. 2. Max pooling ignores other significant responses. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

"higher" is against the pixel level representation, e.g., SIFT [10] and HOG [11]). Based on this consideration, we propose a hierarchical BOF framework. In addition to the traditional pipeline, higher level features also generate the codebook and go through the stage of coding and pooling. In the end, a global histogram describing the frequency of connections between features and visual words are obtained.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 provides the details of various coding methods based on the hierarchical framework. Section 4 evaluates our framework on two different datasets and discusses why the two-layer framework improves the performance. Section 5 concludes the paper with discussion on future research.

2. Related work

In this section, we introduce related work of the BOF framework. A traditional BOF framework generally consists of the following stages:

(1) *Extract local features*: This step involves sampling local patches and describing them via classic feature descriptors. Local patches can be sampled in either a dense (with a fixed grid) or a sparse (with feature detectors) way. One of the typical feature descriptors is the scale-invariant feature transform (SIFT) descriptor [10]. It describes a local area by accumulating pixel gradients from each orientation weighted by their magnitude. In image classification, the general operation usually divides orientations into 8 bins in 16 sub-regions. Other typically used descriptors include local binary pattern (LBP) [12] and histogram of gradients (HOG) [11]. The inputs of this step are images, and the outputs are feature vectors.

(2) *Generate a codebook*: This step generates a codebook via learning from local features. For the computational efficiency, usually a subset of descriptors are randomly selected from all feature vectors obtained from the first step. The learning procedure is often implemented by unsupervised learning, e.g., K-means [13], or supervised learning [14]. Clustered centers are approximations of features and are often called codewords. In general, performance would be enhanced as the number of codewords becomes larger, since feature appearance spans over a large space and more codewords can present more sophisticated appearance of features. The inputs of this step are feature vectors and the output is the codebook consisting of codewords.

(3) *Encode features*: This step encodes local features to the codewords. Each feature will activate its nearest codewords measured in the feature space, and one or more codewords might obtain responses. Many encoding methods have emerged since it is not trivial to determine which codeword to activate as well as the weight with it. The input of this step is the codebook and the output is the coding vector. There are mainly five kinds of coding methods [15].

- Voting-based methods [8,16] apply a histogram to approximate the probability distribution of features. Each feature votes to its nearest one or multiple codewords, and the weight with the vote is obtained by hard quantization or soft quantization.
- Reconstruction-based methods [17–19] employ a subset of codewords to reconstruct a feature. Penalty is added to assure that few codewords are employed. So the optimization problem is formulated with certain constraints on the codewords, and the target is to minimize the reconstruction error. Sparse coding is widely used in reconstruction-based methods, wherein constraint terms are the main differences among various methods [20–26].
- Saliency-based coding [27] introduces the concept of codeword saliency, which is measured by relative proximity of the closest codeword compared with other codewords. Combining with MAX pooling, only the strongest response is preserved, indicating that the codeword can independently describe the feature without others.
- Local tangent-based coding [28] models features and codewords based on the manifold theory. It is assumed that codewords are located on the same smooth manifold constituted by all features. The encoding is formulated by using codewords to approximate the manifold. Lipschitz smooth function is applied to express the feature manifold.
- Fisher coding [29] is based on the Fisher kernel, which uses the gradient vector of its probability density function to describe a signal. IFK [30] employs Gaussian Mixture Model to estimate feature distributions. Each of the multiple Gaussian distributions reflects one pattern of features. Mean vector and covariance matrix are used to encode features.

(4) *Pool features*: This step is implemented via pooling votes obtained by each code. Typical pooling methods involve average pooling by averaging all the votes and MAX pooling by picking the most significant vote. One major drawback of pooling is that it ignores the spatial distribution in the process of the descriptor quantization. The problem can be partially resolved via spatial pyramid matching (SPM) [31] and multiple spatial pooling (MSP) [32]. SPM partitions an image into increasingly finer subregions and then employs pooling independently in them, which accords with the regular spatial structure of images from a particular category. An in-depth research on pooling can be found in [33].

3. A hierarchical coding framework

In this section, the pipeline of our hierarchical coding framework is firstly illustrated in Section 3.1. Then the details of various embedded coding methods are respectively described in Sections 3.2–3.4.

3.1. Pipeline of the hierarchical coding framework

The output of the coding stage is called the coding vector, which records responses of one feature on all the codewords, as shown in Fig. 3. In our proposed hierarchical framework, coding vector of each local feature is deemed as a “higher level” feature. All coding vectors are trained to generate the codebook, encoded and pooled afterwards.

The detailed process is as follows: let $X=[x_1,x_2,\dots,x_N]\in\mathbf{R}^{D\times N}$ denote N D-dimensioned features extracted from a single image, $B_1=[b_1,b_2,\dots,b_M]\in\mathbf{R}^{D\times M}$ denote M codewords obtained via clustering over features, and $V=[v_1,v_2,\dots,v_N]$ denote N coding vectors obtained via encoding. After pooling, a final representation F of a single image is obtained, $F=[f_1,f_2,\dots,f_M]$ is a vector of length M, representing a distribution of visual codewords. Coding vectors V obtained from the first layer is regarded as the second layer features. Let $B_2=[b'_1,b'_2,\dots,b'_{M'}]\in\mathbf{R}^{D'\times M'}$ denote M' codewords obtained via clustering over V, where $D'=M$ is the dimension of the second layer codewords. After pooling, another representation F' of a single image is obtained, $F'=[f'_1,f'_2,\dots,f'_{M'}]$ is a vector of length M'.

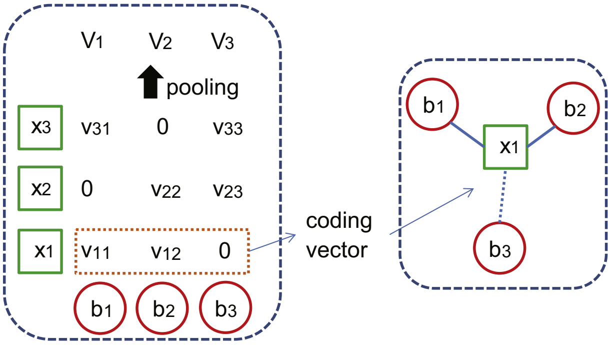


Fig. 3. Coding vectors are coupled after pooling (on the left), while they reflect connections between features and codewords (on the right). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Fig. 4 illustrates the difference between our approach and the traditional BOF framework.

3.2. Hierarchical voting-based coding

Voting-based coding methods approximate the probability distribution of codewords by a histogram of votes. Hard voting [8] only assigns each feature to their nearest codeword, then denotes codewords' existence by simple 0/1 response, hence too coarse to get higher accuracy. Instead, soft voting [16] (SV) applies a kernel function to measure the similarity between features and their nearest several codewords. In this paper, we combine our framework with soft voting, and the coding strategy is as follows:

$$v(i)=\frac{\exp(\|x-b_i\|_2^2/\sigma)}{\sum_{k=1}^{K}\exp(\|x-b_k\|_2^2/\sigma)},\quad i=1,2,\dots,M, \tag{1}$$

where x and b are feature and codeword respectively, $\sum_{k=1}^{K}\exp(\|x-b_k\|_2^2/\sigma)$ is the normalization factor, σ is a smooth parameter and $v=[v(1),\dots,v(M)]$ is the coding vector obtained by the first layer. Recent work in [34] demonstrates that higher accuracy is obtained when K is set to a small number rather than M. In the second layer, the soft coding strategy is reproduced to formulate a higher level representation:

$$v'(i)=\frac{\exp(\|v-b'_i\|_2^2/\sigma)}{\sum_{k=1}^{K'}\exp(\|v-b'_k\|_2^2/\sigma)},\quad i=1,2,\dots,M'. \tag{2}$$

Fig. 5 illustrates the pipeline of the proposed hierarchical framework by voting-based coding. To better illustrate this, assuming in the first layer a SIFT feature is extracted to describe a patch, maintaining a pixel-level representation. After extracting features and training them, the codebook of the first layer is obtained. Next, each local feature (red square) constructs connections with its surrounding codewords (blue circle) in the procedure of coding. After that, the pipeline of the second layer starts, and connections (dashed rectangle) are trained to generate higher level codewords (red diamond). The second layer framework will

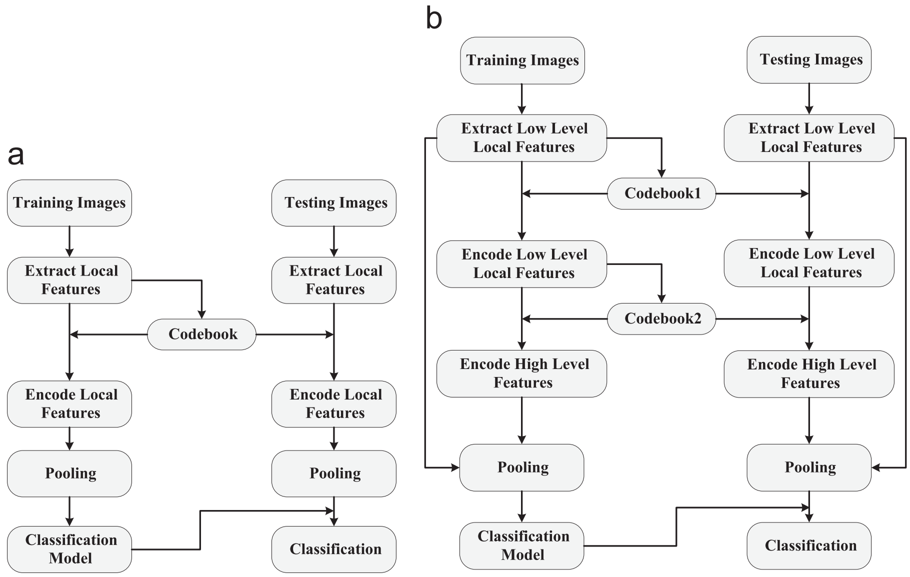


Fig. 4. Comparison between (a) the traditional BOF framework and (b) the proposed hierarchical BOF framework.

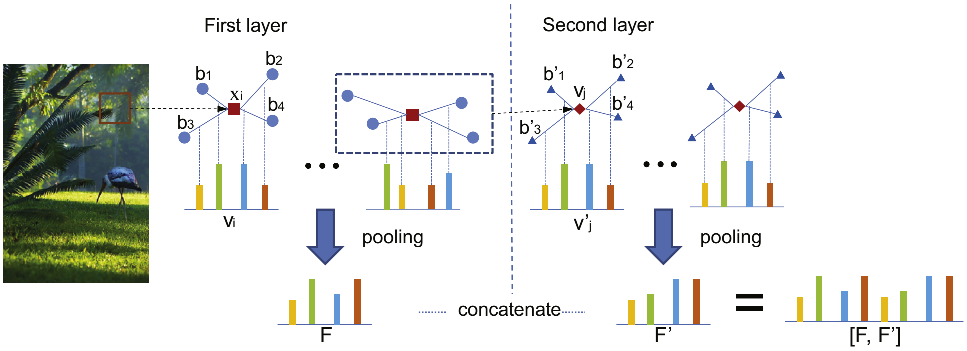


Fig. 5. Pipeline of the hierarchical framework by voting-based coding. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

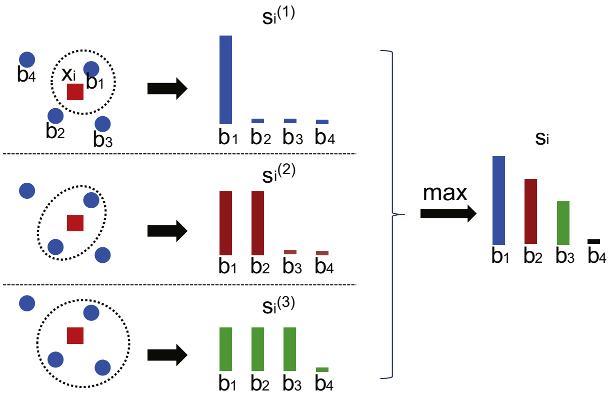


Fig. 6. Group saliency coding. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

go through the pipeline the same as the original framework. After pooling, representations of both layers are concatenated as the input of the classifier.

3.3. Hierarchical saliency-based coding

Saliency-based coding [27] (SAC) approximates the salient degree one codeword might have, relative proximity is used to measure the salient degree. Despite the simplicity of SAC, results demonstrate that it can compete with sparse coding [17] and consumes less time. In traditional saliency coding, only the nearest codeword receives the response of one feature. Recent work called group saliency coding [35] (GSC) demonstrates that higher accuracy can be obtained if a group of codewords receive responses together. Fig. 6 illustrates the coding strategy of group saliency coding.

In this paper, we combine GSC with our framework, and the coding strategy is as follows:

$$v(i)=\max\{s_i^k\},\quad k=1,\dots,K$$

$$s_i^k=\begin{cases}\psi^k(x) & \text{if}\quad b^i\in g(x,k)\\ 0 & \text{otherwise}\end{cases}$$

$$\psi^k(x)=\sum_{j=1}^{K+1-k}\|x-\overline{b}_{k+j}\|_2-\|x-\overline{b}_k\|_2 \qquad (3)$$

where s_i^k is the ith entry of the coding result obtained with the group size k, $\psi^k(x)$ is the function measuring the group saliency degree, $g(k,x)$ is the set of the k closest codewords of x, $\overline{b}_i$ is the ith nearest neighbouring codeword, K is the maximum group size, and $v=[v(1),\dots,v(M)]$ is the coding vector obtained by the first layer. In the second layer, group saliency coding strategy is reproduced to formulate a higher level representation:

$$v'(i)=\max\{s_i'^k\},\quad k=1,\dots,K$$

$$s_i'^k=\begin{cases}\psi'^k(x) & \text{if}\quad b'^i\in g(x,k)\\ 0 & \text{otherwise}\end{cases}$$

$$\psi'^k(x)=\sum_{j=1}^{K+1-k}\|v-\overline{b'}_{k+j}\|_2-\|v-\overline{b'}_k\|_2. \qquad (4)$$

3.4. Other hierarchical coding methods

It is evident that the hierarchical framework can also embed other coding methods such as reconstruction-based methods (LCC [18] and LLC [19]), fisher-kernel coding [29], and super vector coding [36]. The implementation detail is similar with the hierarchical framework mentioned above.

4. Experimental results and discussion

Our approach is evaluated on two databases: VOC07 [37] and 15 natural scenes [38]. To explore the compatibility of our hierarchical framework with different coding strategies, we chose three representative coding algorithms. The choice of the pooling operation is based on previous evaluation rules [15]. They are:

1. Soft voting with the average pooling operation.
2. Group saliency coding with the max pooling operation.
3. Fisher coding with the average pooling operation.

In our hierarchical framework, coding strategies remain the same as they are in the first layer:

1. Soft voting with the average pooling operation plus the second layer representation.
2. Group saliency coding with the max pooling operation plus the second layer representation.
3. Fisher coding with the average pooling operation plus the second layer representation.

Our experimental settings are the following: gray SIFT descriptors [10] are used to extract local features by dense sampling. Three scales, 16 × 16, 24 × 24, 32 × 32, are adopted to extract different sizes of features. For FK coding, visual codes are generated by GMM (Gaussian Mixture Model); for other methods, visual codes are generated by the K-means clustering algorithm. The SPM of [1 × 1, 2 × 2, 1 × 3] are adopted for both datasets, and Lib-linear SVM [39] is employed for classification. All three coding strategies are re-implemented in the same framework to achieve effective comparison. Our results might be slightly different from those of the original authors due to the implementation details.

4.1. PASCAL VOC07 dataset

The PASCAL VOC07 dataset [37] is one of the most challenging datasets for image classification. It contains 9963 images originated from 20 classes including person, bicycle, bird, etc. The dataset is challenging due to large variations of size, scale, viewpoint, clutter and deformation. Training and testing images have been carefully divided and the labels of testing images have been released.

We firstly study the performance improved by the second layer codebook. To focus on the second layer, we fix the first layer codebook size to 32 and test the performance of both SV and GSC. The result is shown in Fig. 7. For both SV and GSC, the overall tendency is that more codewords generate better performance. Because of the over-fitting effect, the accuracy of both coding methods will decrease when the dimension of the representation gets very large. The performance curve shows that SV is more sensitive to the over-fitting effect than GSC. The performance of SV stops increasing when the codebook size is 256, whereas the one of GSC still increases until the codebook size reaches 4096.

Our next experiment is designed to reflect the overall tendency of coding dimension. To make a universal comparison among different codebook sizes, we uniformly set the second layer codebook size 8 times of the first layer codebook size. The result is shown in Fig. 8. For SV, the accuracy improves 0.82%, 1.01% and 0.71% in terms of first layer size 32, 128 and 512 respectively. However, when the first layer codebook size reaches 2048, both the baseline and hierarchical framework obtain the mean average accuracy of 51.53%. For GSC, the hierarchical framework obtains more improvement, i.e. 6.51%, 2.71% and 1.28% respectively in terms of size 32, 128 and 512.

For both SV and GSC, the result shows that more improvement is obtained when the first layer codebook size is small. Because a small codebook size fails in providing accurate descriptions of features, the hierarchical framework can complement more than that of the large size codebook.

The dimension of Fisher coding based representation is proportional to the production of the codebook size and feature dimension. We only test the case when the codebook size of both layers are 16. Following the general operation of FK coding, we apply Principle Component Analysis (PCA) [40] to the raw SIFT patch of 128 dimensions, and obtain an 80-dimensioned vector. After encoding, the coding vector would be 2560 ($2 \times 16 \times 80$) dimensions. Since 2560 is too large for FK encoding, we apply PCA again for dimension reduction. We test different levels of energy preserved after PCA, i.e. different kinds of reduced dimensions. Results demonstrate that the reduced dimension should be neither

Table 1
Performance of hierarchical FK coding with different feature dimensions in the second layer on the VOC07 dataset (codebook sizes of both layers are 16).

Second layer feature dimension	80	320	640	960	1280	Baseline
MAP (%)	56.49	57.65	57.96	57.88	57.74	56.85

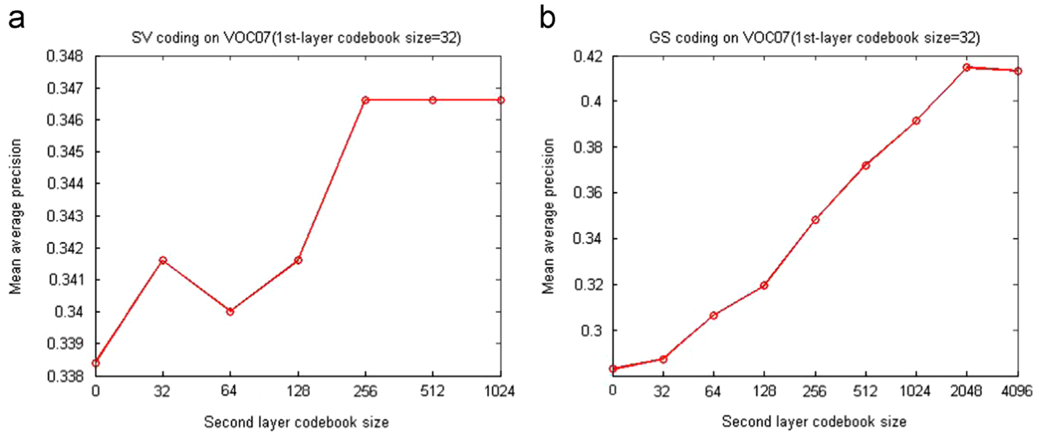


Fig. 7. Influence of second layer codebook size when the first layer codebook size is set to 32. (a) SV on the VOC-07 dataset. (b) GSC on the VOC-07 dataset.

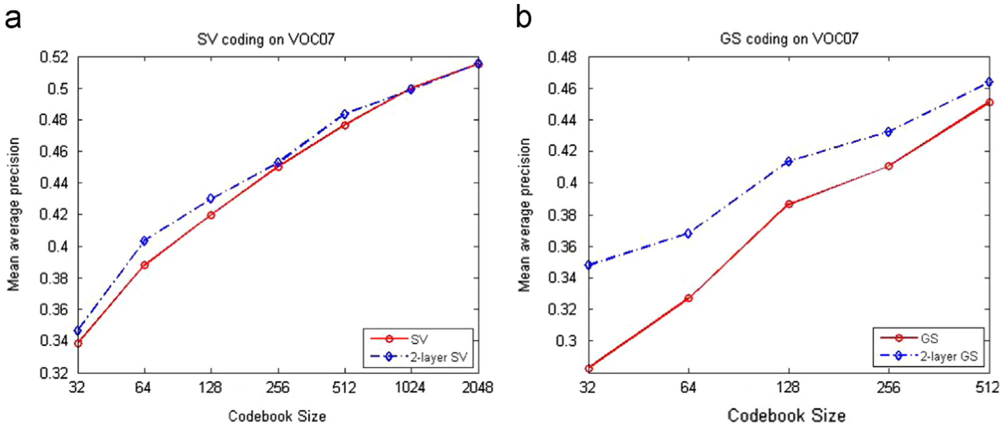


Fig. 8. Performance comparison of SV and GSC on the PASCAL VOC07 dataset.

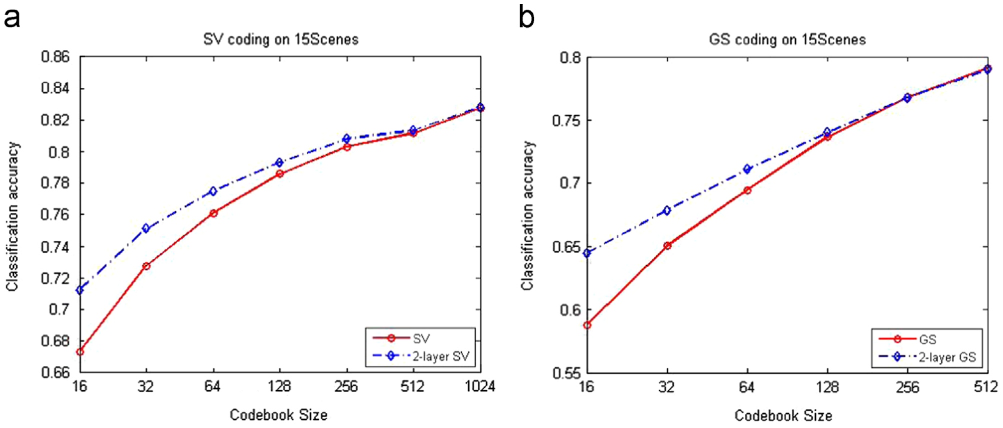


Fig. 9. Performance comparison of SV and GSC on the 15-Scenes dataset.

too high nor too low. Mean average precision has been improved to 57.65% when the reduced dimension is 320. Performance of different second layer feature dimensions is listed in Table 1.

4.2. 15-Scenes dataset

The 15-Scenes dataset contains 4485 images in 15 categories of natural and human scenes. Each category consists of 200–400 images. We follow the traditional experimental setup used in [31], wherein 100 images are randomly selected from each category for training and the rest for testing.

We test SV and GSC on the 15-Scenes dataset. The overall tendency is similar to that displayed on the VOC07 dataset. The smaller the codebook size, the greater enhancement can be obtained by our framework. Moreover, GSC gets more improvement than SV when the codebook size is small. The result is shown in Fig. 9.

4.3. Discussion

While the traditional combination of coding and pooling ignores the connections between features and codewords, our model is to preserve and utilize the information contained by them. A strong connection to one codeword means it could accurately describe a feature, whereas a weak connection indicates the ambiguity of the codeword. In our hierarchical model, connections recorded by the coding vectors are deemed as “higher level” features. So it is reasonable and nature to apply the BOW framework to the “higher level” features. The experimental results validate our analysis especially when there are fewer codewords in the first layer. Because few codewords provide only vague representations of visual features, the distance (connection) between a feature and its surrounding codewords falls in a wide range. So the coding histogram in the second layer presents a more accurate representation measured in feature-codeword distance (connection). That is why our hierarchical model improves the performance.

5. Conclusion

In this paper, we have discussed the drawback caused by the traditional combination of coding and pooling in the BOF framework. Motivated by that, we have proposed a hierarchical framework wherein coding vectors obtained in the first layer are treated as higher level features. The hierarchical framework is flexible wherein various coding and pooling methods can be easily embedded. Experimental results have demonstrated that our approach can effectively improve the accuracy for image classification. In future, further efforts might be focused on two aspects: (1) to overcome the drawbacks of coding and pooling, use other methods to explore the information of feature-codeword connections and (2) add more layers to formulate higher level representation of an image.

References

[1] A. Vailaya, M.A. Figueiredo, A.K. Jain, H.-J. Zhang, Image classification for content-based indexing, IEEE Trans. Image Process. 10 (1) (2001) 117–130.

[2] X. Tian, Y. Lu, Discriminative codebook learning for web image search, Signal Process. 93 (8) (2013) 2284–2292.

[3] X. Tian, D. Tao, X. Hua, X. Wu, Active reranking for web image searchimage processing, Signal Process. 19 (3) (2010) 805–820.

[4] R. Kosala, H. Blockeel, Web mining research: a survey, ACM SIGKDD Explor. Newsl. 2 (1) (2000) 1–15.

[5] Y. Huang, K. Huang, D. Tao, T. Tan, Enhanced biologically inspired model for object recognition, IEEE Trans. Syst., Man, Cybern., Part B: Cybern. 41 (6) (2011) 1668–1680.

[6] F. Zhao, Y. Huang, L. Wang, T. Tan, Relevance topic model for unstructured social group activity recognition, in: Advances in Neural Information Processing Systems (NIPS), 2013.

[7] T. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Springer, Berlin, 1998.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: European Conference on Computer Vision (ECCV), 2004.

[9] Y. Huang, K. Huang, C. Wang, T. Tan, Exploring relations of visual codes for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2011.

[10] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[11] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition (CVPR), 2005.

[12] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recognit. 29 (1) (1996) 51–59.

[13] S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–137.

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Supervised dictionary learning, in: Advances in Neural Information Processing Systems (NIPS), 2008.

[15] Y. Huang, Z. Wu, L. Wang, T. Tan, Feature coding in image classification: a comprehensive study, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2013) 493–506.

[16] J.C. van Gemert, J.-M. Geusebroek, C.J. Veenman, A.W. Smeulders, Kernel codebooks for scene categorization, in: European Conference on Computer Vision (ECCV), 2008.

[17] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2009.

[18] K. Yu, T. Zhang, Y. Gong, Nonlinear learning using local coordinate coding, in: Advances in Neural Information Processing Systems (NIPS), 2009.

[19] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2010.

[20] S. Gao, I. W. Tsang, L.-T. Chia, P. Zhao, Local features are not lonely – Laplacian sparse coding for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2010.

[21] J. Yang, K. Yu, T. Huang, Efficient highly over-complete sparse coding using a mixture model, in: European Conference on Computer Vision (ECCV), 2010.

[22] N. Kulkarni, B. Li, Discriminative affine sparse codes for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2011.

[23] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, S. Ma, Image classification by non-negative sparse coding, low-rank and sparse decomposition, in: Computer Vision and Pattern Recognition (CVPR), 2011.

[24] S. Gao, L.-T. Chia, I.-H. Tsang, Multi-layer group sparse coding for concurrent image classification and annotation, in: Computer Vision and Pattern Recognition (CVPR), 2011.

[25] L. Cao, R. Ji, Y. Gao, Y. Yang, Q. Tian, Weakly supervised sparse coding with geometric consistency pooling, in: Computer Vision and Pattern Recognition (CVPR), 2012.

[26] Y. Yang, L. Pan, Y. Gao, Y. He, G.N. Zhang, Visual word coding based on difference maximization, Neurocomputing 120 (1) (2013) 277–286.

[27] Y. Huang, K. Huang, Y. Yu, T. Tan, Salient coding for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2011.

[28] K. Yu, T. Zhang, Improved local coordinate coding using local tangents, in: International Conference on Machine Learning (ICML), 2010.

[29] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: Computer Vision and Pattern Recognition (CVPR), 2007.

[30] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: European Conference on Computer Vision (ECCV), 2010.

[31] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Computer Vision and Pattern Recognition (CVPR), 2006.

[32] Y. Huang, Z. Wu, L. Wang, C. Song, Multiple spatial pooling for visual object recognition, Neurocomputing 129 (10) (2014) 225–231.

[33] Y.-L. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: Computer Vision and Pattern Recognition (CVPR), 2010.

[34] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: International Conference on Computer Vision (ICCV), 2011.

[35] Z. Wu, Y. Huang, L. Wang, T. Tan, Group encoding of local features in image classification, in: International Conference on Pattern Recognition (ICPR), 2012.

[36] X. Zhou, K. Yu, T. Zhang, T. S. Huang, Image classification using super-vector coding of local image descriptors, in: European Conference on Computer Vision (ECCV), 2010.

[37] 〈http://pascallin.ecs.soton.ac.uk/challenges/voc/voc2007/index.html〉.

[38] 〈http://www.cs.unc.edu/-lazebnik/research/scenecategories.zip〉.

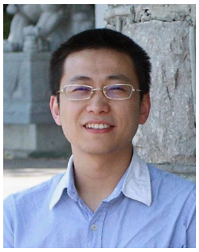
[39] 〈http://www.csie.ntu.edu.tw/cjlin/liblinear/〉.

[40] J. Shlens, A Tutorial on Principal Component Analysis, 2009.

Jingyu Liu received his B.E. degree from Hunan University, Changsha, China, in 2011, and M.E. degree from Beijing Jiaotong University, Beijing, China, in 2014. He is now a Ph.D. candidate in Institute of Automation, Chinese Academy of Sciences (CASIA). His current research interests include pattern recognition, computer vision and machine learning.

Yongzhen Huang received his B.E. degree from Huazhong University of Science and Technology, in 2006, and Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (CASIA), in 2011. In July 2011, he joined the National Laboratory of Pattern Recognition (NLPR), CASIA, where he is currently an Associate Professor. He has published more than 40 papers in the areas of computer vision and pattern recognition at international journals and conferences such as IEEE TPAMI, TSMC-B, CVPR, NIPS, ICIP and ICPR. His current research interests include pattern recognition, computer vision, machine learning and biologically inspired vision computing. He is a member of IEEE.

Liang Wang received both the B. Eng. and M. Eng. degrees from Anhui University in 1997 and 2000 respectively, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), in 2004. From 2004 to 2010, he has been working as a Research Assistant at Imperial College London, United Kingdom and Monash University, Australia, a Research Fellow at the University of Melbourne, Australia, and a lecturer at the University of Bath, United Kingdom, respectively. Currently, he is a full Professor of Hundred Talents Program at the National Lab of Pattern Recognition, CASIA. His major research interests include machine learning, pattern recognition and computer vision. He has widely published at highly-ranked international journals such as IEEE TPAMI and IEEE TIP, and leading international conferences such as CVPR, ICCV and ICDM. He is an associate editor of IEEE Transactions on SMC-B, International Journal of Image and Graphics, Signal Processing, Neurocomputing and International Journal of Cognitive Biometrics. He is currently a Senior Member of IEEE.



Shu Wu received the B.S. degree from Hunan University, China, in 2004, the M.S. degree from Xiamen University, China, in 2007, and the Ph.D. degree from the University of Sherbrooke, Canada, in 2012, all in computer science. He is an assistant professor in the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. His research interests include data mining, recommendation systems, and pervasive computing.