Pacific-Rim Conference on Multimedia 2013

Cross-Codebook Image Classification

Shuai Liao[1,2], Xirong Li[1,2], Xiaoyong Du[1,2,3]

[1]Key Lab of Data Engineering and Knowledge Engineering, MOE, China
[2]School of Information, Renmin University of China, Beijing 100872, China
[3]State Key Lab of Software Development Environment, Beijing 100191, China
{leoshine,xirong,duyong}@ruc.edu.cn

Abstract. Representing images by bag of visual codes (BoVC) features has been the cornerstone of state-of-the-art image classification system. Since the BoVC features depend on a precomputed codebook in use, when the codebook applied to test images differs from the codebook of an existing image classification system, the system becomes inapplicable. To resolve the codebook incompatibility problem, we propose in this paper *cross-codebook* image classification. This is achieved by transforming BoVC features derived from one codebook to make them compatible with another codebook. Two BoVC transform methods, i.e., code-reassignment and least squares, are studied. Experiments on a popular image classification benchmark set show that both methods are better than random guess when crossing the codebooks. In particular, when the BoVC features are transformed from a higher dimension to a relatively small dimension, cross-codebook image classification has a similar performance compared to within-codebook image classification, with a relative performance loss of 1.3% only. The results justify the feasibility of the proposed cross-codebook image classification.

Keywords: Image classification, bag of visual codes, cross-codebook

1 Introduction

Automatically classifying images into a set of predefined visual categories such as 'beach', 'building', and 'food' is crucial for organizing and retrieving the ever-growing amounts of images on personal devices and the Internet. Image classification is thus a focal topic in multimedia computing and applications.

The state-of-the-art in image classification is to learn Support Vector Machines classifiers from manually labeled examples represented by bag of visual codes (BoVC) features [1–3]. Since the seminal work by Csurka *et al.* [4] in 2004, BoVC features have been the *de facto* choice for image classification.

To extract a BoVC feature vector for a given image, a number of local descriptors, e.g., SIFT [5], are first extracted from image patches. These descriptors are then quantized by a precomputed *codebook*. The codebook is constructed by conducting k-means clustering on many local descriptors, with each cluster center corresponding to a specific visual code. Consequently, the given image is

represented by a histogram with its dimensionality equal to the size of the codebook. Each bin of the histogram corresponds to a visual code, and its value is the accumulated frequency of that code. Therefore, the extraction of BoVC features depend on the codebook in use.

When the codebook applied to extract BoVC from a test image set is different from the codebook with which a classification system is built, the existing system is inapplicable to classify the test images. We term this problem as *codebook incompatibility*, and conceptually illustrate it in Fig. 1.

One might argue to avoid the codebook incompatibility problem by sticking to one codebook. For instance, use the codebook of the existing classification system to re-extract BoVC features of the test data. However, this solution is unfeasible when the original test images are inaccessible because of privacy or copyright concerns, or they are simply no longer available due to varied reasons including the limit in storage. Consider for instance the NUS-WIDE [6], a popular dataset for consumer photo classification. Due to the copyright concern, the dataset releases BoVC features, without providing the original images. As a consequence, one cannot directly use this dataset to test the effectiveness of an image classification system at hand. In addition, as re-extracting BoVC features for large-scale datasets is cumbersome [3], a lightweight solution to codebook incompatibility is desirable.

In order to make BoVC features derived from one codebook compatible with another codebook, we propose in this paper cross-codebook image classification, as shown in Fig. 1. In particular, we study BoVC transform methods, given that the local descriptors are of the same type. Cross-codebook image classification is, to the best of our knowledge, non-existing in the literature.

The rest of the paper is organized as follows. We elaborate in Section 2 the proposed BoVC transform methods, followed by cross-codebook image classification experiments in Section 3. Conclusions are given in Section 4.

2 BoVC Transform for Cross-Codebook Classification

Given an image classification system based on one codebook C_s and a set of test images with their BoVC features w.r.t another codebook C_t, we cannot directly employ the system to classify the test data due to the incompatibility between C_s and C_t. For varied reasons as discussed in Section 1, re-extracting BoVC features of the test data using C_s or re-training the system with C_t might be infeasible. In that regard, we study how to transform the BoVC features of the test data into new features that are compatible with C_s, and thus can be accepted by the existing classification system.

2.1 Problem Formalization

To make our discussion more formal, we first introduce a few notations. Let $C_t = \{c_{t1}, c_{t2}, \ldots, c_{tk_t}\}$ be the codebook with which BoVC features of the test data are extracted, where c_{ti} denotes the vector of the i-th code, and k_t is the size

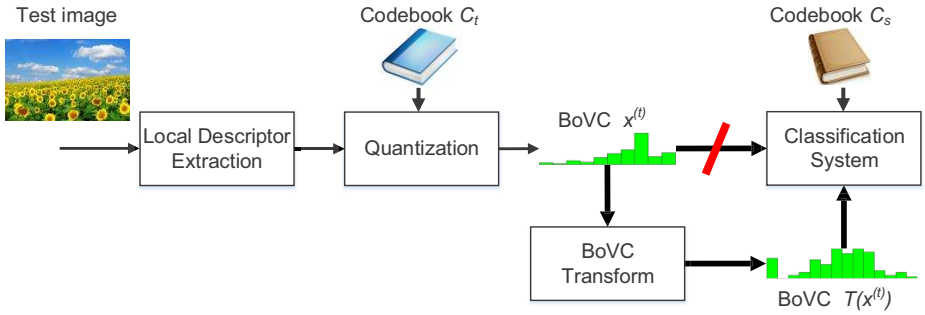

Fig. 1. Illustrating the codebook incompatibility problem in bag-of-visual-codes (BoVC) based image classification. When the codebook (C_t) applied to a test image differs from the codebook (C_s) upon which a classification system is built, the existing system is inapplicable to classify the test image. In order to make BoVC features w.r.t C_t compatible with C_s, we propose in this paper cross-codebook image classification.

of the codebook. In a similar fashion, we denote the codebook for the existing image classification system as $C_s = \{c_{s1}, c_{s2}, \ldots, c_{sk_s}\}$.

Let x be a specific image. Its BoVC feature quantized by C_t is denoted by $x^{(t)} = (x_1^{(t)}, \ldots, x_{k_t}^{(t)})$, and $x^{(s)} = (x_1^{(s)}, \ldots, x_{k_s}^{(s)})$ indicates the BoVC feature quantized by C_s. For a given category ω, let $H(x^{(s)}, \omega)$ be its classifier. As aforementioned, this classifier is built on the base of the codebook C_s, so it cannot handle $x^{(t)}$.

To resolve the codebook incompatibility problem, we look for a *transform* function $T : \mathcal{R}^{k_t} \rightarrow \mathcal{R}^{k_s}$ such that classification with the new feature $T(x^{(t)})$ would have similar performance compared to classification with $x^{(s)}$, i.e.,

$$\arg_T sign(H(T(x^{(t)}), \omega)) = sign(H(x^{(s)}, \omega)), \tag{1}$$

where $sign : \mathcal{R} \rightarrow \{1, -1\}$ is a decision function.

Notice that Eq. (1) states an ideal case of BoVC transform. In this paper we instantiate the transform function T as a linear mapping for its simplicity and efficiency. In particular, we define

$$T(x^{(t)}) := x^{(t)} A, \tag{2}$$

where A is a $k_t \times k_s$ matrix. By doing so, cross-codebook image classification boils down to finding the matrix A.

Next, we give two BoVC transform methods, which compute A by visual code re-assignment and by least squares, respectively.

2.2 BoVC Transform Methods

As we have discussed in Section 1, if the codebook C_t is simply a re-order of the codebook C_s, codebook incompatibility is resolved by re-assigning the i-th bin

of $x^{(t)}$ to the bin corresponding to c_{sj}, the code in C_s that is the closest to c_{ti}. In this case, A in Equation (2) is a binary matrix. In practice, however, such a hard re-assignment method would lose much information. To that end, we first introduce a *soft re-assignment* method which generates the new feature $T(x^{(t)})$ by allocating each dimension of $x^{(t)}$ in terms of visual code similarities.

Method I: Code Re-assignment. Given two visual codes c_{ti} and c_{sj}, let $d(c_{ti}, c_{sj})$ be the Euclidean distance. To convert the distance into a similarity score, we use a Gaussian function, and consequently the similarity function between the two codes is defined as $\mathcal{K}(c_{ti}, c_{sj}) = \exp(-\frac{d(c_{ti},c_{sj})}{\rho})$. The bandwidth parameter ρ is empirically set to be the mean of the distances of all code pairs.

In the code re-assignment method, all dimensions of $x^{(t)}$ can contribute to each dimension of $T(x^{(t)})$, with more importance given to closer codes. In particular, we compute the j-th dimension of the transformed feature as

$$T_j(x^{(t)}) = \sum_{i=1}^{k_t} x_i^{(t)} A_{ij}, \tag{3}$$

where the matrix element A_{ij}, computed as

$$\frac{\mathcal{K}(c_{ti}, c_{sj})}{\sum_{l=1}^{k_s} \mathcal{K}(c_{ti}, c_{sl})}, \tag{4}$$

determines the amount of c_{ti} that is assigned to the bin corresponding to c_{sj}.

As shown in Eq. (4), the code re-assignment method only utilizes the two codebooks to compute A, without explicitly considering the connection between $T(x^{(t)})$ and $x^{(s)}$. Given that extracting $x^{(s)}$ from the test data is unfeasible, as an alternative, we leverage an auxiliary image set for which BoVC features quantized by distinct codebooks can be extracted. In that regard, we propose the second *least square* method as follows.

Method II: Least Squares. In order to distinguish the auxiliary set from the test set in consideration, we use $\mathbf{z}$ to denote an auxiliary image, and let $\{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ be the set of m auxiliary images. Assuming that the BoVC feature $\mathbf{z}^{(s)}$ can be fully reconstructed from $\mathbf{z}^{(t)}$ by A, we have a system of m linear equations, i.e.,

$$\mathbf{z}_i^{(s)} = A \cdot \mathbf{z}_i^{(t)}, \qquad i = 1, \ldots, m. \tag{5}$$

When $m > k_t$, Eq. (5) is mostly an overdetermined system, meaning none exist of an exact solution. Therefore, we resort to least squares, a classical approach to approximated solutions of overdetermined systems [7]. Accordingly, we define the objective function as

$$\underset{A}{\operatorname{argmin}} \sum_{i=1}^{m} ||A \cdot \mathbf{z}_i^{(t)} - \mathbf{z}_i^{(s)})||^2. \tag{6}$$

Notice that the least squares method requires no labeled examples, so the auxiliary set can be obtained, e.g., from the Internet, with ease.

3 Empirical Study

3.1 Experimental Setup

Data sets. We use the ImageCLEF 2010 development set [8], a leading benchmark for image classification. The set consists of two prespecified subsets, one subset with 5,000 images for training, and the other subset with 3,000 images for testing. All the images were collected from Flickr, with ground truth available for 93 visual concepts. To resolve the optimization problem as defined in Equation (6), we downloaded a number of Flickr images as the source of *unlabeled* auxiliary data, without using its annotations.

BoVC feature extraction. To extract the BoVC feature for each image, we use the color descriptor software [3] to extract the SIFT descriptor in a dense manner. In order to simulate a cross-codebook scenario, the descriptors of a training image are quantized by the codebook C_s, while the descriptors of a test image are quantized by the codebook C_t different from C_s.

Image classification. We adopt the fast intersection kernel SVMs (FikSVMs) for its effectiveness and efficiency [9]. The key technique for FikSVMs comes from the observation that the decision function of intersection kernel SVMs can be expressed as the sum of decision functions with respect to individual feature dimensions. For each dimension, its decision function can be efficiently computed by linear interpolation on a limited set of precomputed points. As a consequence, classification boils down to a few table-lookup operations, and thus becomes very fast. In this work, we use the implementation of FikSVM by Li *et al.* [10].

Notice that for most of the 93 concepts, their positive training examples are in minority. Hence, to address class imbalance for training SVMs, the positive class and the negative class are assigned with different cost parameters in light of the reciprocal of their distribution in the training data.

Evaluation criteria. For each concept, we report the widely used Average Precision. To measure the overall performance, we use mean Average Precision (mAP).

3.2 Experiments

To verify the effectiveness of the two BoVC transform methods, the codebook C_t and the codebook C_s were independently generated by performing k-means clustering on random subsets of SIFT descriptors extracted from the training set. For comparison, we build a within-codebook image classification system using C_s only. We also build a random guess run, which ranks the test set at random. We take the mean score of multiple random guess runs. While random guess with an mAP of 0.129 is a performance lower bound, the within-codebook system establishes a performance upper bound on cross-codebook image classification.

We construct a number of distinct codebooks, by executing k-means clustering multiple times on different sets of local descriptors. Depending on whether the size of C_t is equal to the size of C_s, we divide our experiments into two parts, that is, BoVC transform with $k_t = k_s$ and BoVC transform with $k_t \neq k_s$.

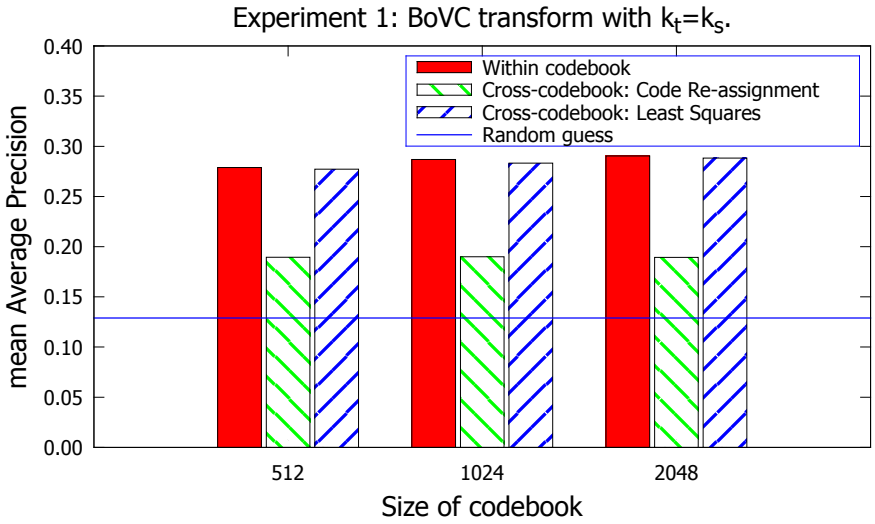

Fig. 2. Experiment 1: BoVC transform with $k_t = k_s$. With the least squares BoVC transform method, the cross-codebook image classification system is comparable to the within-codebook system.

Experiment 1: BoVC transform with $k_t = k_s$. In this experiment, BoVC features derived from the two distinct codebooks are of the same dimensionality. We try codebooks of varied sizes by choosing k_t and k_s from {512; 1,204; 2,048}.

As shown in Fig. 2, both the code re-assignment method and the least squares method outperform the random guess run. Comparing the two methods, least squares is clearly better than code re-assignment. We attribute this to the fact that by exploiting extra data, the least squares method finds a better mapping between BoVC features derived from C_s and C_t. Moreover, using the mapping found by solving least squares, the cross-codebook classification system is comparable to the within-codebook system, with a relative performance loss of 1.3% at most. From the above results we conclude that when the two codebooks are of equal size, the codebook incompatibility problem can be well resolved by the least squares method, and consequently cross-codebook image classification is doable.

Experiment 2: BoVC transform with $k_t \neq k_s$. To study BoVC transform between codebooks of different sizes, we fix k_s to be 512, and choose k_t from {64; 256; 1,024; 2,048; 4,096}.

As shown in Fig. 3, the least squares method is again better than the code re-assignment method. When $k_t < k_s$, the performance of both methods improves as k_t increases, but is lower than the within-codebook run, which has an mAP of 0.289. This is because we are mapping BoVC features of the test data from

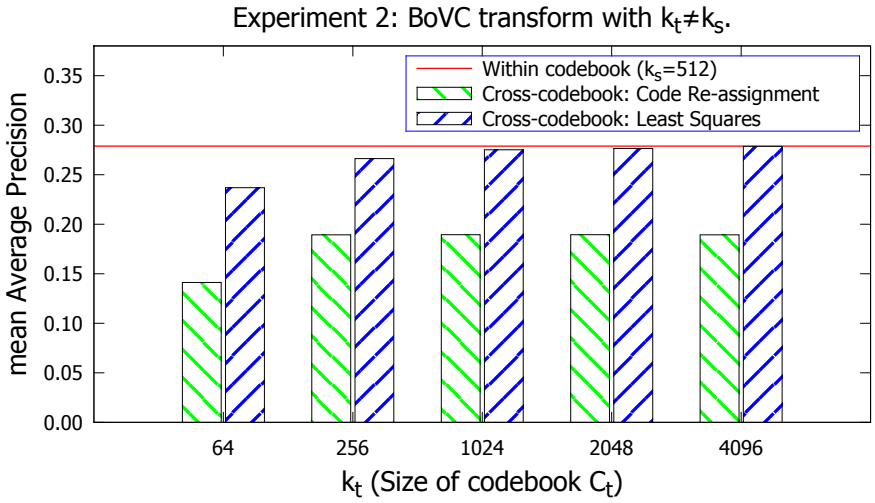

Fig. 3. Experiment 2: BoVC transform with $k_t \neq k_s$. We fix the size of C_s to be 512, and vary the size of C_t. When the size of C_t is larger than the size of C_s, the codebook incompatibility problem can also be well addressed by least squares BoVC transform.

a lower dimensions, e.g., 64 or 256, to the higher dimension, i.e., 512, but the transform methods do not bring in novel information about the original images.

When $k_t > k_s$, while the performance of the code re-assignment method remains lower than the performance of the within-codebook run, the cross-codebook runs with the least squares method are close to the within-codebook run. Our explanation is that while mapping BoVC features of the test data from a higher dimension, e.g., 2,048 or 1,024 to the lower dimension, i.e, 512, also causes information loss, the transformed BoVC features seem adequate for the existing image classification system. Therefore, when the size of C_t is larger than the size of C_s, the codebook incompatibility problem can also be well addressed by least squares BoVC transform.

4 Conclusions

In this paper we extend the research on bag of visual codes (BoVC) based image classification in a new dimension, namely cross-codebook image classification. When the codebook applied to test images differs from the codebook of an existing image classification system, the system is inapplicable to classify the test images. To tackle the codebook incompatibility problem, we study two BoVC transform methods, i.e., code re-assignment and least squares.

From experiments on the ImageCLEF 2010 development set we draw the following conclusions. While both methods are better than random guess, the least squares method is superior to its code re-assignment counterpart. When the size of the codebook of the test data is no smaller than the size of the codebook of the existing classification system, cross-codebook classification with BoVC transformed by the least squares method is found to be comparable to within-codebook classification, with a relative performance loss of 1.3% at most. Given local descriptors of the same type, cross-codebook image classification is feasible, and the least squares method is our recommendation.

For future work, it will be interesting to evaluate the effectiveness of the proposed BoVC transformation methods for other image analysis tasks such as image retrieval and copy detection. Another challenging extension is to consider cross-codebook image classification for local descriptors of different types.

Acknowledgments. This research was supported by the Basic Research funds in Renmin University of China from the central government (13XNLF05), State Key Laboratory of Software Development Environment Open Fund (SKLSDE-2012KF-09), and National Natural Science Foundation of China (No. 61303184).

Bibliography

[1] Perronnin, F., Akata, Z., Harchaoui, Z., Schmid, C.: Towards good practice in large-scale learning for image classification. In: CVPR. (2012) 3482–3489

[2] Jiang, Y.G., Yang, J., Ngo, C.W., Hauptmann, A.: Representations of keypoint-based semantic concept detection: A comprehensive study. IEEE Transactions on Multimedia **12**(1) (2010) 42–53

[3] van de Sande, K., Gevers, T., Snoek, C.: Empowering visual categorization with the gpu. IEEE Transactions on Multimedia **13**(1) (2011) 60–70

[4] Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV Workshop on Statistical Learning in Computer Vision. Volume 1. (2004) 22

[5] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110

[6] Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: CIVR. (2009) 48

[7] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer (2001)

[8] Nowak, S., Huiskes, M.: New strategies for image annotation: Overview of the photo annotation task at ImageCLEF 2010. In: CLEF. (2010)

[9] Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR, IEEE (2008) 1–8

[10] Li, X., Snoek, C., Worring, M., Koelma, D., Smeulders, A.: Bootstrapping visual categorization with relevant negatives. IEEE Transactions on Multimedia **15**(4) (2013) 933–945