LEARNING A DISCRIMINATIVE VISUAL CODEBOOK USING HOMONYM SCHEME

SeungRyul Baek, Chang D. Yoo and Sungrack Yun

Dept. of Electrical Engineering, KAIST,
373-1, Guseong Dong, Yuseong Gu, Daejeon, 305-701, Korea
mudge@kaist.ac.kr, cdyoo@ee.kaist.ac.kr, yunsungrack@kaist.ac.kr

ABSTRACT

This paper studies a method for learning a discriminative visual codebook for various computer vision tasks such as image categorization and object recognition. The performance of various computer vision tasks depends on the construction of the codebook which is a table of visual-words (*i.e.* codewords). This paper proposed a learning criterion for constructing a discriminative codebook, and it is solved by the homonym scheme which splits codeword regions by labels. A codebook is learned based on the proposed homonym scheme such that its histogram can be used to discriminate objects of different labels. The traditional codebook based on the k-means is compared against the learned codebook on two well-known datasets (Caltech 101, ETH-80) and a dataset we constructed using google images. We show that the learned codebook consistently outperforms the traditional codebook.

Index Terms— Computers and information processing, Image processing, Machine vision, Object recognition, Bag-of-words model.

1. INTRODUCTION

With the bag-of-words (BOW) model, documents can be represented as an unordered collection of words, disregarding grammar and even word order in the form of histograms. For document classification, its probabilistic relationship with document and topic/subtopic is modeled using probabilistic latent semantic analysis (pLSA) [1] and latent dirichlet allocation (LDA) [2]. Recently, the BOW model has been used to represent an image [3, 4]. As in document classification [5], images are represented as histograms of visual-words describing only their appearance while ignoring their spatial structure.

The classification performance using the BOW depends mainly on the word dictionary and the classification algorithm. The focus of this paper is on the construction of word dictionary. The relationship between a document and a word is clearly defined in the text domain since document is composed of words from a word dictionary that we all use. However, the relationship between an image and a visual-word is not obvious since image is not a composition of visual-words that we all know and understand. In previous works [3, 4, 6], clustering algorithms such as k-means and mean-shift are applied to construct the codebooks.

This work was supported (National Robotics Research Center for Robot Intelligence Technology, KAIST) by Ministry of Knowledge Economy under Human Resources Development Program for Convergence Robot Specialists. This work was conducted under the research at the Personal Plug&Play DigiCar Research Center at KAIST which was supported by the National Research Foundation of Korea Grant funded by the Korean Government (No.2010-0028680).

Constructing visual-word dictionary (*i.e.* codebook) is one of the most important issues in classification tasks using the BOW model in the image domain, and several approaches have recently been proposed to obtain a discriminative codebook [7–12]. Perronnin et al. [7] construct an adaptive vocabulary by combining class specific codebooks and universal codebook. Farquhar et al. [8] construct a discriminative codebook using the maximum a posterior (MAP) estimates of the gaussian mixture model (GMM). Moosmann et al. [9] construct discriminative visual-word vocabularies using randomized clustering forests. Winn et al. [10] construct a compact and discriminative vocabulary by pair-wise merging visual-words as long as merging does not degrade the performance much. Gemert et al. [11] model ambiguities in clustering algorithms and get more effective codebook. Li et al. [12] measure similarities between codewords by Kullback Leibler (KL) divergence, and merge codewords the similarities are high.

In this paper, a learning criterion which can improve the histogram intersection kernel (HIK) based classification performance is studied. Due to difficulties in optimizing the proposed criterion, *homonym scheme* is proposed as a solution. The scheme construct a codebook as follows: 1) it measures familiarities between codewords in a given codebook and labels. 2) it splits codewords, which are not familiar with a certain label, by incorporating label information. We show that this scheme will increase the classification performance. Also, this can be easily applied to the above mentioned methods to improve classification performance using the HIK. The traditional codebook based on k-means is compared against the learned codebook using three datasets (Caltech-101, ETH-80, face/non-face images obtained as a result of OpenCV face detector using google images). We demonstrated that the learned codebook consistently outperforms the traditional codebook.

This paper is organized as follows. Section 2 proposes a novel learning criterion and methods to generate a codebook which satisfies this criterion. Section 3 shows the performance of the proposed method on three datasets. Section 4 concludes the paper.

2. PROPOSED METHOD

Various generative models such as pLSA and LDA and discriminative models such as support vector machine (SVM) can be used as a classifier using the BOW model. It has been recently shown that the HIK presents better results and operates faster than the Euclidean distance in supervised learning tasks with histogram features [13]. Also, Wu et al. [14] state that when histogram features are employed, the HIK should be used. As a consequence, classifiers using the HIK such as histogram intersection kernel SVM (IKSVM) are popularly used for various tasks when using the BOW model [11, 13, 14].

2.1. The Histogram Intersection Kernel

Let $\mathbf{x} = [x_1, ..., x_K]$ and $\mathbf{x}' = [x'_1, ..., x'_K] \in \mathbb{R}_+^K$ be two different K-dimensional histogram, where x_i and x'_i and $\mathbb{R}_+^K$ denote the ith bin of $\mathbf{x}$, $\mathbf{x}'$ and the K-dimensional positive real number space. The HIK of $\mathbf{x}$ and $\mathbf{x}'$ is defined as:

$$K_{HI}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{K} \min(x_i, x'_i) \tag{1}$$

In [15], it is proven that K_{HI} is a valid positive-definite kernel when x_i and $x'_i \geq 0$ for $i = 1, ..., K$. This means there exists an underlying mapping ϕ associated with K_{HI} such that $\mathbf{x}$ maps to a corresponding $\phi(\mathbf{x})$ in a high dimensional feature space Φ (*i.e.* $\phi : \mathbb{R}_+^K \rightarrow \Phi$), such that $K_{HI}(\mathbf{x}, \mathbf{x}')$ is equivalent to the inner product of two histograms in Φ [14]. Therefore, the kernel trick can be used to incorporate K_{HI} in the SVM framework.

2.2. Learning Criterion

The goal of this paper is to construct a discriminative codebook, which leads to high classification accuracy. In this subsection, the criterion satisfying the goal is derived. Kernel functions, which measure similarities between two input data, should be minimized when the corresponding labels are different and maximized when the labels are same. Thus, if $\mathbf{x}$ and $\mathbf{x}'$ are histogram features for images with different labels, minimizing $K_{HI}(\mathbf{x}, \mathbf{x}')$ will improve the performance of classification task based on the HIK.

Since $\mathbf{x}$ and $\mathbf{x}'$ are normalized histograms, $\|\mathbf{x}\|_1 = \|\mathbf{x}'\|_1 = 1$, where $\|\mathbf{x}\|_1$ and $\|\mathbf{x}'\|_1$ denote L_1-norm of $\mathbf{x}$ and $\mathbf{x}'$ respectively. This leads to the following formulation:

$$\begin{aligned} & \|\mathbf{x}\|_1 + \|\mathbf{x}'\|_1 \\ = \; & 2 \cdot K_{HI}(\mathbf{x}, \mathbf{x}') + \|\mathbf{x} - \mathbf{x}'\|_1 \\ = \; & 2. \end{aligned} \tag{2}$$

Notice that maximizing $\|\mathbf{x} - \mathbf{x}'\|_1$ is equivalent to minimizing $\kappa_{HI}(\mathbf{x}, \mathbf{x}')$. However, maximizing L_1 distances between all histogram pairs of different labels while minimizing the distances between all histogram of the same label may be impossible to do. Instead, we will maximize L_1 distance between average histograms of different labels.

Let $I_i^j \in \mathcal{I}$ for $i = 1, ..., N_j$ be the ith training image in the jth label, where $\mathcal{I}$ denotes a 2-dimensional image space. Let $\mathbf{H_C} : \mathcal{I} \rightarrow \mathbb{R}_+^K$ be a function which maps a 2-dimensional image into a K-dimensional histogram using codebook $\mathbf{C}$. The histogram of the ith image in the jth label is expressed as:

$$\mathbf{H_C}(I_i^j) = [H_{\mathbf{C}1}(I_i^j), ..., H_{\mathbf{C}K}(I_i^j)]. \tag{3}$$

In this paper, a binary classification task is considered. Assume that there are two labels $j = A$ and $j = B$. Then, learning criterion can be formulated as follows:

$$\begin{aligned} \mathbb{L}(\mathbf{C}) : \mathbf{C^{new}} = \arg\max_{\mathbf{C}} \; & \|\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^A) - \bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^B)\|_1 \\ \text{subject to} \; & \|\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^A)\|_1 = 1, \\ & \|\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^B)\|_1 = 1 \end{aligned}$$

where

$$\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^A) = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{H_C}(I_i^{j=A}), \tag{4}$$

$$\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^B) = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{H_C}(I_i^{j=B}). \tag{5}$$

Table 1. Types of words in the image domain

Homonym	Similar appearance, Different Label
Synonym	Different appearance, Same Label [12]

When function $\mathbf{H_C}$ can be explicitly expressed as a function of the parameter vector $\mathbf{C}$, $\mathbb{L}(\mathbf{C})$ can be optimized by conventional optimization methods and $\mathbf{C}$ can be obtained. However, since it is difficult to get an explicit form for $\mathbf{H}$ as a function of $\mathbf{C}$, optimizing $\mathbb{L}(\mathbf{C})$ is a difficult problem.

Instead, we obtain the following inequalities using the fact that $\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^j)$ are normalized K-dimensional vector whose L_1-norm is 1:

$$0 \leq \|\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^A) - \bar{\mathbf{H}}_{\mathbf{C}}\{\mathbf{I}^B)\|_1 \leq 2 \tag{6}$$

and

$$\begin{aligned} & \|\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^A) - \bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^B)\|_1 \\ & \qquad = \sum_{i=1}^{K} |\bar{H}_{\mathbf{C}i}(\mathbf{I}^A) - \bar{H}_{\mathbf{C}i}(\mathbf{I}^B)| \qquad (7) \\ & \qquad \leq \sum_{i=1}^{K} (\bar{H}_{\mathbf{C}i}(\mathbf{I}^A) + \bar{H}_{\mathbf{C}i}(\mathbf{I}^B)) = 2 \qquad (8) \end{aligned}$$

where $\bar{H}_{\mathbf{C}i}(\cdot)$ denotes the ith bin of $\bar{\mathbf{H}}_{\mathbf{C}}(\cdot)$. The upper bound of Eq.(7) is Eq.(8). If we make Eq.(7) close to Eq.(8), object function will be maximized. Notice that converting *minus* sign in Eq.(7) to *plus* sign makes Eq.(7) closer to its upper bound. This intuition motivated the following method.

2.3. Suboptimal Solution

2.3.1. Homonym scheme

Most methods for constructing codebook in the past are based on gradient-based features (*e.g.* SIFT, SURF, HOG) to distinguish objects of different labels. Since gradient-based features use appearance information, similarly-shaped patches of different labels have similar feature values and they will be assigned to the same codeword. Of course, this assignment should be allowed, however, lots of these assignments degrade the classification performance using the BOW model. In this paper, codewords which have many of those assignments are called *homonyms*. In Table 1, word type is shown according to the label and appearance in the image domain. Homonym scheme assigns these features into different codewords by splitting original codeword to produce histograms that is more discriminative.

2.3.2. How homonym scheme satisfies learning criterion

In this section, we describe homonym scheme by an example. There are 7 codewords: $\bar{H}_{\mathbf{C}1}, .., \bar{H}_{\mathbf{C}7}$. $\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^A)$ and $\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^B)$ are average histograms for images of label A and B using codebook $\mathbf{C}$. $\bar{H}_{\mathbf{C}1}$, $\bar{H}_{\mathbf{C}2}$ and $\bar{H}_{\mathbf{C}3}$ are frequently appeared codewords for label A, while $\bar{H}_{\mathbf{C}5}$ and $\bar{H}_{\mathbf{C}6}$ are frequently appeared codewords for label B. Also, $\bar{H}_{\mathbf{C}4}$ is ambiguous codewords (*i.e.* homonym) since it appears in label A and B with similar frequency. As in Fig.1(b), we split $\bar{H}_{\mathbf{C}4}$ into two regions according to the label (one is for label A, another is for label B). In Fig.1(a), the average histograms of label A and B before splitting $\bar{H}_{\mathbf{C}4}$ are shown and those after splitting $\bar{H}_{\mathbf{C}4}$ are shown in Fig.1(c). Before splitting the region of $\bar{H}_{\mathbf{C}4}$, Eq.(7) for 4th

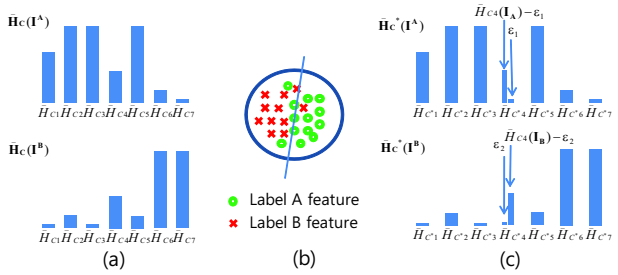


Fig. 1. (a) Before homonym scheme applied. (b) Features associated with a homonym codeword into distinct label regions. (c) After homonym scheme applied.

codeword is expressed as follow:

$$\begin{aligned} & |\bar{H}_{\mathbf{C}4}(\mathbf{I}^A) - \bar{H}_{\mathbf{C}4}(\mathbf{I}^B)| \\ = \; & \max(\bar{H}_{\mathbf{C}4}(\mathbf{I}^A), \bar{H}_{\mathbf{C}4}(\mathbf{I}^B)) - \min(\bar{H}_{\mathbf{C}4}(\mathbf{I}^A), \bar{H}_{\mathbf{C}4}(\mathbf{I}^B)) \\ = \; & \bar{H}_{\mathbf{C}4}(\mathbf{I}^A) - \bar{H}_{\mathbf{C}4}(\mathbf{I}^B). \end{aligned} \tag{9}$$

After splitting the region of $\bar{H}_{\mathbf{C}4}$, it becomes:

$$\begin{aligned} & |\bar{H}_{\mathbf{C}^*\mathbf{4}}(\mathbf{I}^A) - \bar{H}_{\mathbf{C}^*\mathbf{4}}(\mathbf{I}^B)| \\ = \; & |(\bar{H}_{\mathbf{C}4}(\mathbf{I}^A) - \epsilon_1) - \epsilon_2| + |\epsilon_1 - (\bar{H}_{\mathbf{C}4}(\mathbf{I}^B) - \epsilon_2)| \\ = \; & \bar{H}_{\mathbf{C}4}(\mathbf{I}^A) - \epsilon_1 - \epsilon_2 + \bar{H}_{\mathbf{C}4}(\mathbf{I}^B) - \epsilon_2 - \epsilon_1 \end{aligned} \tag{10}$$

where ϵ_1 and ϵ_2 are errors that occur in splitting, and $\bar{H}_{\mathbf{C}^*\mathbf{4}}(\cdot)$ denotes the 4th bin in the histogram $\bar{\mathbf{H}}_{\mathbf{C}^*}(\cdot)$ using the new codebook $\mathbf{C}^*$. If ϵ_1 and ϵ_2 are small, it leads to the following:

$$\begin{aligned} & |\bar{H}_{\mathbf{C}^*4}(\mathbf{I}^A) - \bar{H}_{\mathbf{C}^*4}(\mathbf{I}^B)| \\ \approx \; & \max(\bar{H}_{\mathbf{C}4}(\mathbf{I}^A), \bar{H}_{\mathbf{C}4}(\mathbf{I}^B)) + \min(\bar{H}_{\mathbf{C}4}(\mathbf{I}^A), \bar{H}_{\mathbf{C}4}(\mathbf{I}^B)) \\ = \; & \bar{H}_{\mathbf{C}4}(\mathbf{I}^A) + \bar{H}_{\mathbf{C}4}(\mathbf{I}^B). \end{aligned} \tag{11}$$

The right-hand-side of Eq.(10) is larger than that of Eq.(9) when $-\epsilon_1 - \epsilon_2 + \bar{H}_{\mathbf{C}4}(\mathbf{I}^B) - \epsilon_2 - \epsilon_1 \geq -\bar{H}_{\mathbf{C}4}(\mathbf{I}^B)$, which is equal to:

$$\bar{H}_{\mathbf{C}4}(\mathbf{I}^B) = \min(\bar{H}_{\mathbf{C}4}(\mathbf{I}^A), \bar{H}_{\mathbf{C}4}(\mathbf{I}^B)) \geq \epsilon_1 + \epsilon_2. \tag{12}$$

This shows that proper splitting makes Eq.(7) closer to its upper bound, which satisfies the proposed learning criterion.

Also, in this example, the difference between Eq.(9) and Eq.(10) is approximately $2 \cdot \min(\bar{H}_{\mathbf{C}4}(\mathbf{I}^A), \bar{H}_{\mathbf{C}4}(\mathbf{I}^B))$ by Eq.(11). Thus, the difference will be maximized if we split the ith bin whose $\min(\bar{H}_{\mathbf{C}i}(\mathbf{I}^A), \bar{H}_{\mathbf{C}i}(\mathbf{I}^B))$ is the largest. However, since $\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^A)$ and $\bar{\mathbf{H}}_{\mathbf{C}}(\mathbf{I}^B)$ are average histograms, this criterion is not appropriate, thus the posterior probability of a label given a codeword is used for determining the codewords to be split.

2.4. Codebook Learning Algorithm

The learning algorithm is summarized into 4 steps:

1. The posterior probability of a label given a codeword is calculated for the input codebook as follow:

$$p(L = A|w_i) = \frac{p(w_i|L = A)p(L = A)}{\sum_l p(w_i|L = l)p(L = l)}. \tag{13}$$

When $p(L = A|w_i)$ is high, it means that the ith codeword frequently appears in images of label A, while the low value means that the codeword frequently appears in images of label B.

2. Determine the codewords to be split. Notice that codewords whose $p(L = A|w_i)$ close to $\frac{1}{2}$ should preferentially be split. The decision rule for determining codewords to be split is as follow:

$$i^* = \arg\min_i |p(L = A|w_i) - 0.5|, 1 \leq i \leq K. \tag{14}$$

3. Split i^*th codeword in the initial codebook and split another codeword in the obtained codebook which satisfies Eq.(14) iteratively. The SVM is used to split the homonyms. At each splitting step, the SVM parameters for the target homonym are calculated, and the codebook size is increased by 1. If this is performed β times, the size of the codebook becomes K+β. In many cases, codewords are not linearly separable according to their labels, thus radial basis function (RBF) kernel defined below is used for good performance:

$$\kappa_{RBF}(\mathbf{u}, \mathbf{v}) = e^{-\gamma|\mathbf{u}-\mathbf{v}|^2} \tag{15}$$

4. We map test features to the j^*th codeword in the obtained codebook using the following rule

$$j^* = \arg\min_j \|\mathbf{f} - \mathbf{m_j}\|_2, 1 \leq j \leq K \tag{16}$$

where $\mathbf{m_j}$ is the center of codeword j and $\mathbf{f}$ is a test feature. If j^*th codeword is split, the test feature is further decided where to be mapped using pre-calculated SVM parameters.

3. EXPERIMENT

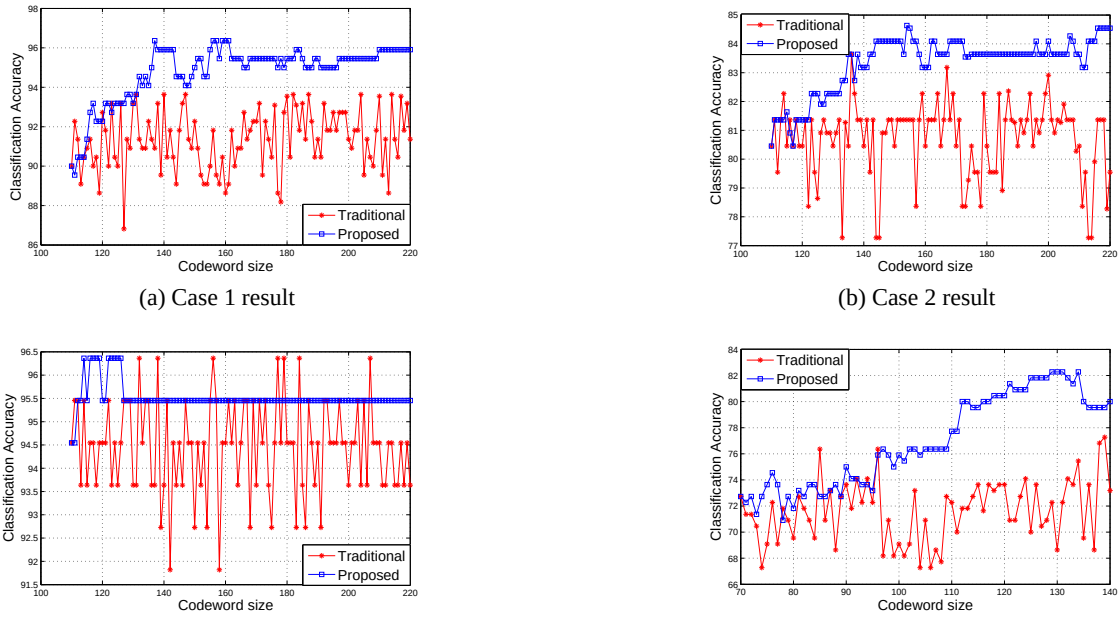
3.1. Experimental Setting

We use LIBSVM [16], LIBHIK [13, 17] library to split codewords and to evaluate the performance of the obtained codebook using ETH-80 [18] and Caltech101 [19] databases and additional face/non-face image sets obtained using OpenCV face detector on google images. Also, we use the SIFT feature implemented in [20]. Details of experiments are described below. Parameter K and the number of training images are set according to the number of total features such that make each codeword has approximately 1000 training features. We use 25% of training data as a validation set, and select γ which yields the best performance using the validation set, differently for each codeword.

- Case 1: Classify apple and pear images from ETH-80, 600 training images and 220 test images. Initial K=110.
- Case 2: Classify apple and tomato images from ETH-80, 600 training images and 220 test images. Initial K=110.
- Case 3: Classify face and background images from Caltech-101, 220 training images and 110 test images. Initial K=110.
- Case 4: Classify face and non-face images resulted from OpenCV face detector, 600 training images and 220 test images. Initial K=70.

3.2. Experimental Results

Fig.2. shows the experimental results for 4 different cases. The classification accuracy is shown according to the size of codebook. Since splitting codewords increases the size of codebook by 1, we compare the traditional k-means approach by increasing the value of K. As codeword size increases, the proposed method improves the accuracy for case 1 and 2 by about 3-4%, and for case 4 by about 8-9% compared to the traditional approach. For case 3, the accuracy of traditional approach is quite high, and it is difficult to obtain remarkable improvements in the classification accuracy.



(a) Case 1 result

(b) Case 2 result

(c) Case 3 result

(d) Case 4 result

Fig. 2. Classification accuracy is shown as the codeword size is changed for 4 cases.

4. CONCLUSION

This paper proposed a learning criterion that maximizes L_1 distance between average histograms of different labels, which can improve the performance of classifiers using the HIK. Due to difficulties in optimizing the proposed criterion, the *homonym scheme* is proposed as a solution. We demonstrated that it leads to a more discriminative codebook than the codebook without using the homonym scheme.

5. REFERENCES

[1] T.Hoffman, "Unsupervised learning by probabilistic latent semantic analysis," in *Machine Learning*, 2001.

[2] D.M.Blei, A.Y.Ng, and M.I.Jordan, "Latent dirichlet allocation," in *JMLR*, 2003.

[3] G.Csurka, C.Bray, C.Dance, and L.Fan, "Visual categorization with bags of keypoints," in *ECCV*, 2004.

[4] J.Sivic and A.Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[5] G.Salton and M.McGill, *Introduction to modern information retrieval*, McGrawHill, 1983.

[6] F.jurie and B.Triggs, "Creating efficient codebooks for visual recognition," in *ICCV*, 2005.

[7] F.Perronnin, C.Dance, G.Csurka, and M.Bressan, "Adapted vocabularies for generic visual categorization," *IEEE Trans. PAMI*, 2008.

[8] J.Farquhar, S.Szedmak, H.Meng, and J.Shawe-Taylor, "Improving bag-of-keypoints image categorisation," in *Tech. report Univ. of Southampton*, 2005.

[9] F.Moosmann, B.Triggs, and F.Jurie, "Randomized clustering forests for building fast and discriminative visual vocabularies," in *NIPS*, 2007.

[10] J.Winn, A.Criminisi, and T.Minka, "Object categorization by learned universal visual dictionary," in *ICCV*, 2005.

[11] J.C.van Germert, C.J.Veenman, A.W.M. Smeulders, and J.M.Geusebroek, "Visual word ambiguity," *IEEE Trans. PAMI*, 2009.

[12] T.Li and IS.Kweon, "Measuring conceptual relation of visual words for visual categorization," in *ICIP*, 2009.

[13] S.Maji, A.C.Berg, and J.Malik, "Classification using intersection kernel support vector machines is efficient," in *CVPR*, 2008.

[14] J.Wu and Rehg.J.M, "Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *ICCV*, 2009.

[15] F.Odone, A.Barla, and A.Verri, "Building kernels from binary strings for image matching.," *IEEE Trans. Image Processing*, vol. 14(2), pp. 169–180, 2005.

[16] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[17] Jianxin Wu, "A fast dual method for hik svm learning," in *European Conference on Computer Vision (ECCV)*, 2010.

[18] B.Leibe and B.Schiele, "Analyzing appearance and contour based methods for object categorization," in *CVPR*, 2003.

[19] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *CVPR Workshop on Generative-Model Based Vision*, 2004.

[20] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.