4-2014

A hamming embedding kernel with informative bag-of-visual words for video semantic indexing

Feng WANG

Wen-Lei ZHAO

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Bernard MERIALDO

A Hamming Embedding Kernel with Informative Bag-of-Visual-Words for Video Semantic Indexing

FENG WANG, East China Normal University
WAN-LEI ZHAO, INRIA Rennes
CHONG-WAH NGO, City University of Hong Kong
BERNARD MERIALDO, Institute Eurecom

In this paper, we propose a novel Hamming Embedding kernel with Informative Bag-of-Visual-Words to address two main problems existing in traditional BoW approaches for video semantic indexing. First, Hamming Embedding is employed to alleviate the information loss caused by SIFT quantization. The Hamming distances between keypoints in the same cell are calculated and integrated into SVM kernel to better discriminate different image samples. Second, to highlight the concept-specific visual information, we propose to weight the visual words according to their informativeness for detecting specific concepts. We show that our proposed kernels can significantly improve the performance of concept detection.

Categories and Subject Descriptors: I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*; I.5.4 [**Pattern Recognition**]: Applications—*Computer vision*

General Terms: Algorithms, Experimentation, Performance.

Additional Key Words and Phrases: Bag-of-Visual-Word, Hamming embedding, kernel optimization, video semantic indexing.

ACM Reference Format:

Wang, F., Zhao, W., Ngo, C. W., and Merialdo, B. 2013 A Hamming Embedding Kernel with Informative Bag-of-Visual-Words for Video Semantic Indexing. ACM Trans. Multimedia Comput. Commun. Appl. 1, 1, Article 1 (January 2013), 20 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

In the past decade, image/video semantic indexing (SIN) has attracted a lot of research attentions especially due to the great efforts of TRECVID Workshop [TRECVID 2012]. SIN task is to automatically detect various concepts (e.g. people, objects, scenes, and events) so as to understand the semantic content of the given images/videos. This serves as a basic step for many applications such as content based image/video retrieval, multimedia event detection and recounting. Despite of the encouraging

The work described in this paper was partially supported by a grant from City University of Hong Kong (Project No. 7008178), the National Natural Science Foundation of China (No. 61103127 and 61375016), Shanghai Pujiang Program (No. 12PJ1402700), and the Fundamental Research Funds for the Central Universities.
Author's address: Feng Wang, Dept. of Computer Science and Technology, East China Normal University, 500 Dongchuan Rd., Shanghai, China 200241; email: fwang@cs.ecnu.edu.cn; Wan-Lei Zhao, INRIA Rennes, France; email: stonescx@gmail.com; Chong-Wah Ngo, Dept. of Computer Science, City University of Hong Kong; email: cwngo@cs.cityu.edu.hk; Bernard Merialdo, Dept. of Multimedia Communications, Institute Eurecom, France; email: Bernard.Merialdo@eurecom.fr.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

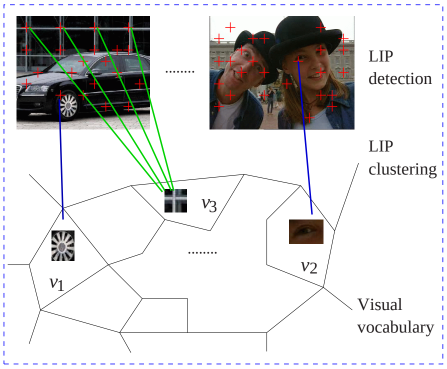

Fig. 1. Construction of visual vocabulary.

advances achieved recently, SIN remains a difficult problem due to the well-known semantic gap between low-level visual information and high-level semantic concepts. Typically SIN is treated as a *one* vs. *all* binary classification problem. Two classes are defined: positive examples which contain the given concept and negative ones in which the concept is not present. Different approaches and features have been proposed. For classifiers, SVM (Support Vector Machine) has been widely adopted. Various features including color, texture, audio and local features are used. Among these features, Bag-of-Visual-Words (BoW) [Jiang et al. 2007] has achieved great success due to its efficiency and effectiveness by capturing discriminative local image information. A lot of efforts have been devoted to improve the performance of BoW features [Nowak et al. 2006; Quelhas et al. 2007; Sudderth et al. 2008; Tuytelaars and Schmid 2007; Winn et al. 2005; Batra et al. 2008; Jurie and Triggs 2005; Fulkerson et al. 2008; Yang et al. 2008; Alhwarin et al. 2008].

For the generation of BoW feature, a visual vocabulary is first constructed on a set of training images as illustrated in Figure 1. In each image, the local interest points (LIPs) are detected [Mikoljczyk and Schmid 2004], and described with SIFT (Scale Invariant Feature Transformation) [Lowe 2004]. The LIPs are then clustered into different groups to form a visual vocabulary. This process actually segments the SIFT descriptor space into different Voronoi cells, each corresponding to a visual word (see Figure 1). To compute the BoW feature vector of a given image, each detected LIP is assigned to one or few nearest visual word(s) [Jiang et al. 2007; Jiang and Ngo 2008]. This results in a histogram on the visual vocabulary, which can be used as the input for classifier training and testing. Although BoW feature can effectly and efficiently capture the local visual information in an image, in the following, we discuss two main problems with this approach.

1.1 Information Loss Problem

During the construction of the visual vocabulary, SIFT descriptor space is quantized into Voronoi cells corresponding to different visual words. Such kind of quantization simply treats all points falling in one cell as identical which results in an inaccurate distance measure between image samples. Figure 2(a) illustrates an example. There are 4 points from each of the three images (I_1, I_2, and I_3) assigned to the same visual word (or lie in the same Voronoi cell c_1). With BoW approach, the distances among three images are all zeros since the BoW histogram simply counts the number of keypoints in each cell. However, inside the given cell, the distance between I_1 and I_3 is obviously larger than that between I_1 and I_2. This difference is not considered in BoW model which assumes all points in the same cell to be identical. This is because the representing feature of a keypoint has been lost at quantization stage. Typically, BoW feature is generated by segmenting the 128-dimension SIFT descriptor space into only 200 to 20000 cells, and thus the size of each cell is very large. The points assigned to the

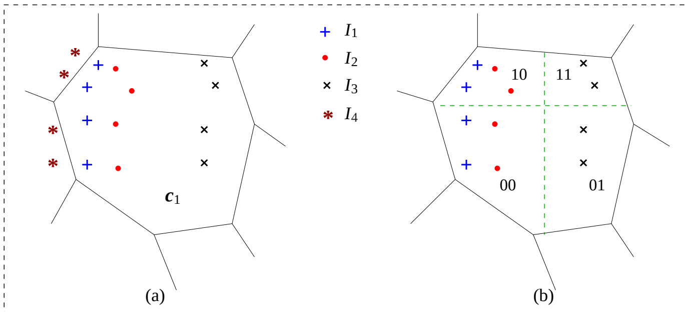

Fig. 2. (a) Information loss problem with BoW feature: there are 4 points from each of the three images (I_1, I_2, and I_3) mapped to the same visual word. The distance between I_1 and I_3 is obviously larger than that between I_1 and I_2. This difference is ignored by BoW approach. The information loss differs from the mismatch problem where similar points (e.g. points from I_4 and I_1) are mapped to different visual words (see Section 2.1). (b) In our approach, Hamming embedding is employed to attach a binary signature to each descriptor to encdoe its location inside the cell, which is then used to measure the intra-cell distance.

same visual word could be much different from each other. Because of ignoring this difference, BoW approach largely loses discriminative power of SIFT descriptors, and suffers from inaccurate distance measure between different image samples.

1.2 Concept-Specific Informativeness of Visual Words

BoW feature captures the visual distribution of an image on the whole SIFT descriptor space. The typical size of the visual vocabulary is 200 to 20000, and the number of keypoints in each image ranges from tens to hundreds. The visual information of a concept actually cannot be evenly distributed over the whole vocabulary. Given a concept, only some of the visual words frequently appear while the presence of the others are nearly random. In other words, some visual words are more informative or important for the detection of a specific concept, while the others may be noisy. For instance, in Figure 1, the visual word v_1 containing a wheel-like image patch is quite important for detecting the concept *Car* or *Vehicle*, but may not contribute to the detection of concept *Person*. Similarly, the presence of v_2 containing an eye-like image patch implies there is a *Person* or *Human_face* with high probability. However, the relation between the concept *Car* and visual word v_2 could be weak.

In the traditional BoW representations, for different concepts, each visual word is treated equally. The discriminative ability of those informative visual words would be seriously reduced considering two factors: i) The background of the images from open video domains is extremely complex, which contains lots of visual information besides the concept-related objects and scenes. Thus, the informative visual words can be easily noised. ii) In most current datasets, the number of positive examples are usually limited. For instance, in the datasets used for TRECVID evaluation, there are only tens of positive examples for many concepts. Therefore, denoising is hard without enough training data available. This inspires us to select the visual words which are more informative to build more discriminative classifiers for detecting specific concepts.

In this paper, we propose a novel Hamming Embedding Kernel with Informative Bag-of-Visual-Words to address the above mentioned problems in BoW based approaches and boost the performance of video semantic indexing. The remaining of this paper is organized as follows. Section 2 briefly reviews the related works. In Section 3, we propose to integrate Hamming embedding distance [Jegou et al. 2008] into SVM kernel to address the information loss problem. In Section 4, we present our approach to weight the concept-specific informativeness of different visual words to improve the discriminative ability of classifiers. Experiments are conducted on several challenging datasets and presented in Section 5. Finally, Section 6 concludes this paper.

2. RELATED WORKS

2.1 Information Loss Problem of Visual Words

In [Gemert et al. 2010; Jiang et al. 2007; Jiang and Ngo 2008], soft weighting is proposed to alleviate the mismatch problem which is also caused by SIFT quantization. For instance, in Figure 2(a), although two sets of keypoints from I_1 and I_4 are visually similar, they are assigned to two different visual words. In [Jiang et al. 2007; Jiang and Ngo 2008], the mismatch problem is alleviated by expanding each keypoint from one word to few nearest neighboring words. This actually enlarges the Voronoi cells and allows overlap between neighboring cells so that similar keypoints could be assigned to the same visual words. However, soft weighting does not consider the location information [1] of the keypoint in the Voronoi cell. All keypoints in the same cell are still assumed identical and expanded to the same visual words. In this paper, we address the information loss problem. In other words, soft weighting considers the similarity between keypoints in different cells, while we address the dissimilarity between keypoints in the same cell. Actually, soft weighting would even worsen the intra-cell dissimilarity by increasing the size of each cell. For instance, in Figure 2(a), when the points from I_4 (and other points in neighboring cells) are expanded to the cell c_1, all points from 4 images would be considered identical despite of the large dissimilarities among them.

The above mentioned information loss problem due to SIFT quantization can be alleviated by using a larger visual vocabulary with smaller Voronoi cells. However, a larger vocabulary generally causes considerably more mismatches. Furthermore, each image just has tens to hundreds of keypoints detected. A large vocabulary will result in very sparse feature vectors. As a result, the detection model cannot be fully trained to capture the variations among image samples of the same concept. The situation becomes even worse when there are not enough positive examples available for training (which is the case in most current datasets). This will eventually tune down the performance of concept detection.

In this paper, we employ Hamming embedding to alleviate the information loss problem of BoW features in video semantic indexing. Low-dimension embedding is a critical problem in many applications to map high-dimension data to low-dimension space [Saul and Roweis 2003]. A lot of approaches have been proposed such as principal component analysis (PCA), multi-dimensional scaling (MDS) and locally linear embedding (LLE) [Roweis and Saul 2000]. In multimedia information retrieval, the low-dimension embedding is also addressed. Yang et al. [2012] propose a semi-supervised approach to learn a better data representation with relevance feedback information. In [Cai et al. 2011], linear discriminant projection (LDP) is employed for reducing the dimensionality and improving the discriminability of local image descriptors. In [Jegou et al. 2008], Hamming embedding is employed in BoW based image search. To achieve a tradeoff between the large memory/time cost by using high-dimension descriptors and low precision by using BoW representation, SIFT descriptors are mapped to Hamming space. For each keypoint, a binary signature is generated encoding its location information inside the cell. The similarity between keypoints mapped to the same visual word is further estimated by the Hamming distance between their binary signatures to filter out the dissimilar keypoints. In [Jain et al. 2012], Hamming embedding is used to refine the patch matching for image classification. In [Sibiryakov 2009], an approach similar to [Jegou et al. 2008] is proposed while Hamming embedding is learned by maximizing the entropy of the resulting binary codes. In [Jain et al. 2011], an asymmetric Hamming embedding scheme is proposed for large scale image search. Wang, J. et al. [2012] propose a semi-supervised approach to maximize the accuracy of the binary partition. In [Shakhnarovich 2005], different weights are learned and assigned to different bits of the binary signature to boost the perfor-

[1] In this paper, the terminology “location information” refers to the location of a keypoint in its corresponding Voronoi cell in SIFT descriptor space, while the coordinate of a keypoint in the image is not considered here.

mance of Hamming embedding. In [Wang et al. 2006], larger weights are assigned to the higher bits to highlight their importance for distance measure in image search. In [Jiang et al. 2013], a query-adaptive approach is proposed to derive finer-grained ranking in image search by addressing the discrete values of Hamming distance. In this paper, we employ Hamming embedding in video semantic indexing to derive a more precise measure between keypoints mapped to the same visual word, which is then integrated into SVM kernel for discriminative classification. The novelty of our approach lies in the proposed Hamming embedding kernel which enhances the distinctiveness of BoW model. By considering the location information of keypoints in each cell, our proposed kernel can more precisely measure the distance between different image samples and improve the SVM performance.

2.2 Weighting Concept-Specific Informativeness of Visual Words

As discussed in Section 1.2, the traditional BoW representation ignores the informativeness of different visual words for specific concepts. Some attempts have been made to address the concept-specific information in image categorization. One way is to build concept-specific vocabularies. In [Perronnin 2008], besides the universal vocabulary which describes the visual content of all classes of images, class vocabularies are constructed by adapting the universal vocabulary using class-specific data. An image is then characterized by a set of histograms - one per class - where each histogram describes whether the image content is best modeled by the universal vocabulary or the corresponding class vocabulary. In [Yang et al. 2007; Moosman et al. 2008], to capture the desired information of a given object category, discriminative visual vocabulary is generated. In this kind of approaches, given an image, a set of BoW histograms are generated on all different concept-specific vocabularies. This would be quite time-consuming in video semantic indexing when there are thousands of concepts for annotation.

In contrast to constructing discriminative vocabularies, another way is to select informative visual words from a single universal vocabulary. Some previous works propose to detect and remove the non-informative visual words. Sivic and Zisserman [2003] consider the visual words that frequently appear in almost all images as useless. Tirilly et al. [2008] propose to eliminate useless visual words based on the geometric properties of the keypoints and the probabilistic latent semantic analysis. Yuan et al. [2007] propose to locate useless information by significance test. Kesorn and Poslad [2012] try to discover non-informative visual words with document frequency and statistical association between visual words and concepts. Most of these works borrow ideas from feature selection methods in text categorization and statistic information is widely employed to detect useless visual words. However, it is difficult to associate low-level visual words with high-level semantic concepts based on statistic information in open video domains. Given an image/video, there are usually abundant visual information in the background besides the target concepts. For instance, DFs (Document Frequencies) are popularly used in feature selection for text categorization. However, in video concept detection, DFs of visual words do not convey much useful information. This is because each image contains a lot of noisy information in the background, and each visual word can be found in the positive samples of almost every concept, meaning that DFs for different words are almost the same. Furthermore, in most datasets, there are usually limited positive examples for many concepts, and thus the statistic information computed from the training sets are usually not reliable.

Besides the difficulty in computing statistical associations between visual words and semantic concepts, there are still two main problems to address in this kind of approaches. First, although some of the non-informative visual words are detected and removed, the informativeness of remaining visual words for specific concepts are not considered. Second, the ultimate aim is to improve the discriminative ability of classifiers. However, the feature selection method based on statistic information is disconnected from the classifier learning process. The selection of visual words is based on some statistical rules, instead of their contributions to improve the classifiers' performances. In this paper, we propose an Informave Bag-of-Visual-Words (IBoW) representation by considering the informativeness of each visual word for a given concept. A universal visual vocabulary is first constructed. To capture the concept-specific visual information, different weights are assigned to visual words according to their contributions to SVM classification. Finally, the resulting weights are used in the SVM kernel for concept detection.

3. HAMMING EMBEDDING KERNEL

In our approach, a visual vocabulary is first constructed on a set of training images. In each image, keypoints are detected using Difference of Gaussian (DoG) [Lowe 2004] and Hessian-Laplacian detectors [Mikoljczyk and Schmid 2004], and described with SIFT [Lowe 2004]. K-means algorithm is employed to cluster all keypoints and construct a visual vocabulary $V = \{v_i | i = 1, 2, \cdots, S\}$. Given an image, each detected keypoint is assigned to the nearest visual word in V. This results in an S-bin histogram to describe the visual information in the given image, which can be used as the input for SVM classifiers. Let $X_p = [x_{p1}, x_{p2}, \cdots, x_{pS}]$ and $X_q = [x_{q1}, x_{q2}, \cdots, x_{qS}]$ be the BoW feature vectors of two images I_p and I_q respectively. For SVM classification, the following $\chi^2 - RBF$ kernel is the most frequently used

$$K(I_p, I_q) = \exp(-\sigma \cdot \sum_{c=1}^{S} \frac{|x_{pc} - x_{qc}|^2}{x_{pc} + x_{qc}}) \tag{1}$$

As discussed in Section 1, due to the information loss caused by quantization, this traditional BoW approach simply assumes that all the keypoints assigned to the same visual word to be identical. As a result, intra-cell distances among keypoints within the same cell are not considered. This reduces the precision of distance measure considering the large scale of the Voronoi cells. In this section, we propose a novel Hamming embedding kernel by considering the intra-cell distance for concept detection. First we briefly recall the algorithm in [Jegou et al. 2008] to associate each descriptor with a binary signature encoding its location information inside the corresponding Voronoi cell. The Hamming distance between the signatures of different descriptors are then calculated to estimate their Euclidean distance. Finally, we integrate Hamming distance into SVM kernel for more precise distance measure between image samples so as to improve the performance of classification.

3.1 Training Hamming Embedding

The most precise intra-cell distance between descriptors could be computed directly with the SIFT descriptors. However, this information is lost during quantization. Furthermore, it would be much space-consuming to keep all 128-dimensional descriptors and time-consuming to compute the distance between them. To cope with this problem, we associate a binary signature with each descriptor to encode its location information inside the Voronoi cell (as illustrated in Figure 2(b)) by employing Hamming embedding, which was previously proposed for image search and copy detection [Jegou et al. 2008]. Hamming embedding follows the scheme of Locality-Sensitive Hashing (LSH) [Indyk and Motwani 1998; Charikar 2002]. The main idea of LSH is that highly similar objects are indexed by the hash table with high probability. The similarity between two objects x, y can thus be approximated by the probability of collision of two inputs in the hash table

$$sim(x, y) = Pr_{h \in \mathcal{F}}[h(x) = h(y)] \tag{2}$$

where $\mathcal{F}$ is a family of hash functions. This leads to compact representations of high-dimension objects and efficient algorithms for similarity estimation between them. In Hamming embedding approach, each SIFT descriptor is first projected onto a set of randomly generated orthogonal vectors. Each projected component is then binarized by using a threshold learned from a training dataset. The threshold for each component actually partitions the Voronoi cell into two (as illustrated in Figure 2(b) where two components segment the cell into four). Finally a bit-sequence signature is generated for each descriptor, which provides a more precise representation for the location information of the descriptor in the corresponding Voronoi cell. The Hamming distance between two signatures can then be used to estimate the distance between two high-dimensional descriptors.

To generate the binary signature, offline training of Hamming embedding is needed to generate the projection matrix and the thresholds for binarization. The algorithm for offline training can be summarized as follows [Jegou et al. 2008]:

—**Step 1. Random matrix generation**: Generate an orthogonal projection matrix $\mathbf{P}$ ($l_b \times S$), where l_b is the length of the binary signature, and S is the the vocabulary size. Draw an $S \times S$ matrix of Gaussian values and apply a QR factorization to it. The first l_b rows of the orthogonal matrix obtained by the decomposition form the matrix $\mathbf{P}$. In our implementation, l_b is empirically set to 32, which is enough to precisely describe the location information of the descriptors..

—**Step 2. Feature vector projection and assignment**: A large set of feature vectors from an independent dataset are randomly sampled. These feature vectors are first assigned to words in the visual vocabulary. Additionally, given a descriptor r assigned to a visual word v_i, it is further projected by matrix $\mathbf{P}$. This operation results in a component t^r with l_b dimensions.

—**Step 3. Median values of projected descriptors**: For each visual word v_i, compute the median values for l_b dimensions based on all projected components t^r which fall into word v_i. This results in an $S \times l_b$ matrix $\mathbf{M}$. Each row of $\mathbf{M}$ corresponds to one visual word in the vocabulary.

3.2 Binary Signature Generation

With Hamming embedding, we generate a binary signature for each descriptor to encode its location information inside the Voronoi cell. This is carried out by first projecting the descriptor onto the above generated matrix P and then binarizing each component with the learned median value. The algorithm proceeds as follows. Given a 128-dimention SIFT descriptor r,

—**Step 1. Quantize r to the nearest visual word v_i.**

—**Step 2. Project r using matrix P (the same one used at the training stage)**. This results in a vector t^r with l_b dimensions.

—**Step 3. Compute the binary signature.** By comparing the resulting vector t^r with the i-th row of matrix $\mathbf{M}$ which corresponds to visual word v_i, a binary signature $b(r) = [b_1(r), ..., b_{l_b}(r)]$ for r is generated by

$$b_k(r) = \begin{cases} 1 & \text{if } t_k^r \geq M_{i,k} \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

where t_k^r is the k-th element of t^r, and the binary signature $b(r)$ records the approximate location of r in the Voronoi cell.

Given two keypoints with descriptors r and r' which are assigned to the same visual word, the Hamming distance between them can be calculated based on the associated binary signatures as

$$H(r, r') = \sum_{1 \leq j \leq l_b} 1 - \delta_{b_j(r) b_j(r')} \tag{4}$$

According to Equation 2, the Hamming embedding and binary signature are designed so that the Hamming distance calculated by Equation 4 reflects the Euclidean distance between two descriptors in the same cell [Jegou et al. 2008]. This provides a way of recording the location information of descriptors

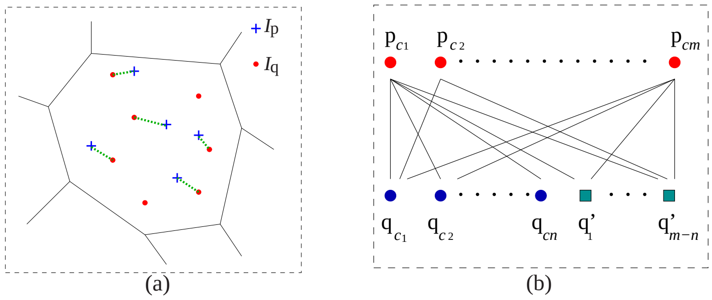

Fig. 3. (a) Intra-cell distance by keypoint matching: Two sets of points from two images I_p and I_q respectively are assigned to the same cell. An optimal matching between two point sets are found and the dashed lines link each point to its matched point, which can be used to calculate the distance between two point sets. (b) Bipartite point matching between P_c and Q_c: The distance (cost) between each pair is defined in Equation 5.

and estimating the distance between them in the same cell with an elegant representation (by just appending l_b bits to each descriptor together with the index of the word in BoW vector).

3.3 Hamming Embedding Kernel

In this section, with Hamming embedding described above, we propose a new distance measure between image samples with higher precision by considering the intra-cell distance between descriptors, and incorporate it into SVM kernel to improve the performance of concept detection.

First we consider only one bin (or visual word) of the BoW histogram. Let $P_c = \{p_{c1}, p_{c2}, \cdots, p_{cm}\}$ and $Q_c = \{q_{c1}, q_{c2}, \cdots, q_{cn}\}$ be two keypoint sets from images I_p and I_q respectively, and assigned to the same word v_c. In traditional BoW approach (see Equation 1), by assuming any two keypoints in the same cell to be identical, the difference of the two histogram bins $d_c(I_p, I_q) = |m - n|$ are used to calculate the distance between them.

With Hamming embedding to encode each keypoint's location information inside the Voronoi cell, the Euclidean distance between any two keypoints in P_c and Q_c can be estimated by Equation 4 with higher precision. This enables us to develop a distance measure between P_c and Q_c by considering the distance between keypoints in the same cell. To this end, we need to find the matching between P_c and Q_c. For image search and ND (Near-Duplicate) detection, geometry constraints [Jegou et al. 2008] can be used to refine the matching points between two images and detect the duplications. However, for concept detection, geometry alignment does not exist since the images (or image patches) are usually not duplicate to each other even within the same concept. To measure the distance between two keypoint sets P_c and Q_c, we perform one-to-one matching between them and then use the distance between matching points as the distance between two images on the visual word v_c.

As illustrated in Figure 3(a), the distance between two point sets in the same cell can be calculated by summing up the distance between each point and its matched point from another image. The problem is then to find the optimal matching between P_c and Q_c. We treat this as a bipartite point matching problem. In Figure 3(b), $m - n$ virtual points are added to Q_c. The cost of each edge $e \in P_c \times Q_c$ is defined by

$$\begin{cases} d(p_{ci}, q_{cj}) = \frac{H(p_{ci}, q_{cj})}{l_b} \\ d(p_{ci}, q'_k) = 1 \end{cases} \tag{5}$$

where $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n$, $k = 1, 2, \cdots, m - n$, and $H(\cdot)$ is the Hamming distance between two keypoints (descriptors) calculated by Equation 4.

This is a typical assignment problem to match the points between P_c and Q_c with the minimum cost (distance). In our approach, we adopt Hungarian algorithm [Kuhn 1955] to find the globally optimal matching between P_c and Q_c. The Hungarian method is a classical combinatorial optimization algorithm which solves the assignment problem in polynomial time. With this algorithm, we find the globally optimal matching between P_c and Q_c. Thus, in the cell corresponding to word v_c, each keypoint is linked to its best match from another image and their Hamming distance is used to measure the difference between them. The distance between P_c and Q_c is then calculated as

$$\hat{d}_c(I_p, I_q) = \sum_{i=1}^{m} d(p_{ci}, \phi(p_{ci})) \tag{6}$$

where $\phi(\cdot)$ is the optimal matching from P_c to Q_c. According to the definition in Equation 5, each pair of matched points contribute to $\hat{d}_c(I_p, I_q)$ by a value lying in $[0, 1]$ depending on the Hamming distance between them. On the other hand, each point that is not matched (or matched to a virtual point) in P_c will directly contribute a value *1* to the distance between two image samples.

By considering the location information of each descriptor in the Voronoi cell, Equation 6 provides a more precise distance measure, and thus can better discriminate different image samples. Based on Equation 6, we then propose a new kernel, namely **Hamming Embedding Kernel** by modifying Equation 1 as

$$\hat{K}(I_p, I_q) = \exp(-\sigma \cdot \sum_{c=1}^{S} \frac{\hat{d}_c^2(I_p, I_q)}{x_{pc} + x_{qc}}) \tag{7}$$

By comparing the distance measures used in Equations 1 and 7, the traditional BoW approach is actually a simplified version of Equation 7 by setting $d(p_{ci}, q_{cj}) = 0$ for all point pairs p_{ci} and q_{cj} ($1 \leq i \leq m$, $1 \leq j \leq n$) in the same cell v_c. This implies that all the keypoints assigned to the same visual word are identical. Since the large variations between them are ignored, the resulting distance measure suffers from precision loss. In our approach, Hamming distance is used to measure the intra-cell difference between keypoints. The new kernel in Equation 7 employs a more precise measure between different image samples. As indicated by comprehensive experiments, it is more discriminative and demonstrates consistent improvement over traditional models.

4. INFORMATIVE BAG-OF-VISUAL-WORDS

As discussed in Section 1, different visual words are not equivalently important for detecting a specific concept. In Equations 1 and 7, for every concept, all visual words are treated as the same. In this section, we further revise the SVM kernel by measuring the informativeness of visual words for specific concept and then assigning different weights to them accordingly to capture concept-specific visual information. Here we just consider the detection of one given concept and denote the weight vector of visual words as $w = [w_1, w_2, \cdots, w_S]$, where $\sum_{c=1}^{S} w_c = 1$. By attaching the concept-specific informativeness of visual words to traditional BoW, we propose the Informative Bag-of-Visual-Words (IBoW) representation and employ it in SVM classification by rewriting Equation 7 as

$$\tilde{K}(I_p, I_q) = \exp(-\sigma \cdot \sum_{c=1}^{S} w_c \cdot \frac{\hat{d}_c^2(I_p, I_q)}{x_{pc} + x_{qc}}) \tag{8}$$

where the weight w_c measures the informativeness of the word v_c for detecting the given concept. As a result, the more important visual words contribute more to the distance measure between image samples. In our approach, we calculate a set of weights for each specific concept. This is treated as a kernel optimization problem. We first employ Kernel Alignment Score (KAS) to evaluate the discriminative ability of SVM kernels. The weights of visual words are then iteratively updated so as to produce an optimal kernel for classification.

4.1 Evaluating SVM Kernels

To measure the fitness of the weights of visual words, the classification accuracy of SVM is the best hint. However, it is not applicable to evaluate the performance of SVM by training, cross-validation and testing from time to time during the weighting procedure. The performance of SVM is mainly dependent on the ability of kernel matrix to discriminate between positive and negative samples. Different factors can affect kernel matrix such as kernel function format, features, and parameter settings. Kernel optimization is to find a better kernel by optimizing these factors. In this paper, we weight the visual words in the framework of kernel optimization, i.e. we attempt to find the optimal weights that can produce the best kernels.

For SVM, an optimal kernel K^{opt} [Cristianini et al. 2002] should satisfy

$$K_{pq}^{opt} = \begin{cases} +1 & \text{if } l_p = l_q \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where for simplicity, $K_{pq}^{opt} = K^{opt}(I_p, I_q)$ is the kernel value between two image samples I_p and I_q, $l_p = l(I_p)$ is the label of I_p, and $l_p = +1$ (or -1) if I_p is a positive (or negative) sample. In K^{opt}, the kernel values between samples with the same labels are maximized, while the values between samples with different labels are minimized. Thus, this optimal kernel can perfectly discriminate between different classes.

However, the actual kernels used in practice are usually not optimal due to the imperfect features and kernel functions. To measure how well a given kernel $\tilde{K}$ is aligned with the optimal kernel defined by Equation 9, the following Kernel Alignment Score (KAS) [Cristianini et al. 2002] is used

$$\mathcal{T} = \frac{\sum_{p,q} \tilde{K}_{pq} \cdot l_p \cdot l_q}{N \cdot \sqrt{\sum_{p,q} \tilde{K}_{pq}^2}} \tag{10}$$

where N is the total number of samples and the denominator is a normalization factor. If $l_p = l_q$ (p and q belong to the same class and $l_p \cdot l_q = 1$), $\tilde{K}_{pq}$ is expected to be maximized in the optimal kernel. Thus, in Equation 10, a larger (or smaller) $\tilde{K}_{pq}$ value will increase (or reduce) the KAS score $\mathcal{T}$ as expected. Similarly, if $l_p \neq l_q$, i.e. $l_p \cdot l_q = -1$, $\tilde{K}_{pq}$ should be minimized and a larger $\tilde{K}_{pq}$ value will be penalized in the calculation of $\mathcal{T}$. Generally, a kernel with higher KAS score is better at discriminating samples of different classes, and can potentially achieve better performance for classification. In our approach, we employ KAS to measure the fitness of SVM kernel and weight the informativeness of visual words by maximizing KAS scores.

Equation 10 assumes the two classes are balanced. However, this is not the case for most current datasets in video semantic indexing, where there are usually many more negative examples than positive ones. This may bias the resulting KAS towards the negative class. To deal with this imbalance problem of the datasets, we modify Equation 10 by assigning different weights to positive and negative examples as follows

$$\alpha_p = \begin{cases} 1 & \text{if } l_p = -1 \\ \frac{N^-}{N^+} & \text{otherwise} \end{cases} \tag{11}$$

where N^- and N^+ are the numbers of negative and positive examples in the training dataset respectively. Equation 10 is then modified as

$$\mathcal{T} = \frac{\sum_{p<q} \tilde{K}_{pq} \cdot l_p \cdot l_q \cdot \alpha_p \cdot \alpha_q}{N' \cdot \sqrt{\sum_{p<q} \alpha_p \cdot \alpha_q \cdot \tilde{K}_{pq}^2}} \tag{12}$$

where $N' = \sum_{p<q} \alpha_p \cdot \alpha_q$. Eventually the weighting problem is formulated as searching for an optimal weight vector w^{opt} such that the KAS score $\mathcal{T}$ defined by Equation 12 is maximized.

4.2 Gradient-based Weight Optimization

Gradient-descent approach is widely used for optimization. In [Igel et al. 2007], gradient-based algorithm is employed to select SVM parameters for bacterial gene start detection in biometrics. In our approach, we weight the informativeness of visual words by adopting a similar gradient-descent algorithm to optimize the SVM kernels by maximizing the KAS score in Equation 12. Based on Equation 12, we calculate the partial derivative of $\mathcal{T}$ to the weight w_c as

$$\frac{\partial \mathcal{T}}{\partial w_c} = \sum_{p<q} \frac{\partial \mathcal{T}}{\partial \tilde{K}_{pq}} \cdot \frac{\partial \tilde{K}_{pq}}{\partial w_c} \tag{13}$$

$$\frac{\partial \tilde{K}_{pq}}{\partial w_c} = \tilde{K}_{pq} \cdot (-\sigma \cdot \frac{\hat{d}_c^2(I_p, I_q)}{x_{pc} + x_{qc}}) \tag{14}$$

Besides the weights of visual words, we also optimize σ in Equation 8 which is an important parameter for SVM kernels

$$\frac{\partial \mathcal{T}}{\partial \sigma} = \sum_{p<q} \frac{\partial S}{\partial \tilde{K}_{pq}} \cdot \frac{\partial \tilde{K}_{pq}}{\partial \sigma} \tag{15}$$

$$\frac{\partial \tilde{K}_{pq}}{\partial \sigma} = \tilde{K}_{pq} \cdot (-\sum_{c=1}^{S} w_c \cdot \frac{\hat{d}_c^2(I_p, I_q)}{x_{pc} + x_{qc}}) \tag{16}$$

In Equation 14, the weights of different visual words are assumed to be independent on each other. According to our experiment, this assumption is reasonable. Strictly speaking, there might be some weak correlations between the importance of different visual words. For instance, two visually similar words may have the similar weights. Based on Equations $13-16$, we iteratively update the weight vector w of visual words so as to maximize the kernel alignment score defined by Equation 12. Below is the algorithm for optimization:

(1) *Initialize* $w_c = \frac{1}{S}$ *for* $c = 1, 2, \cdots, S$, *and* $\sigma = \sigma_0$. *Calculate the initial KAS score* $\mathcal{T}$ *by Equation 12.*

(2) *For each weight* w_c *and* σ, *calculate the partial derivative* $\frac{\partial \mathcal{T}}{\partial w_c}$ *and* $\frac{\partial \mathcal{T}}{\partial \sigma}$ *by Equations 13-16.*

(3) *Update weights* $w'_c = w_c \cdot (1 + sign(\frac{\partial \mathcal{T}}{\partial w_c}) \cdot \delta_w)$ *and* $\sigma' = \sigma \cdot (1 + sign(\frac{\partial \mathcal{T}}{\partial \sigma}) \cdot \delta_\sigma)$, *where* $sign(t) = \begin{cases} +1 & if \;\; t > 0 \\ 0 & if \;\; t = 0 \\ -1 & if \;\; t < 0 \end{cases}$. δ_w *and* δ_σ *are two constants to be determined. Get the new weights* $w_c = \frac{w'_c}{\sum_{k=1}^{S} w'_k}$.

(4) *Calculate the new kernel alignment score* $\mathcal{T}'$ *using the updated weights and* σ. *If* $\frac{\mathcal{T}' - \mathcal{T}}{\mathcal{T}} < thres$, *stop; otherwise,* $\mathcal{T} = \mathcal{T}'$ *and go to step 2.*

In step 1, $\sigma_0 = N' / \sum_{i<j}(-\sum_{c=1}^{S} w_c \cdot \frac{\hat{d}_c^2(I_p,I_q)}{x_{pc}+x_{qc}})$ which is the inverse of the average distance between all training samples, and a good empirical choice of σ for SVM paramter selection. In step 3, the weight vector (and σ) is updated by a small value δ_w (and δ_σ). For the determination of δ_w (and δ_σ), a larger value can push the weights (and σ) to the optimal one at higher speed at the beginning. However, this risks missing the optimal weight vector (and δ_σ) by skipping a large distance in each step. A smaller δ_w (and δ_σ) can avoid this problem, but it takes more iterations to converge. In our experiments, we empirically set $\delta_w = 0.02$ and $\delta_\sigma = 0.1$. In step 4, $thres$ is set to be 0.5% so as to stop the optimization when the improvement on $\mathcal{T}$ becomes minor. After the optimization process, the resulting weight vector is used to train SVMs with the kernel defined in Equation 8 for concept detection.

5. EXPERIMENTS

In this section, we carry out experiments on several datasets for video semantic indexing with our proposed approaches. The following datasets are used.

—*Sound & Vision dataset:* There are about 100 hours of news magazine, science news, news reports, documentaries, educational programming, and archival videos in MPEG-1. This dataset is separated into development and test sets containing 21532 and 22084 shots respectively.

—*IACC.1 collection*: This dataset is composed of 4 subsets: IACC.1.tv10.training, IACC.1.A, IACC.1.B, and IACC.1.C [TRECVID 2012], each containing approximately 200-hour Internet Archive videos. In total there are about 27200 videos with duration between 10 seconds and 3.5 minutes.

In our experiments, we mainly validate the performance improvement by employing our proposed approach compared with BoW based approaches. A visual vocabulary is first constructed on the development set. One keyframe is used to represent the visual information in each video shot. Difference of Gaussian (DoG) [Lowe 2004] and Hessian Affine [Mikoljczyk and Schmid 2004] detectors are used to detect LIPs (Local Interest Points), and 128-dimension SIFT feature [Lowe 2004] is extracted to describe each local image patch. The visual vocabulary is then generated by clustering all SIFT descriptors from training images into a number of visual words with k-means algorithm. Finally, given a keyframe, each SIFT descriptor of a LIP is mapped to the nearest visual word to form the BoW histogram. For classification, SVM with χ^2-RBF kernel is employed. This traditional BoW approach is used in our experiments for comparison.

5.1 Performance of Informative Bag-of-Visual-Words

In this section, we first experiment the Informative Bag-of-Visual-Words (IBoW) with the weights estimated by our approach proposed in Section 4. By considering the weights of visual words for detecting a specific concept t, χ^2-RBF kernel in Equation 1 is modified as

$$K(I_p, I_q) = \exp(-\sigma \cdot \sum_{c=1}^{S} w_c \cdot \frac{|x_{pc} - x_{qc}|^2}{x_{pc} + x_{qc}}) \tag{17}$$

We compare our approach with existing weighting methods and dimension reduction approaches including uniform weighting, TF-IDF (Term Frequency - Inverse Document Frequency), Chi-square statistic, Linear Discriminant Analysis (LDA), and Locality Preserving Projection (LPP). For uniform weighting scheme as used in traditional BoW approach, all visual words are assigned with the same weights, i.e. $w_c = 1$ for $c = 1, 2, \cdots, S$.

TF-IDF is a classical weighting algorithm widely used in information retrieval. Chi-square statistic is also intensively studied for feature selection in text categorization. In some recent works [Kesorn and Poslad 2012; Alhwarin et al. 2008], chi-square statistic is employed to measure the dependency between visual words and concepts to improve the performance of BoW representation. In [Kesorn and Poslad 2012], the visual words with chi-square values lower than a threshold are considered as useless and removed. LDA and LPP are two methods widely used for dimension reduction. LDA attempts to find the interdependency between features by expressing a dependent feature as a linear combination of other features. LPP can be seen as an alternative to Principal Component Analysis (PCA). It is a linear approximation of the nonlinear Laplacian Eigenmap and capable of discovering the non-linear structure of the data manifold [He and Niyogi 2003].

Our experiment is carried out on the Sound & Vision dataset. A visual vocabulary of 1000 words is constructed on the development set. We adopt Average Precision (AP) to evaluate the performance of concept detection. Two-fold cross-validation is carried out on development and test sets. Figure 4 compares the performances of different approaches. For TF-IDF approach, minor improvement (0.48%) on the MAP (Mean Average Precision) for 20 concepts is observed compared with uniform weighting approach. For different concepts, the performance improvement by TF-IDF is inconsistent. Although TF-IDF has proved useful in information retrieval, it is not appropriate for weighting the visual words in concept detection. First, as discussed in Section 1, DFs for different words are almost the same since each image contains a lot of noisy information in the background, and each visual word can be found in the positive samples of almost every concept. Second, TF is also not a good hint for the importance of the visual word. The importance of a visual word for the detection of a concept is dependent on its informativeness instead of its frequency. For instance, in Figure 1, the presence of word v_1 is very important for detecting the concept *Car*. Although there is only one keypoint assigned to v_1, it is more important than word v_3 with many keypoints in the background assigned to it.

By employing chi-square statistics between visual words and concepts, a slight improvement of 1.87% is achieved compared with the uniform weighting scheme. On one hand, this shows that chi-square statistic is better at discovering the dependency between visual words and concepts compared with TF-IDF. However, as can be seen in Figure 4, the statistic information is reliable and useful only when there are enough training samples for the given concepts such as *Cityscape* and *Hand*. For those concepts with limited positive samples such as *Bus* and *Classroom*, the performance would be even reduced. For LDA and LPP, no significant improvement or reduction on MAP is observed. This shows that both of them are able to discover the interdependency between different features and useful for dimension reduction. On the other hand, the informativeness of features and the interdependency between features and concepts should be further measured in order to improve the discriminative abilities of the classifiers.

Figure 4 also presents the performance of our weighting approach proposed in Section 4. Overvall an improvement of 7.55% is achieved on MAP compared with the uniform weighting scheme. Large margins on the performances of these two approaches can be observed for some concepts including *Bus* (27.4%), *Telephone* (25.8%), *Demonstration* (28.9%), and *Classroom* (21%). These concepts are mostly object-related with specific exterior appearance. By assigning larger weights to the visual words describing their appearances, the detection accuracy of these concepts can be significantly improved. Furthermore, the improvement is consistent for different concepts. Compared with the statistic based approaches, our proposed weighting algorithm aims at optimizing the discriminative ability of SVM kernels. This connects the weighting procedure and the classifier training, and thus the resulting weights can consistently improve the performance of SVM classification.

Figure 5 plots the computed weights of visual words for two concepts: *Boat_Ship* and *Telephone* by our approach (for the convenience of illustration, we use a small vocabulary with 200 visual words). In Figure 5, the visual words are sorted in descending order of their weights for concept *Boat_Ship*, and the weights for *Telephone* (red marks) are then plotted accordingly for comparison. From Figure 5, we can see: i) For a given concept (*e.g. Boat_Ship*), some visual words are assigned with much larger

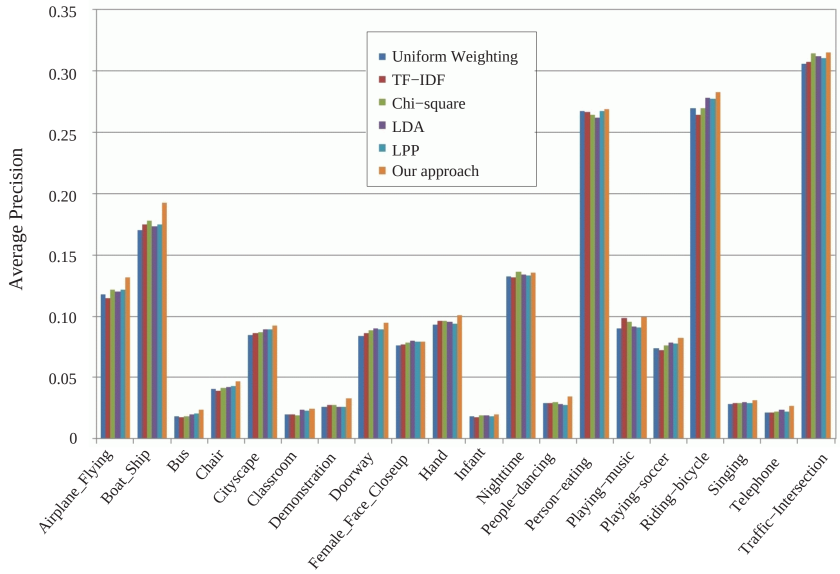

Fig. 4. Comparison between different approaches for weightting informativeness of BoW.

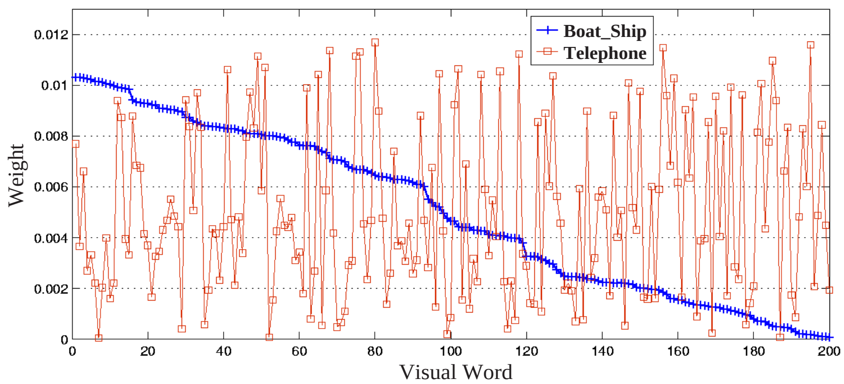

Fig. 5. Computed informativeness of visual words for two concepts.

weights than others. This demonstrates the variations of different visual words' informativeness for detecting a given concept. Therefore, it is important to select the most informative visual words in order to improve the discriminative ability of the classifiers. ii) The weights of the same visual word for different concepts are also quite different. One visual word which is important for a given concept may not be important for another one. Thus, it is necessary to weight the visual words for different concepts instead of using the same weights for deteting all concepts.

5.2 Performance of Hamming Embedding Kernel with IBoW

In this section, we experiment the proposed Hamming Embedding Kernel with Informative Bag-of-Visual-Words. The experiment is carried out on IACC.1 video collections. 50 concepts from the TRECVID 2012 Semantic Indexing (SIN) task (light submission) [TRECVID 2012] are detected. Table I presents the detailed experimental results of five approaches. We first compare our proposed approaches with the traditional BoW approach in Equation 1. Soft weighting [Jiang and Ngo 2008] is

Table I. Comparison between different approaches. (IBoW: Informative Bag-of-Visual-Words; HE: Hamming Embedding; Soft: Soft weighting.)

Concept	BoW	IBoW		HE		HE + IBoW		HE +IBoW + Soft	
		AP	Improve	AP	Improve	AP	Improve	AP	Improve
Adult	0.124	0.130	4.55%	0.136	9.49%	0.143	15.12%	0.152	22.46%
Airplane_Flying	0.082	0.093	13.24%	0.091	11.27%	0.101	22.94%	0.104	26.66%
Animal	0.036	0.041	15.30%	0.038	7.18%	0.043	22.00%	0.045	26.51%
Asian_People	0.029	0.030	3.22%	0.033	14.03%	0.035	20.80%	0.038	31.48%
Bicycling	0.062	0.073	17.58%	0.073	**18.38%**	0.082	**32.68%**	0.086	39.03%
Boat_Ship	0.120	0.143	19.41%	0.135	12.47%	0.162	**35.24%**	0.181	50.87%
Building	0.242	0.271	12.33%	0.259	7.34%	0.291	20.57%	0.305	26.15%
Bus	0.072	0.089	**22.76%**	0.079	8.80%	0.097	**34.36%**	0.104	43.80%
Car	0.175	0.206	17.88%	0.193	10.51%	0.229	**31.00%**	0.241	37.71%
Cheering	0.065	0.071	9.98%	0.072	11.08%	0.077	19.92%	0.087	35.05%
Cityscape	0.311	0.364	17.09%	0.335	7.84%	0.382	22.79%	0.389	25.21%
Classroom	0.035	0.044	24.41%	0.037	3.78%	0.043	22.36%	0.045	26.54%
Computer_Screens	0.266	0.281	5.76%	0.291	9.62%	0.311	17.15%	0.328	23.42%
Computers	0.061	0.075	**22.39%**	0.071	15.53%	0.081	**31.50%**	0.086	41.02%
Dancing	0.101	0.105	4.05%	0.104	3.31%	0.111	10.22%	0.114	12.87%
Dark-skinned_People	0.250	0.267	6.80%	0.263	5.33%	0.272	9.14%	0.282	12.89%
Demonstration	0.051	0.063	**23.81%**	0.055	8.19%	0.061	20.61%	0.065	27.73%
Doorway	0.167	0.177	5.82%	0.185	10.87%	0.199	18.89%	0.216	29.02%
Explosion_Fire	0.264	0.289	9.62%	0.299	13.30%	0.317	20.43%	0.327	23.86%
Female_Person	0.196	0.221	12.52%	0.213	8.27%	0.235	19.72%	0.245	24.86%
Female-Face	0.243	0.253	4.00%	0.272	11.94%	0.283	16.42%	0.300	23.40%
Flowers	0.069	0.079	14.39%	0.080	16.14%	0.094	**36.59%**	0.102	48.12%
Ground_Vehicles	0.176	0.291	9.23%	0.214	**21.52%**	0.226	28.44%	0.233	32.73%
Hand	0.090	0.096	6.67%	0.103	13.97%	0.110	22.25%	0.120	32.92%
Helicopter_Hovering	0.014	0.019	**38.90%**	0.017	**22.48%**	0.025	**83.44%**	0.030	116.81%
Indoor	0.168	0.178	5.70%	0.183	8.42%	0.194	14.92%	0.206	22.41%
Indoor_Sports_Venue	0.435	0.460	5.64%	0.494	13.37%	0.530	21.72%	0.552	26.73%
Infants	0.033	0.041	24.16%	0.039	16.53%	0.048	**44.68%**	0.050	51.26%
Instrumental_Musician	0.163	0.172	5.71%	0.188	15.74%	0.203	24.78%	0.222	36.14%
Landscape	0.277	0.291	4.87%	0.303	9.15%	0.321	15.73%	0.330	19.07%
Male_Person	0.118	0.125	5.83%	0.125	5.93%	0.136	14.97%	0.147	24.40%
Military_Base	0.048	0.052	7.06%	0.050	4.63%	0.054	12.02%	0.055	14.71%
Mountain	0.371	0.390	5.26%	0.400	7.99%	0.440	18.66%	0.446	20.38%
News_Studio	0.545	0.572	4.95%	0.564	3.49%	0.592	8.62%	0.617	13.30%
Nighttime	0.232	0.234	0.86%	0.280	**20.64%**	0.284	22.20%	0.308	32.55%
Old_People	0.113	0.117	3.13%	0.130	14.35%	0.133	17.30%	0.149	31.39%
Plant	0.188	0.201	7.01%	0.206	9.45%	0.222	18.40%	0.232	23.38%
Road	0.352	0.381	8.33%	0.371	5.62%	0.398	13.05%	0.411	16.99%
Running	0.093	0.104	12.19%	0.110	**19.08%**	0.123	32.55%	0.135	45.89%
Scene_Text	0.081	0.099	**22.01%**	0.095	16.70%	0.118	**44.90%**	0.122	49.90%
Singing	0.133	0.137	3.12%	0.151	13.51%	0.156	17.11%	0.160	20.37%
Sitting_Down	0.102	0.108	6.10%	0.109	7.12%	0.115	12.67%	0.127	24.34%
Stadium	0.098	0.114	16.12%	0.111	12.91%	0.129	31.77%	0.134	36.62%
Swimming	0.364	0.398	9.33%	0.390	7.12%	0.428	17.61%	0.434	19.17%
Telephones	0.066	0.081	**23.06%**	0.074	12.46%	0.093	**41.49%**	0.100	51.50%
Throwing	0.100	0.103	3.00%	0.109	8.20%	0.112	11.91%	0.118	17.62%
Vehicle	0.131	0.152	16.13%	0.142	8.28%	0.169	29.36%	0.184	40.95%
Walking	0.171	0.186	8.48%	0.190	11.06%	0.203	18.33%	0.219	27.76%
Walking_Running	0.154	0.163	5.30%	0.167	8.20%	0.176	13.94%	0.189	22.07%
Waterscape	0.370	0.404	9.05%	0.444	**19.76%**	0.479	29.18%	0.488	31.81%
MAP	0.165	0.180	**8.92%**	**0.182**	**10.54%**	**0.198**	**20.30%**	**0.208**	**26.27%**

Table II. T-test on the significance of the performance improvement by proposed approaches.

Approach Pair	Paired difference				t	df	Sig. (2-tailed)
	Mean	Std. deviation	Std. Error Mean	95% Confidence Interval			
IBoW - Bow	0.0146	0.0104	0.0015	[0.0116, 0.0175]	9.914	49	0.000
HE - BoW	0.0173	0.0141	0.0020	[0.0133, 0.0213]	8.671	49	0.000
HE+IBoW - BoW	0.0332	0.0220	0.0031	[0.0269, 0.0394]	10.66	49	0.000

then integrated into our approach to further alleviate the mismatch problem caused by SIFT quantization.

Compared to traditional BoW approach, IBoW and Hamming embedding kernel improves the MAP by 8.92% and 10.54% respectively. More importantly, the improvements are consistent for different concepts. By employing the proposed intra-cell distance measure with higher precision, the Hamming embedding kernel shows to be better at discriminating different classes, and this benefits the detection of all concepts. By further integrating IBoW into Hamming embedding kernel (HE+IBoW), an improvement of 20.30% on MAP is achieved compared with the traditional BoW approach. Significant improvement can be observed for concepts *Boat_Ship* (35.24%), *Bicycling* (32.68%), *Car* (31%), *Flowers* (36.59%), *Helicopter_Hovering* (83.44%), *Scene_Text* (44.90%) and *Telephones* (41.49%).

We also combine our approach with soft weighting where one descriptor is assigned to 3 nearest visual words [Jiang and Ngo 2008]. As can be seen in Table I, an improvement of 26.27% is achieved compared with the traditional BoW approach. Soft weighting solves the ambiguity problem when assigning similar descriptors to different Voronoi cells, while our approach addresses the difference between descriptors in the same cell due to the information loss. Our experiments show that these two approaches can complement each other for better BoW representation. Considering that only BoW feature is employed, the overall MAP (0.208) is quite encouraging and competitive. In Table II, we perform paired-samples t-test with IBM SPSS (Statistical Product and Service Solutions) software to validate the significance of the performance improvement by employing the proposed approaches. As can be seen in Table II, by comparing the difference between three approaches (IBoW, HE, and HE + IBoW) and traditional BoW, the $P-$ values (Sig.) are close to 0, which indicate that both Hamming embedding kernel and IBoW can significantly improve the performance of semantic indexing.

In Figure 6, we experiment the effects of various vocabulary sizes on the performance of our approaches. As discussed in Section 2.1, a large vocabulary can somewhat alleviate the information loss problem. However, it will cause more mismatch problems by assigning similar keypoints to different visual words and the performance will be reduced significantly when the vocabulary gets too large. Thus, as can be seen in Figure 6, no apparent relationship between vocabulary size and detection accuracy is observed. For different vocabulary sizes, both Hamming embedding kernel and IBoW improve the performances significantly. Even for a large vocabulary (e.g. with 50000 visual words), Hamming embedding is necessary since each Voronoi cell is still extremely huge by quantizing a 128-dimension space into 50000 or more words.

Table III shows the performances when different l_b values (i.e. the length of the binary signature) are used in Hamming embedding. According to [Charikar 2002; Sibiryakov 2009], the accuracy of similarity estimation can be increased by using more projection vectors or longer binary signatures. Basically, larger l_b value results in more precise representation of the location information in Voronoi cells and distance measure between keypoints. As can be seen in Table III, the performance gets better when l_b increases. The improvement becomes insignificant when $l_b \geq 32$. Thus, in our implementation, we set $l_b = 32$. This is equivalent to segmenting each Voronoi cell into 2^{32} sub-cells, which is precise enough to distinguish different keypoints. Furthermore, larger l_b value results in longer binary signatures for keypoints, which take more space to store the feature vectors.

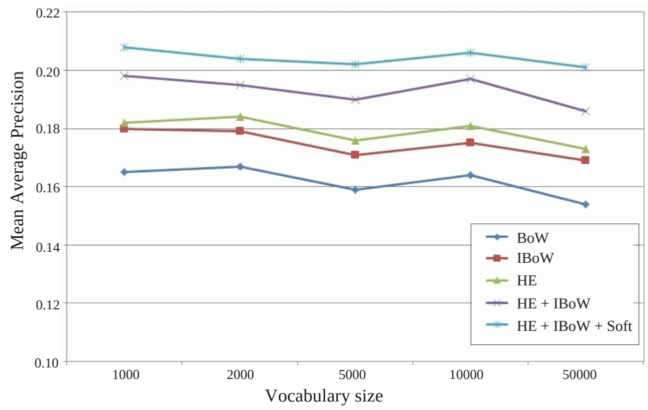

Fig. 6. Performances of different approaches with various vocabulary sizes.

Table III. Determination of parameter l_b in Hamming embedding.

l_b	0	8	16	32	64
MAP	0.165	0.170	0.179	0.181	0.181

For the efficiency issue, Table IV presents the time cost for classifier training and testing with different approaches. The experiment is carried out on IACC.1 dataset and a visual vocabulary of 1000 words is used. A workstation with 4 CPUs, 32GB memory and 8TB hard drive is used for computation. For IBoW, extra computation is needed to weight the informativeness of visual words in training stage compared with traditional BoW approach. The complexity of this algorithm is $O(N^2)$ where N is the number of samples. This procedure could be further speeded up by employing stepwise instead of continuous-value weights for visual words. The time for classification procedure is not significantly increased (except that some additional multiplication operators are inserted in the kernel function). For Hamming embedding kernel, extra time is spent on the matching between keypoint sets with Hungarian algorithm. The complexity of this algorithm is $O(n^3)$ with n being the number of points in the cell. In each image, there are usually hundreds of keypoints in total. Thus, in most cases, only few or at most tens of keypoints are mapped to the same visual word. The worst case (i.e. many points from both images are assigned to the same cell) is seldom encountered. To alleviate the computation load, in our implementation, we use the pre-computed kernel matrix for training so as to avoid repeating the computation in multiple iterations. Besides Hungarian algorithm, some approximation algorithms such as the greedy algorithm could be used to find sub-optimal matching between keypoint sets to speed up this procedure. Overall, although the time cost with HE kernel and IBoW is nearly doubled compared with traditional BoW approach, it is still affordable for semantic indexing even on a large dataset.

Table IV. Total training and testing time (hours) with different approaches.

Approaches	BoW	IBoW	HE	HE + IBoW	HE + IBoW +Soft
Traing	231.4	304.5	360.2	447.6	586.5
Testing	48.4	51.9	75.1	81.8	107.7

To further validate the effectiveness of our proposed approaches, we employ the Hamming embedding kernel and IBoW in TRECVID 2012 Semantic Indexing Task (light submission). In total, 50 concepts are detected and 15 of them are selected by TRECVID for evaluation [Quenot and Awad 2012]. Figure 7 shows the evaluation results of our submissions [Wang, F. et al. 2012]. Among the four submitted runs, color, texture, audio and traditional BoW features are fused as the baseline. Hamming embedding kernel, IBoW, and Soft Weighting are then incorporated into SVM classification for comparison.

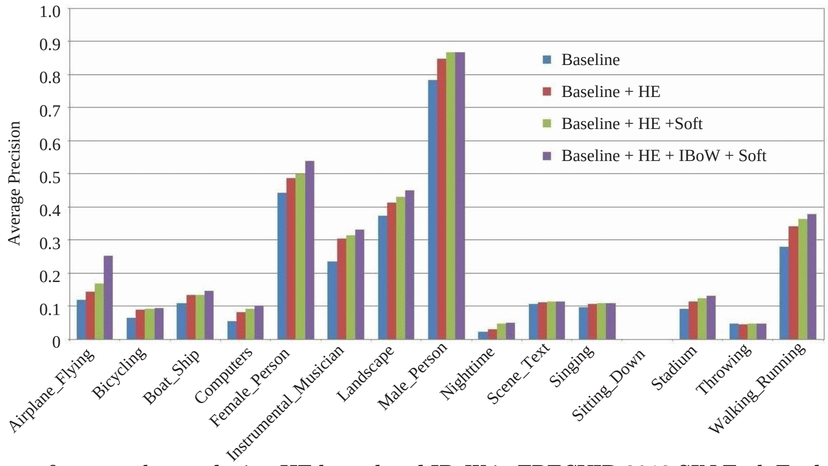

Fig. 7. System performance by employing HE kernel and IBoW in TRECVID 2012 SIN Task Evaluation.

As can be seen in Figure 7, compared with the baseline, Hamming embedding improves the MAP by 14.80%. Among all the evaluated concepts, significant improvement can be observed for the concepts *Airplane_Flying* (20.0%), *Bicycling* (38.5%), *Boat_Ship* (23.9%), *Computers* (46.4%), *Nighttime* (36.4%) and *Instrumental_Musician* (29.4%). By further incorporating soft weighting, another 4.65% improvement is achieved. This improvement is basically consistent with the results reported in [Jiang and Ngo 2008]. Lastly, by employing IBoW, an improvement of 6.26% is gained. Among all concepts, significant improvements are achieved for *Airplane_Flying* (49.1%), *Computers* (11%), *Boat_Ship* (8.9%), and *Nighttime* (8.5%). Noting that the proposed approach is compared with the whole system (instead of only BoW based approach), this improvement could be considered significant.

6. CONCLUSION

We have presented a novel Hamming Embedding Kernel with Informative Bag-of-Visual-Words by addressing two problems existing in traditional Bag-of-Visual-Words approach for semantic indexing. First, by employing Hamming embedding to encode the location information of each descriptor inside the Voronoi cell, our proposed approach measures the distance between samples with higher precision and thus is better at discriminating different classes. Our experiments show that this consistently and significantly improves the performances of concept detection. Furthermore, we also demonstrate that the Hamming embedding kernel and the widely-used soft weighting approach are actually complementary to each other to achieve better BoW representations. Second, by weighting the informativness of each visual word for detecting a given concept, the Informative Bag-of-Visual-Words capture the concept-specific visual information, which proves to be important in SVM classification. In our approach, the weighting of visual words is treated as a kernel optimization problem. This directly connects the weighting procedure and the discriminative abilities of SVM kernels. Compared with other weighting approaches, our algorithm shows to be more effective in capturing the concept-specific visual information and significantly improves the classification accuracy. For future work, we will extend the proposed approaches to other applications such as image categorization and video event detection.

REFERENCES

ALHWARIN, F., WANG, C., RISTIC-DURRANT, D., AND GRASER, A. 2008. Improved SIFT-features matching for object recognition. In *Int. Academic Conf. of Computer Science-BSC*, 179–190.

BATRA, D., SUKTHANKAR, R., AND CHEN, T. 2008. Learning Class-Specific Affinities for Image Labelling. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*.

CAI, H., MIKOLAJCZYK, K., AND MATAS, J. 2011. Learning Linear Discriminant Projections for Dimensionality Reduction of Image Descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence. 33,* 2, 338–352.

CHARIKAR, M. 2002. Similarity Estimation Techniques from Rounding Algorithms. *ACM Symposium on Theory of Computing*.

CRISTIANINI, N., KANDOLA, J., ELISSEEFF, A., AND TAYLOR, J. S. 2002. On Kernel Target Alignment. *Advances in Neural Information Processing Systems. 14*, 367–373.

FULKERSON, B., VEDALDI, A., AND SOATTO, S. 2008. Localizing Objects with Smart Dictionaries. *European Conf. on Computer Vision*, 179–192.

GEMERT, J., GEUSEBROEK, J., VEENMAN, C., AND SMEULDERS, A. 2008. Kernel Codebooks for scene categorization. In *Proceedings of European Conf. on Computer Vision*, 696–709.

GEMERT, J., VEENMAN, C., SMEULDERS, A., AND GEUSEBROEK, J. 2010. Visual Word Ambiguity. *IEEE Trans. on Pattern Analysis and Machine Intelligence. 32,* 7, 1271–1283.

HE, X. AND NIYOGI, P. 2003. Locality Preserving Projections. In *Advances in Neural Information Processing Systems.*.

IGEL, C., GLASMACHERS, T., MERSCH, B., PFEIFER, N., AND MEINICKE, P. 2007. Gradient-based Optimization of Kernel-Target Alignment for Sequence Kernels Applied to Bacterial Gene Start Detection. *IEEE/ACM Trans. on Computational Biology and Bioinformatics. 4,* 2, 216–226.

INDYK, P. AND MOTWANI, R. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of Symposium on Theory of Computing*.

JAIN, M., BENMOKHTAR, R., AND GROS, P. 2012. Hamming Embedding Similarity-based Image Classification. *ACM Int. Conf. on Multimedia Retrieval*.

JAIN, M., JEGOU, H., AND GROS, P. 2011. Asymmetric Hamming Embedding: Taking the Best of Our Bits for Large Scale Image Search. *ACM Multimedia*.

JEGOU, H., DOUZE, M., AND SCHMID, C. 2008. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *Proceedings of European Conf. on Computer Vision*.

JIANG, Y. G., NGO, C. W., AND YANG, J. 2007. Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval. In *Proceedings of Int. Conf. on Image and Video Retrieval*.

JIANG, Y. G., AND NGO, C. W. 2008. Bag-of-Visual-Words Expansion Using Visual Relatedness for Video Indexing. In *Proceedings of ACM SIGIR*.

JIANG, Y. G., ZENG, X., YE, G., ELLIS, D., AND CHANG, S. F. 2010. Columbia-UCF TRECVID 2010 Multimedia Event Detection: Combining Multiple Modalities, Contextual Concepts, and Temporal Matching. *NIST TRECVID Workshop*.

JIANG, Y. G., WANG, J., XUE, X., AND CHANG, S. F. 2013. Query-Adaptive Image Search With Hash Codes. *IEEE Trans. on Multimedia. 15,* 2, 442–453.

JURIE, F., AND TRIGGS, B. 2005. Creating Efficient Codebooks for Visual Recognition. *IEEE Conf. on Computer Vision*, 604–610.

KESORN, K. AND POSLAD, S. 2012. An Enhanced Bag-of-Visual Word Vector Space Model to Represent Visual Content in Athletics Images. *IEEE Trans. on Multimedia. 14,* 1.

KUHN, H. W. 1955. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly. 2,* 83–97.

LOWE, D. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. Journal of Computer Vision. 60*.

MIKOLJCZYK, K. AND SCHMID, C. 2004. Scale and affine invariant interest point detectors. *Int. Journal of Computer Vision. 60*, 63–86.

MOOSMANN, F., NOWAK, E., AND JURIE, F. 2008. Randomized Clustering Forests for Image Classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence. 30,* 9, 1632–1646.

NATARAJAN, P., NATARAJAN, P., WU, S., ZHUANG, X, et al. 2012. BBN VISER TRECVID 2012 Multimedia Event Detection and Multimedia Event Recounting Systems. *NIST TRECVID Workshop*.

NOWAK, E., JURIE, F., AND TRIGGS, B. 2006. Sampling Strategies for Bag-of-Features Image Classification. In *Proceedings of European Conf. on Computer Vision*.

PERRONNIN, F. 2008. Universal and Adapted Vocabularies for Generic Visual Categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence. 30,* 7, 1243–1256.

PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., AND ZISSERMAN, A. 2008. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*.

QUELHAS, P., MONAY, F., ODOBEZ, J., PEREZ, D., AND TUYTELAARS, T. 2007. A Thousand Words in a Scene. *IEEE Trans. on Pattern Analysis and Machine Intelligence. 29,* 9, 1575–1589.

QUENOT, G. AND AWAD, G. 2012. TRECVID 2012 Semantic Indexing Task. *NIST TRECVID Workshop*.

ROWEIS, S. AND SAUL, L. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science. 290,* 5500.

SAUL, L. AND ROWEIS, S. 2003. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research. 4,* 12, 119–155.

SHAKHNAROVICH, G. 2005. Learning Task-specific Similarity. *PhD dissertation, Massachusetts Institute of Technology*.

SIBIRYAKOV, A. 2009. High-entropy Hamming embedding of local image descriptors using random projections. In *Proceedings of IEEE Int. Workshop on Multimedia Signal Processing*.

SIVIC, J., AND ZISSERMAN, A. 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of IEEE Conf. on Computer Vision*, 1470–1477.

SUDDERTH, E., TORRALBA, A., FREEMAN, W., AND WILLSKY, A. 2008. Describing Visual Scenes Using Transformed Objects and Parts. *Int. Journal of Computer Vision. 77,* 1-3, 291–330.

TIRILLY, P., CLAVEAU, V., AND GROS, P. 2008. Language Modeling for Bag-of-Visual-Words Image Categorization. In *Proceedings of Int. Conf. on Content-Based Image and Video Retrieval*, 249–258.

TUYTELAARS, T. AND SCHMID, C. 2007. Vector Quantizing Feature Space with a Regular Lattice. In *Proceedings of IEEE Conf. on Computer Vision*.

WANG, F. AND MERIALDO, B. 2010. Weighting Informativeness of Bag-of-Visual-Words by Kernel Optimization for Video Concept Detection. In *Proceedings of Int. Workshop on Very-Large-Scale Multimedia Corpus, Mining and Retrieval*.

WANG, F., SUN, Z., ZHANG, D., AND NGO, C. W. 2012. Semantic Indexing and Multimedia Event Detection: ECNU at TRECVID 2012. *NIST TRECVID Workshop*.

WANG, J., KUMAR, S., AND CHANG, S. F. 2012. Semi-Supervised Hashing for Large-Scale Search. *IEEE Trans. on Pattern Analysis and Machine Intelligence. 34,* 12, 2393–2406.

WANG, X., ZHANG, L., ZHANG, L., AND JING, F. 2006. Annosearch: Image Auto-annotation by Search. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*.

WINN, J., CRIMINISI, A., AND MINKA, T. 2005. Ojbect Categorization by Learned Universal Visual Dictionary. In *Proceedings of IEEE Conf. on Computer Vision*, 1800–1807.

YANG, L., JIN, R., PANTOFARU, C., AND SUKTHANKAR, R. 2007. Discriminative Cluster Refinement: Improving Object Category Recognition Given Limited Training Data. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*.

YANG, L., JIN, R., SUKTHANKAR, R., AND JURIE, F. 2008. Unifying Discriminative Visual Codebook Generation with Classifier Training for Object Category Recognition. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*.

YUAN, J., WU, Y., AND YANG, M. 2008. Discovery of Collocation Patterns: From Visual Words to Visual Phrases. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*.

YANG, Y. NIE, F. XU, D. AND LUO, J. 2012. A Multimedia Retrieval Framework Based on Semi-supervised Ranking and Relevance Feedback. *IEEE Trans. on Pattern Analysis and Machine Intelligence. 34,* 4, 723–742.

TREC Video Retrieval Evaluation, http://www-nlpir.nist.gov/projects/trecvid/.

Guidelines for TRECVID 2012, http://www-nlpir.nist.gov/projects/tv2012/tv2012.html.

LIBSVM, http://www.csie.ntu.edu.tw/˜cjlin/libsvm/.

Received Month Year; revised Month Year; accepted Month Year