Feedback-based Dynamically Weighted BoF for Image Retrieval

Yanyan Gao[1,a], Yingqian Jia[1,b], Ning Li[1,c], Li Li[1,d]

[1]Shijiazhuang University, 050035, Hebei, China

[a]gaoyanyan4468@aliyun.com, [b]jiayingqian_0318@163.com, [c] Omilint_edu@126.com, [d]lili2008mail@163.com

Keywords: Bag of Features, content-based image retrieval, dynamically weighting, post-query process

Abstract. Bag of Features (BoF) has been successfully exploited in content-based image retrieval for several years. Due to its performance and popularity, several extensions have been proposed that involve feature description, dictionary building, feature encoding and post-query process, etc. This paper proposes a dynamically weighting scheme for BoF-based image retrieval based on feedback. It involves two contributions: (i) analyzing the statistical distribution characteristic of similar BoF representations and (ii) computing weights dynamically based on the feedback obtained from different initial query results. We quantitatively evaluate the proposed method on two different databases. Experiments confirm that the proposed weighting scheme has better performance than the baseline of BoF-based image retrieval systems. Meanwhile, the results demonstrate the effectiveness of the weighting scheme in terms of the precision of top-N returned images.

Introduction

Content-Based Image Retrieval (CBIR) [1,2,3] is facing difficulties with the ever increasing size of databases. Bag of Features (BoF) [4], originated from the textual information retrieval literature, has been playing an important role in the development of CBIR. Many state-of-art image retrieval systems were built upon the BoF representation of images, some shown to work well on large scale databases [4,5]. BoF representation of images involves two main aspects: dictionary building and feature quantization. First proposed by Sivic and Zisserman [6], their BoF-based system returns query result based on a similarity score between fixed length vectors of query and target images. Since then, many researchers have extended on this fundamental approach to include (1) feature detection [7,8]and description improvements [9,10], (2) clustering [4,5] and quantization methods [11,12,13] and (3) result reranking by post-query rank adjustment [14,15]. More recent researches tend to focus on the last two aspects.

In this paper, we focus on the post-query processing of BoF-based retrieval system, and propose a dynamically weighted BoF (DWBoF) by utilizing the information of positive instances labeled in the initial query result. Afterwards, the similarity score is re-computed to rerank the query result. This paper is organized as follows. The related works are summarized in Section 2. Section 3 provides the details of the proposed method, and the experiment results are given in Section 4 followed by the conclusion in Section 5.

Related Works

The concept of using BoF as visual words to build dictionaries has been intensively studied in visual search. The local patches should be obtained either by random sampling [7]. or key-point detection [8], often described by SIFT [8] or other descriptors. In Ref [10], Spatial Pyramid Matching were proposed using SIFT descriptors to describe dense grids with the spacing of 8 pixels.

Another key aspect of BoF representation for images is clustering and quantization. For clustering the features, k-means is the most often used method, and a number of variations have been proposed. For example, Phibin et.al. [4] used approximate k-means to build the dictionary. Nister and Stewenius

[5] proposed a hierarchical k-means to represent the vocabulary tree. After obtaining the dictionary, feature descriptors should be encoded to obtain a vector with fixed length, according to the visual words in the dictionary. Many encoding techniques have also been proposed such as histogram encoding, kernel code-book encoding [11], Fisher encoding [12], locality-constrained linear encoding (LLC) [13], etc. Kernel codebook encoding [11] is a variant in which descriptors are assigned to the visual words in a soft manner, rather than histogram encoding. In essence, Fisher encoding [12] captures the average first and second order differences between the image descriptors and the center of a Gaussian Mixture Model (GMM), where GMM is exploited for clustering in the process of building the dictionary. LLC [13] projects each descriptor down to the local linear subspace spanned by M (M<<dictionary size) visual words closest to the current descriptor.

There also have been several proposals in the literature on improving the performance of BoF-based image retrieval process by post-query processing. Jegou et.al. [14] exploited rank aggregation to improve retrieval results, which performed retrieval for several times using separate vocabularies for each query. Chum et.al. [15] proposed query expansion, which re-submitted the top-ranked results from the initial query as the additional queries in an attempt to increase the recall at a given precision.

Broadly, our work aims to improve retrieval performance based on the initial query. It is motivated by the analyzing the statistical distribution characteristics of the similarities in BoF representations. Section 3 gives the details of our method.

Dynamically Weighting Scheme For BoF

Codebook Generation and Local Feature Encoding. The general procedure of generating a BoF model for representing images is shown in Fig.1 and can be summarized as follows:

(1) Building dictionary: Extract local features for all the images in the dataset using some low feature descriptor, then cluster which to build the dictionary (also called visual codebook) with the cluster center as the term (code) of the dictionary.

(2) Assign terms to generate a BoF representation for a given image.

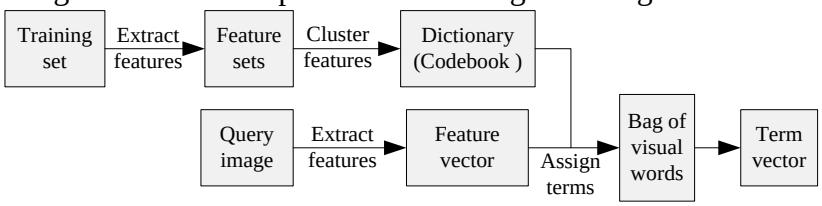
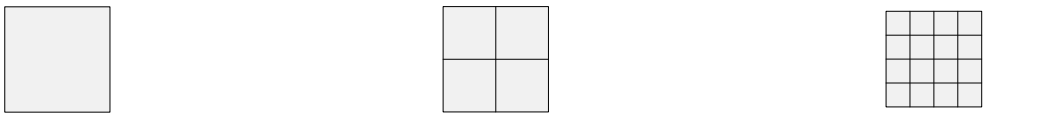


Fig.1 Process of building BoF representation for an image.



Level 0, *weight*=1/4 *Level* 1, *weight*=1/2 *Level* 2, *weight*=1/4

Fig.2 Spatial Pyramid with three levels.

In this paper, Hessian-affine detector[16] is used to obtain local elliptical regions and SIFT descriptor is exploited to describe the extracted regions. Histogram Intersection Kernel (HIK) k-means[17] is then used to generate the codebooks.

In order to take spatial information of images into account, spatial pyramid[10] is employed when encoding local features. Fig.2 shows spatial pyramid with three levels. The quantization is based on hard assignment which projects a local patch onto its nearest word according to the distances between its feature descriptor and words. After assigning each local patch, the algorithm proceeds to count the visual words that fall in each spatial bin, and define their appearance frequency as the histogram descriptor for each bin with weights shown in Fig.2. Finally, the SPBoF representation for an image is

defined by concatenating these histograms in a predefined order. For SPBoF with three levels, if the dictionary size is k, the length of SPBoF descriptor is (1+4+16)*k.

Dynamically Weighted BoF. The distribution of each dimension of BoF vectors is relatively concentrated for similar images, that is, the variance is smaller than for non-similar images. On the other hand, when computing the distance between two images represented by $\boldsymbol{A}$ and $\boldsymbol{B}$, taking an example of Minkowski distance based on l_p norm:

$$dis(\boldsymbol{A},\boldsymbol{B}) = \left[\sum_{l=1}^{n}\left|a_l - b_l\right|^p\right]^{1/p} = \begin{cases} \sum_{i=1}^{n}\left|a_l - b_l\right|, & \text{city-block} \\ \left[\sum_{i=1}^{n}\left|a_l - b_l\right|^2\right]^{1/2}, & \text{Euclidean Distance} \\ \max\left(\sum_{i=1}^{n}\left|a_l - b_l\right|\right), & \text{Chebychv distance} \end{cases} \quad (1)$$

where a_l and b_l represents the elements in the l^{th} dimension of feature descriptors. In the ideal case, the two images are more similar with the smaller difference between a_l and b_l. If the difference is larger in one dimension of the feature vector, we would like the contribution of this dimension for computing the distance to be smaller. If our weighting scheme is considered when computing the similarity measure, the weight on each dimension should be inversely proportional to the variance of this dimension in the feature space. Therefore, we define the weight values as a main contribution of this paper followed by:

$$w_l = \frac{importance_l}{\sum_l importance_l}, \text{ where } importance_l = \frac{1}{1+\sigma_l} \quad (2)$$

where σ_l is the standard deviation of the elements in the l^{th} dimension of feature vectors. Here, we consider post-query processing of the initial query set, so the positive instances are marked to compute the weights for BoF feature as follows.

For each query image, the initial result is obtained by sorting the similarity scores (distances) between it and all images in the database. In this paper, we use χ^2 distance to compute the similarity of two images represented by $\boldsymbol{A}$ and $\boldsymbol{B}$:

$$d_{\chi^2}(\boldsymbol{A},\boldsymbol{B}) = \sum_{l=1}^{n}\frac{\left(a_l - m_l\right)^2}{m_l}, \quad where \ m_l = \frac{a_l + b_l}{2} \quad (3)$$

According to the computed distances, the initial query result is obtained accordingly. $\{\boldsymbol{s}_i\}_{i=1,2,\ldots,m}$ represents m positive samples in the initial retrieval result labeled by users, then compute the weights in the following fashion:

(1) Compute the standard deviation for each dimension of BoF vectors based on m positive labels：

$$\sigma_l = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(s_{il} - \mu_l\right)^2}, \quad \mu_l = \frac{1}{m}\sum_{i=1}^{m} s_{il} \quad (4)$$

s_{il} is the l^{th} element of the BoF vector for the i^{th} labeled image, μ_l and are the corresponding mean value and standard deviation respectively;

(2) Compute the weight value w_l according to Eq.2;

(3) Re-compute the distance:

$$d'_{\chi^2}(\boldsymbol{A},\boldsymbol{B}) = \sum_{l=1}^{n}\frac{\left(w_l a_l - m_l'\right)^2}{m_l'} = \sum_{l=1}^{n}\frac{\left(w_l a_l - w_l m_l\right)^2}{w_l m_l} = \sum_{l=1}^{n}\frac{w_l\left(a_l - m_l\right)^2}{m_l} \quad (5)$$

where $m_l' = \frac{w_l(a_l + b_l)}{2} = w_l \cdot m_l$;

(4) Re-sort the distances to in order to update the retrieval result.

This weighting scheme is dynamic because returned results are different each time, so weights vary for different query images.

Experiments

Dataset and Evaluation Metric. In this paper, we use a subset of the Corel stock photo database, which has 10 classes of 100 images each, and another 10 categories chosen from Caltech 101 database for experiments. Fig.3 illustrates instances of these 20 categories. The first ten images are from Corel, and the rest are from Caltech. The exhaustive search based on the similarity score is performed.

CBIR is a kind of information retrieval problem in essence, so the evaluation metric is used the same as the information retrieval system. Eq.6 gives the definitions of Recall (R) and Precision (P):

$$P = \frac{\text{Number of relevant images retrived}}{\text{Total number of images retrived}},\ R = \frac{\text{Number of relevant images retrived}}{\text{Total number of relevant images in database}} \tag{6}$$

In this paper, we use them and P-R curve to evaluate the performance, which are commonly used to evaluate the retrieval systems [1]. Note that, the search strategy is performed on the two databases separately, and if the returned image and the query image belong to the same category, they are treated as relevant images (similar images). For comparison, in the experiments, we randomly choose 20 images in each class as the query images, and then compute the average precision and recall of all these 20 queries for each category.

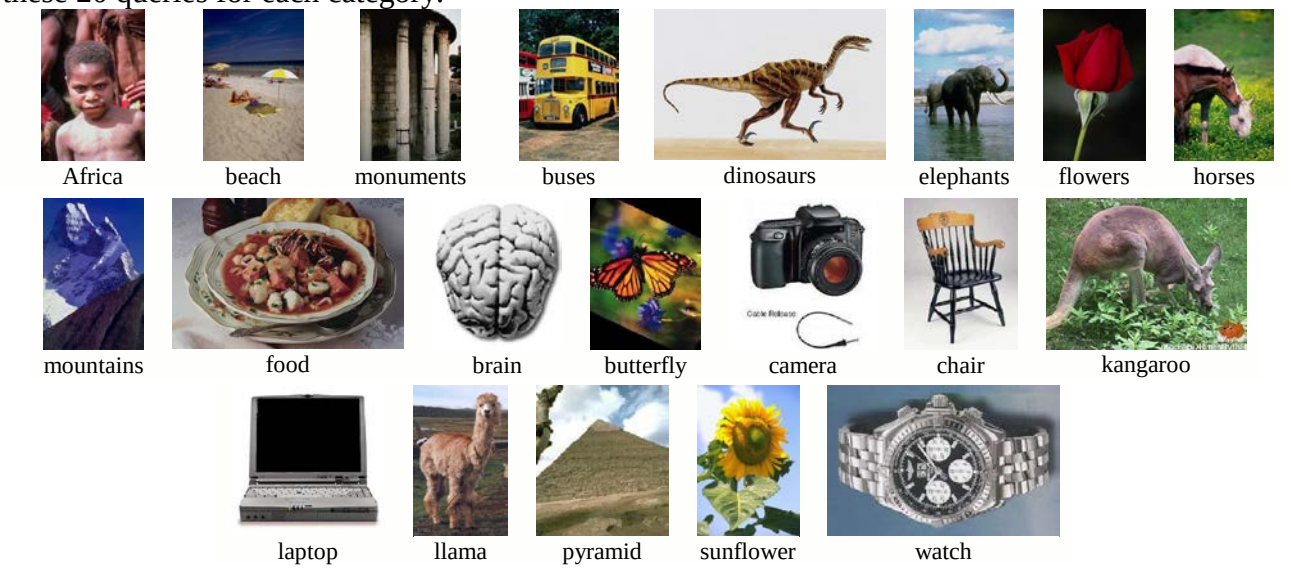


Fig 3. Examples for 20 Categories of Two Different Databases.

Results. In the proposed method, there are two parameters needed to be set: dictionary size *k*, and the number of labeled positive instances *m*. After performing the experiments, we set *k*=100 and *m*=7. The pyramid level is 3 as shown in Fig.2.

Fig.4 gives the average precision at 20 returned images using two different methods: *weighting* represents the result obtained by DWBoF described in Section 3, and *no-weighting* represents the initial query result only by SPBoF without performing the steps described in Section 3.2. The first one is on Caltech, and the other is on Corel. It can be seen in Fig.4 that the proposed *weighting* scheme performs better than *no-weighting* for each category except elephant with 100% precision by both methods. Moreover, the differences are from about 2% up to 10% with variation. Most of the categories have 5% increases except African, elephants, buses and camera. Although the increase for buses is small, the final precision is relatively reasonable.

Fig.5 shows the variation of average PR curves until the recall equal to 50%. The left one is on Caltech, and the other is on Corel. The precision and recall are the average values of all the categories separately computed on two databases. As can be seen, the difference of precision between *weighting* and *no-weighting* decreases as recall increases, and gradually converges. It follows that the proposed weighting scheme can improve the precision of top-N images, and with the

returned number increasing, the precision is converging. Table 1 gives the average precision with different number of returned images by weighting scheme compared with no-weighting on the datasets. The table demonstrates the result obtained from Fig.5. For example, the difference is 11.8 for Caltech with 10 images returned, but the difference is only 1.7 with 100 images returned.

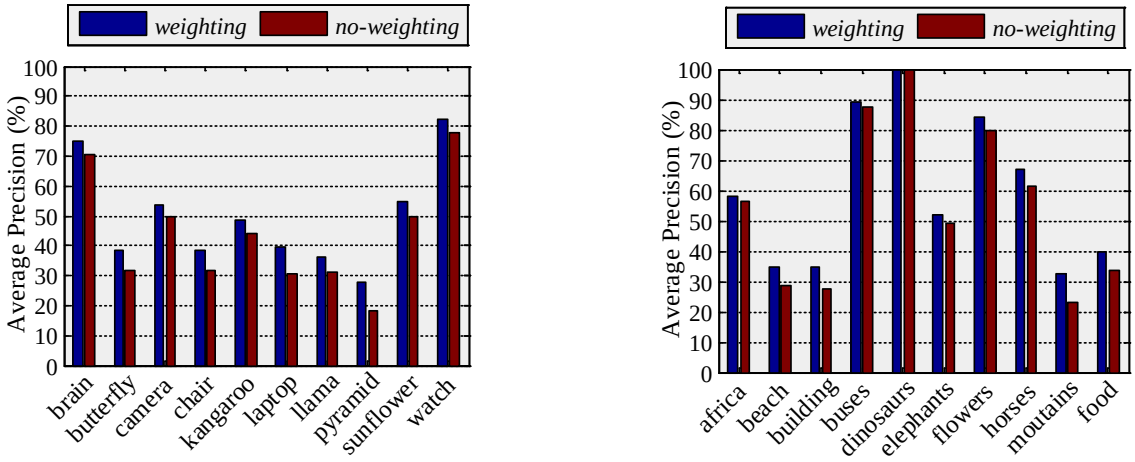


Fig.4 Average Precision@20 Returned Images.

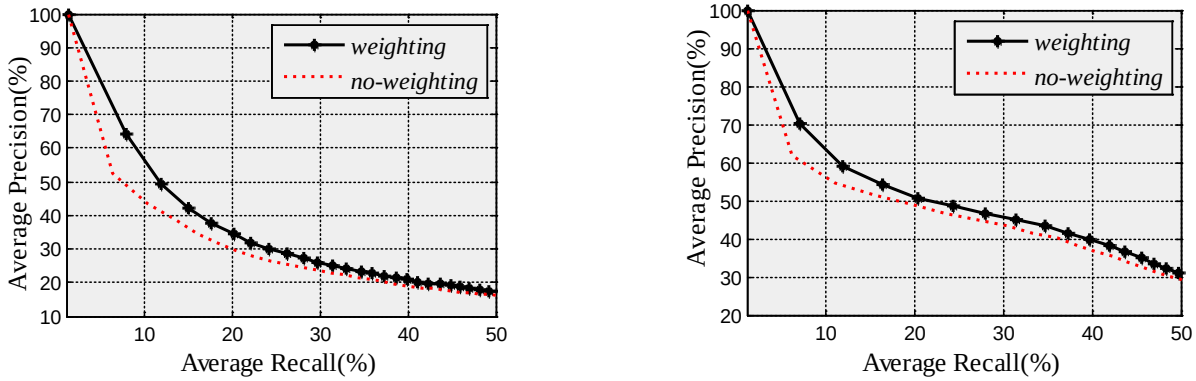


Fig.5 Variation of PR Curves until R=50%.

Table 1. Statistical Precision for All the Queries on Two Datasets.

Number of Returned images	Corel		Caltech	
	weighting	*no-weighting*	*weighting*	*no-weighting*
10	70.3	61.6	64.4	52.6
20	59.2	54.7	49.5	43.7
30	54.2	51.5	42.2	38.7
40	50.8	48.9	37.8	34.7
50	48.5	46.7	34.7	31.8
60	46.5	44.9	32.0	29.6
70	44.8	43.4	29.9	28.0
80	43.3	41.5	28.6	26.6
90	41.4	40.1	27.3	25.5
100	39.7	38.4	26.1	24.4

Conclusion

This paper proposed a dynamically weighting scheme for BoF-based image retrieval. The weighting scheme is based on the statistical distribution of each dimension in the SPBoF representation. In this paper, we utilize *m* labeled positive instances in the initial query result to compute the weight for each query image, and then re-compute the similarity based on the weighted BoF representation to obtain the new query result. The proposed weighting scheme is also dynamic with different query images. In the experiments, we tested our weighting scheme on different

databases using *no weighting* scheme as comparison. Experimental results indicate that the proposed scheme has higher precision than *no-weighting* at 20 returned images and confirm the effectiveness of our technique.

Acknowledgments

This paper is supported by the research initiation funds for doctors of Shijiahzuang University (No.13BS017) and Shijiazhuang Sceience and Technology Project (No.141131231A).

References

[1] R. Datta, D. Joshi, J. Li, J.Z. Wang: Image retrieval: Ideas, influences, and trends of the new age, *ACM Computing Surveys*, vol. 40, no. 2(2008), pp.1-60.

[2] Lin, Jie, et al: Robust fisher codes for large scale image retrieval, International Conference on Acoustics, Speech, & Signal Processing, vol.32(2013), pp.1513-1517.

[3] Zhang, Fan, et al: Ranking-Based Vocabulary Pruning in Bag-of-Features for Image Retrieval, Lecture Notes in Computer Science, 8955(2015), pp.436-445.

[4] D. Nister, H. Stewenius: Scalable recognition with a vocabulary tree, CVPR, vol.2(2006), pp.2161-2168.

[5] J. Philbin, O.Chum, M. Isard, J. Sivic, A. Zisserman: Object retrieval with large vocabularies and fast spatial matching, *CVPR*, (2007), pp.1-8.

[6] J. Sivic and A. Zisserman: Video google: A text retrieval approach to object matching in videos, *ICCV*, vol. 2(2003), pp. 1470-1477.

[7] Zhou, Li, Z. Zhou, and D. Hu: Scene classification using a multi-resolution bag-of-features model, Pattern Recognition， vol.46, no.1(2013), pp.424-433.

[8] D.G. Lowe: Distinctive image features from Scale-Invariant keypoints, *IJCV*, vol. 60, no. 2(2004), pp. 91-110.

[9] Van de Sande KE, Gevers T and Snoek CG: Evaluating color descriptors for object and scene recognition, *PAMI*, vol. 32, no. 9(2010), pp. 1582-1596.

[10] S. Lazebnik, C. Schmid, J.Ponce: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,*CVPR*, vol. 2(2006), pp. 2169-2178.

[11] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, et al: Kernel codebooks for scene categorization, *ECCV*, vol. 5304(2008), pp. 696-709.

[12] F. Perronnin, J. Sánchez, and T. Mensink: Improving the fisher kernel for large-scale image classification, *ECCV*, vol. 6314(2010), pp. 143-156.

[13] J. Wang, J. Yang, K. Yu, et al: Locality-constrained linear coding for image classification, *CVPR*,(2010), pp. 3360-3367.

[14] Jegou H, Schmid C, Harzallah H, et al: Accurate image search using the contextual dissimilarity measure, *PAMI*, vol. 32, no. 1(2010), pp. 2-11.

[15]Chum O, Philbin J, Sivic J, et.al: Total recall: Automatic query expansion with a generative feature model for object retrieval, *ICCV*, (2007), pp. 1-8.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, et al: A comparison of affine region detectors, *IJCV*, vol. 65, no. 1(2005), pp.43-72.

[17] Jianxin, Wu: Efficient HIK SVM learning for image classification, IEEE Transactions on Image Processing， vol.21, no.10(2012), pp.4442-532.