Query-Adaptive Late Fusion for Image Search and Person Re-identification

Liang Zheng[1], Shengjin Wang[1], Lu Tian[1], Fei He[1], Ziqiong Liu[1], and Qi Tian[2]
[1]State Key Laboratory of Intelligent Technology and Systems;
[1]Tsinghua National Laboratory for Information Science and Technology;
[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]University of Texas at San Antonio, TX, 78249, USA
liangzheng06@gmail.com wgsgj@tsinghua.edu.cn qitian@cs.utsa.edu

Abstract

Feature fusion has been proven effective [35, 36] in image search. Typically, it is assumed that the to-be-fused heterogeneous features work well by themselves for the query. However, in a more realistic situation, one does not know in advance whether a feature is effective or not for a given query. As a result, it is of great importance to identify feature effectiveness in a query-adaptive manner.

Towards this goal, this paper proposes a simple yet effective late fusion method at score level. Our motivation is that the sorted score curve exhibits an "L" shape for a good feature, but descends gradually for a bad one (Fig. 1). By approximating score curve's tail with a reference collected on irrelevant data, the effectiveness of a feature can be estimated as negatively related to the area under the normalized score curve.

Experiments are conducted on two image search datasets and one person re-identification dataset. We show that our method is robust to parameter changes, and outperforms two popular fusion schemes, especially on the resistance to bad features. On the three datasets, our results are competitive to the state-of-the-arts.

1. Introduction

This paper[1] considers the task of similar image search, with additional attempts in person re-identification. Given a query (probe) image, we aim at searching for all the similar images in a database (gallery).

Recently, the fusion of multiple features [35, 36, 40] has been pushing the state-of-the-art forward. Ideally, for a given query, if a to-be-fused feature works well by itself *and* is complementary (heterogeneous) to existing features, then it is expected that a higher search accuracy can be achieved. Nevertheless, in realistic settings, the problem is that one does not necessarily know in advance if a heterogeneous feature is good for a given query.

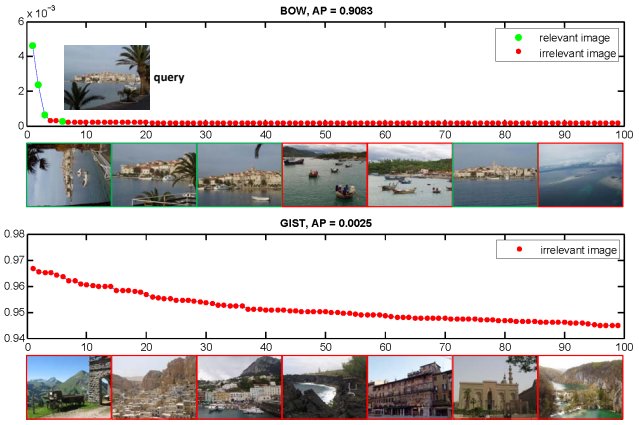

Figure 1. Example of a multi-feature system. For a query in the Holidays [10] dataset, the SIFT (upper) and GIST (bottom) features are employed to obtain two score lists respectively. There are four relevant images for this query, where SIFT produces good performance (AP = 90.83%), but GIST fails (AP = 0.25%). We plot the sorted scores for rank 1-99, and the corresponding 7 top-ranked images. Relevant images are in marked in green, and irrelevant ones red. Note that the sorted score curve is L-shaped for SIFT, but gradually descending for GIST.

Failure in predicting feature effectiveness might result in undesirable search quality. On one hand, the failure of identifying good features may under-utilize features' discriminative power. On the other hand, bad features that escape unpunished may lead to worse consequences: accuracy gets even lower after fusion. This problem is not trivial: some state-of-the-art fusion methods [40, 35], as will be shown, suffer from the fusion of black sheep features.

For this problem, our solution is two-fold. 1) *Query-*

[1]Codes are released at our website: www.liangzheng.com.cn

adaptive. Given a query image, the effectiveness of a to-be-fused feature should be automatically evaluated, so that good features are used, while bad features are ignored. 2) *Unsupervised*. Since we consider generic image search, in which no prior knowledge on the topic of the query image is provided, it is important that we estimate the effectiveness of a feature through unlabeled data [31].

Another issue that should be paid attention to includes the amenability of fusion method to database updating. It requires that the fusion algorithm be independent on the test database, so that its effectiveness can be preserved in an updated database. Although offline calculations are necessary for effective fusion, one should be aware that an image database keeps growing, and it is desirable that the offline steps are not dependent on it. For this issue, the recently proposed methods [36, 35, 4] require expensive offline computations, and the resulting systems are rigid to database change.

In light of the above analysis, this paper proposes a score-level fusion scheme based on a simple motivation (Fig. 1): the score curve of a good feature is "L" shaped, while that of a bad feature is gradually dropping. In a nutshell, the score curves are firstly normalized by reference curves trained on irrelevant data, which are expected to approximate the tails of the initial score curves. Then, feature effectiveness is estimated as negatively related to the area under the normalized score curve (see Fig. 2 for our pipeline). In our method, the offline operation is independent on the test database, making it well suited to dynamic systems. More importantly, our method identifies "good" and "bad" features on-the-fly, and the results are competitive to the state-of-the-arts on three datasets.

The remainder of this paper is organized as follows. First, we briefly review the related works in Section 2. Then, Section 3 introduces the experimental datasets and evaluation protocol. We describe the query-adaptive fusion method in Section 4. The experimental results are presented in Section 5 and conclusions are given in Section 6.

2. Related Work

Basically, there exists two main streams for multiple feature fusion: early and late fusion. In early fusion, descriptors are combined at feature level [15, 27, 2] or even sensor level. Then, combined features are processed together through the learning pipeline. On the other hand, late fusion refers to fusion at score or decision levels [20, 8, 29, 9]. In late fusion, good trade-off can be provided between the information content and the ease in fusion.

In late fusion, Nandakumar *et al.* [20] model the distributions of genuine and impostor match scores as the finite Gaussian mixture model. Jain *et al.* [8] propose to transform the match scores to a common domain and the normalization schemes are data-dependent. The classifier outputs can also be combined using a supervised non-Bayesian method [29] which minimizes classification error under ℓ_1 constraints. For each sample, these methods determine a fixed weight for a specific classifier and does not adapt to sample variations. In [9], user-specific weights are used, but it requires laborious collection of training samples over months. Our method, in essence, belongs to late fusion, and is unsupervised due to the nature of image search.

Feature fusion has been demonstrated as effective in image search. Based on the BoW structure, local features such as color, can be combined with texture [30] or SIFT either by a Bag of Colors (BoC) [33] or coupled Multi-Index (c-MI) [40]. Both methods work on indexing level, using complementary cues to filter out false positive SIFT matches. Zhang *et al.* propose a co-indexing approach [36] to expand the inverted index towards semantic consistency among indexed images. Another good practise consists in propagating the rank list along nodes in a graph [17, 32, 35]. In [35], through link analysis on a fused graph, local and global rank lists are merged with equal weight. In [32], a graph-based learning method is proposed to integrate multiple modalities for visual reranking. These methods, as mentioned, are flawed in either of two aspects. First, complementary features are assumably employed, so there is no fall back if an ineffective feature is integrated. Second, the reranking methods such as [36, 35, 4] heavily rely on the offline steps: all images in the database should be queried and the ranking results are saved. This is potentially problematic if new images are constantly added to the database, and the offline works should be performed all over again.

3. Datasets and Features

3.1. Datasets

Ukbench [21] dataset contains 10,200 images composed of 2,550 groups. Each image is taken as query in turn, and three groundtruth images with extensive variations are provided. We use N-S score as measurement. It counts the number of relevant images in the top-4 ranked images.
Holidays [10] dataset is released with 1,491 personal holiday pictures. There are 500 queries in total. Mean Average Precision (mAP) is used to evaluate search accuracy. It is the mean value of Average Precision (AP), which encodes the area under the precision-recall curve for each query.

3.2. Features

Bag-of-Words (BoW). We adopt the baseline in [10], and the implementation setup in [40]. Namely, Hessian-Affine detector and SIFT descriptor are coupled in feature extraction. A 20k codebook is trained on Flickr60k dataset [10]. We use 128-bit Hamming signature with the Hamming threshold and weighting parameter set to 52 and 26, respectively. We also employ rootSIFT [26], average IDF

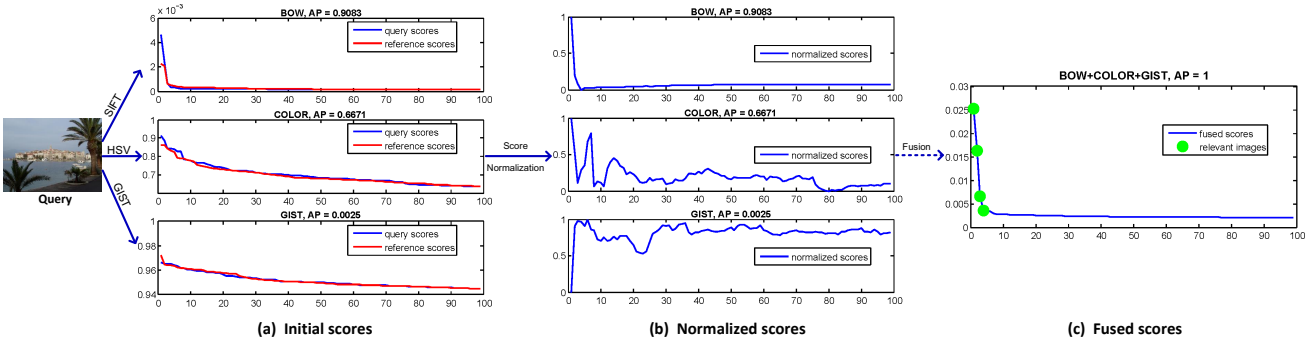

Figure 2. Pipeline of the proposed method. Given a query image, three features (SIFT, HSV, and GIST) are used to obtain initial rank lists. (a) The sorted initial scores are shown for rank 1-99 in blue curve, and the selected reference is depicted in red. (b) The tails of the score curves are eliminated by the reference, and the resulting scores are normalized by min-max normalization. (c) After calculating the feature importance through (b), we obtain the final score curve by Eq. 1. Three features obtain APs of 0.9083, 0.6671, and 0.0025, respectively, and the fusion result is AP = 1. The query-adaptive weights are 0.69, 0.30, and 0.01 for SIFT, HSV, and GIST, respectively.

Datasets	BoW	HS	CNN	GIST	RAND
Holidays, mAP	80.16	61.32	69.33	33.81	13.49
Ukbench, N-S	3.582	3.195	3.397	1.856	1.422

Table 1. Image search accuracy with individual features.

[39], and the burstiness weighting [11]. A standard inverted index is leveraged, and the scores are ℓ_2-normalized.

HSV Histogram. For each image, we compute an ℓ_2-normalized, 1,000-dim HSV histogram. The number of bins for *H*, *S*, *V* are 20, 10, 5, respectively.

Convolutional Neural Network (CNN). We generate an ℓ_2-normalized, 4096-dim CNN descriptor for an input image. Features are extracted from the 6$^{\text{th}}$ layer in the Caffe framework [14].

GIST. We calculate an ℓ_2-normalized, 512-dim GIST [22] descriptor. The images are resized to 256×256. Four scales are used, and the number of orientations for each scale is (8, 8, 8, 8).

Random Projection. To illustrate the robustness of our method to "bad" features, we generate a random transform matrix $P \in \mathbb{R}^{d\times m}$ [34], where d is the target feature dimension (set to 1000 in our experiment), and m is the dimension of the input image (with all pixels concatenated by columns). Entries in P are sampled independently from a zero-mean normal distribution, and each row is ℓ_2 normalized to unit length. In effect, the resulting d-dim feature vector y is computed as $y = Px \in \mathbb{R}^d$, where x is the column-wise input image. Search accuracy of the five features on two datasets is presented in Table 1.

4. Our Method

4.1. Similarity Function

In literature, several strategies are used to combine the scores of multiple features in order to obtain a global confidence measure [16, 1], *e.g.*, the *sum*, *product*, *maximum*, *minimum* rules. Among them, this paper employs the *product rule* for two reasons. First, previous works in biometric multi-modality fusion [16, 1] demonstrate that the product rule has very similar, if not superior, performance to the sum rule. Second, unlike other strategies, the product rule adapts well to input data with various scales and does not require heavily a proper normalization of the data. In fact, in image search, considering the great variety of the query images, one can hardly perform a supervised classifier learning as the case in multi-modal biometrics. So in our work, we choose to merge two score lists by product rule.

Specifically, when K features are fused, given query q and a database image d, the similarity score of d to q *w.r.t* feature $\mathcal{F}^{(i)}, i = 1, ..., K$ is denoted as $s^{(i)}_{d,q}$. Let $w^{(i)}_q, i = 1, ..., K$ encode the weight of feature $\mathcal{F}^{(i)}$ for query q, and has a sum of 1. Then, under product rule, the similarity between q and d is defined as,

$$sim(q,d) = \prod_{i=1}^{K} \left(s^{(i)}_{d,q}\right)^{w^{(i)}_q}, \text{where} \sum_{i=1}^{K} w^{(i)}_q = 1. \quad (1)$$

Note that, Eq. 1 can be transformed into a sum form by $\log(\cdot)$ operator.

4.2. Best and Worst Features

We first describe the extreme cases, *i.e.*, the most desirable and most undesirable features for a given query q. In an image collection with N images, for simplicity, we assume that 1) there is only one relevant image j^* to q and that 2) the image scores are normalized so that the maximum is 1. Intuitively, the best feature satisfies the following criteria,

$$s^{(\text{best})}_{j,q} = \begin{cases} 1, & \text{if } j = j^* \\ 0, & \text{otherwise} \end{cases}, j = 1, 2, ..., N, \quad (2)$$

where $s^{(\text{best})}_{j,q}$ is the score of image j to query q *w.r.t* the best feature. Only the relevant image j^* receives a score of 1,

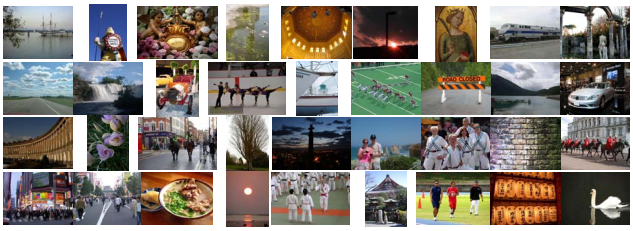

Figure 3. Sample images in the Flickr343Places dataset for reference construction. Images in each row are obtained by the same text query. The queries used to crawl these images are (from top to bottom): "Alexandria", "Canada", "England", and "Tokyo".

and all the irrelevant images 0. To the opposite, the worst feature for query q identifies itself as assigning a score of 0 to image j^* but 1 to the others, *i.e.*,

$$s_{j,q}^{(\text{worst})} = \begin{cases} 0, & \text{if } j = j^* \\ 1, & \text{otherwise} \end{cases}, j = 1, 2, ..., N. \quad (3)$$

The score curves defined by Eq. 2 and Eq. 3, once sorted, exhibit a perfect "L" and a horizontal line, respectively. Ideally, in Eq. 1, weight of the best feature should be $w_q^{(\text{best})} = 1$ and that of the worst feature should be $w_q^{(\text{worst})} = 0$. We find that the weight is negatively related to the area under the sorted score curve.

4.3. Reference Construction

In Fig. 2(a), from the profiles of the three initial score curves, it is quite easy to tell SIFT is a good feature. But the effectiveness of HSV and GIST is not so obvious: both curves has a relatively "high" tail, and scores of the top-ranked images are not remarkably higher than the tail. This is expected, because color and GIST are global features, and there would be more images that share a similar global appearance with the query. In other words, the intrinsic score distribution of a feature is not considered.

In order to alleviate the impact of "high" tails, this paper proposes to find a reference score curve for each query. This reference can be viewed as an approximation to the tail of the initial score curve, and if subtracted, would highlight the protrusion of the top-ranked scores, if any. In practice, we use independent datasets for reference collection. Specifically, for SIFT reference construction, we use the Flickr1M dataset released in [10]. It contains only the SIFT descriptors, which is compatible with the SIFT descriptors used in the test datasets. For the other features, we crawled 1M high-resolution images using the names of 343 countries and regions across the world, called "Flickr343Places" dataset. Images in this dataset vary from scenes to objects and can be viewed as a good sampling of natural images. Some sample images are shown in Fig. 3.

In reference construction, we randomly select Q images as queries. Then, image search is performed in either the Flickr343Places (for HSV, CNN, GIST, and random features) or the Flickr1M (for SIFT) dataset. All the resulting image scores are stored. Together, we have Q score lists for feature $\mathcal{F}^{(i)}$, denoted as $\mathcal{R}^{(i)} = \{r_h^{(i)}\}_{h=1}^{Q}$. Recall that the reference score lists are obtained on a dataset where all images are assumed to be irrelevant to each other. Therefore, the reference can represent the tail distribution of a score curve.

Another consideration is that the collected references should roughly be of the same length with the initial score curve, so that the score distribution would be similar. To this end, if the test database contains N images, we should use roughly N irrelevant images for reference calculation. For large-scale datasets where N is large, both the initial score list and the references are down-sampled before the next step. In this manner, image search efficiency is guaranteed.

4.4. Query-Adaptive Feature Weighting

During online procedure, given query q, the only accessible information we have *w.r.t* feature $\mathcal{F}^{(i)}$ is the sorted score curve $\boldsymbol{s}_q^{(i)}$. The profile of a good feature should take on an "L" shape, while that of a bad feature a gradually descending curve (see Fig. 1 and Fig. 2).

From this observation, we propose to calculate the area under the image score curve, which is taken as the indicator to feature effectiveness. As indicated above, we seek to eliminate the high tail through the subtraction by a proper reference curve. Specifically, given an initial sorted score list $\boldsymbol{s}_q^{(i)}$ obtained by feature $\mathcal{F}^{(i)}$, we aim to find in $\mathcal{R}^{(i)}$ a reference which best matches the tail of $\boldsymbol{s}_q^{(i)}$. For this strategy, the simplest method consists in finding the vector in $\mathcal{R}^{(i)}$ which has the smallest Euclidean distance to $\boldsymbol{s}_q^{(i)}$, *i.e.*,

$$r_q^{(i)^*} = \arg \min_{r_h^{(i)} \in \mathcal{R}^{(i)}} \left\| \boldsymbol{s}_q^{(i)}(u:v) - r_h^{(i)}(u:v) \right\|_2, \quad (4)$$

where $h = 1, 2, ..., Q$, and u, v are parameters that restrict a vector segment on which the nearest neighbor is searched. Basically, it is required that u not be too small, so that the protrusion area is avoided in calculation, and that v be relatively large to capture the tail distribution. Sensitivity to parameters u, v, and Q will be evaluated in Section 5.1.

Alternative to nearest neighbor search, the tail of a score curve can also be approximated by 1) k-nearest neighbor (kNN) search followed by an averaged sum, 2) sparse coding using $\mathcal{R}^{(i)}$ as the codebook. In Section 5.1, we will compare the three methods, *i.e.*, NN, kNN, and sparse coding.

In the next step, the reference is subtracted from the initial score curve of the query,

$$\hat{\boldsymbol{s}}_q^{(i)} = \boldsymbol{s}_q^{(i)} - r_q^{(i)^*}. \quad (5)$$

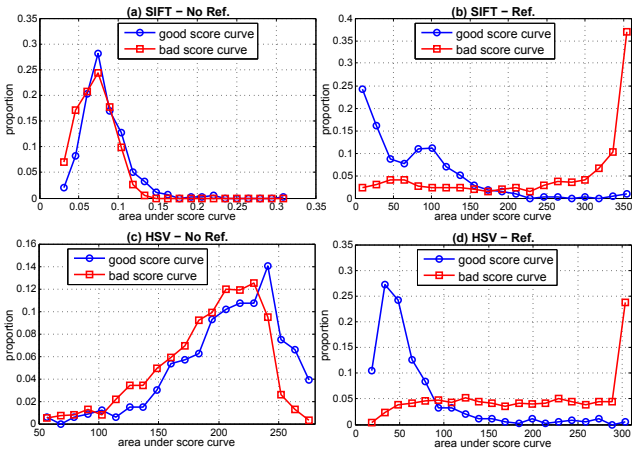

Figure 4. Impact of reference subtraction. We calculate the proportion of good and bad score curves against the area under the score curve. Without reference, for (a) SIFT and (c) HSV features, good and bad curves cannot be distinguished. But when reference is subtracted, for (b) SIFT and (d) HSV features, good and bad curves are clearly separated.

Here, as shown in Fig. 2(b), the reference closely approximates the profile of the tail distribution, so that scores of the top-ranked images can be highlighted in the resulting curve $\hat{\boldsymbol{s}}_q^{(i)}$. Subsequently, $\hat{\boldsymbol{s}}_q^{(i)}$ undergoes min-max normalization,

$$\bar{\boldsymbol{s}}_q^{(i)} = \frac{\hat{\boldsymbol{s}}_q^{(i)} - \min \hat{\boldsymbol{s}}_q^{(i)}}{\max \hat{\boldsymbol{s}}_q^{(i)} - \min \hat{\boldsymbol{s}}_q^{(i)}}, \tag{6}$$

where $\bar{\boldsymbol{s}}_q^{(i)}$ is the normalized score curve to estimate feature effectiveness. To illustrate the working mechanism of reference subtraction, for SIFT and HSV features, we have collected some good and bad score curves from Holidays and Flickr343Places datasets, respectively. Good score curves are those in which rank-1 image is a true match, and bad curves are assured by the irrelevance assumption in Flickr343Places dataset. We calculate the proportion of good and bad score curves against the area under the score curve in Fig. 4. We find that after reference normalization, good queries tend to have a small area under the score curve, and vice versa. In this way, we can roughly tell the effectiveness of a feature after reference subtraction.

For a given query q with K features $\{\mathcal{F}^{(i)}\}_{i=1}^{K}$, we have K score lists $\{\boldsymbol{s}_q^{(i)}\}_{i=1}^{K}$. After normalization to $\{\bar{\boldsymbol{s}}_q^{(i)}\}_{i=1}^{K}$, the query-adaptive weight of feature $\mathcal{F}^{(i)}$ to q is determined as,

$$w_q^{(i)} = \frac{\frac{1}{A_i}}{\sum_{k=1}^{K} \frac{1}{A_k}}, \tag{7}$$

where $A_i, i = 1, ..., K$ represents the area under the i^{th} feature's score curve. We substitute Eq. 7 for Eq. 1 and obtain the desired query-adaptive similarity measurement.

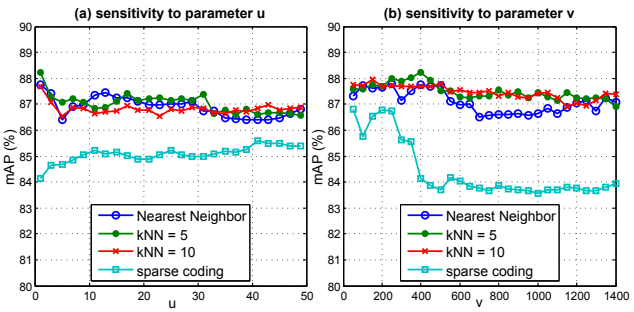

Figure 5. Sensitivity to u and v on Holidays dataset. Five features are fused, *i.e.*, BoW, HSV, CNN, GIST, and Random Projection. mAP is plotted against the two parameters in Eq. 4. Four reference selection methods are compared, *i.e.*, nearest neighbor, kNN (k = 5 or 10), and sparse coding. We find that our method is robust to parameter changes.

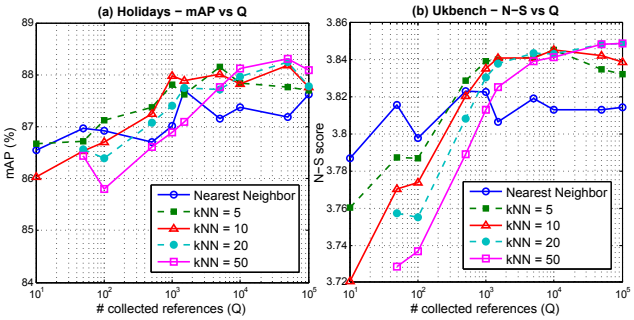

Figure 6. Sensitivity to parameter Q on (a) Holidays and (b) Ukbench datasets. We test kNN = 5, 10, 20, and 50, as well as the nearest neighbor methods. We set $Q = 1000$, and kNN = 10.

Discussion. The proposed method is featured in two aspects. First, for a given query, we estimate a feature's effectiveness in a query-adaptive manner. While existing methods [35, 36] assign fixed weight to all features, our system is more robust to the impact of ineffective features. Second, by constructing a reference codebook offline, the estimation can be performed on-the-fly. Since our method does not require updating the reference codebook, and the nearest neighbor search is very fast, it can be well applied to large-scale and dynamic systems.

5. Experiments

5.1. Image Search Results

Parameter selection. Three parameters are involved in this work. We first evaluate the the matching parameters u, v (Eq. 4), and results are demonstrated in Fig. 5. We can see that, as u and v vary, mAP is relatively stable for "nearest neighbor" and "kNN" methods. Performance of sparse coding is inferior, because the sparse control item has negative impact on the NN search item. Moreover, three NN-based methods perform similarly, and it seems that "kNN = 5" is slightly better. When using kNN, the averaged reference is

Feature Combinations	*Holidays*			*Holidays+1M*	*Ukbench*			
	Graph	Global	Ours	Ours	Co-IDX*	Global*	Ours*	Ours
BoW + GIST	76.39	*81.54*	80.88	67.65	2.766	*3.205*	3.177	3.590
BoW + RAND	76.57	*81.18*	80.91	67.92	2.701	*3.254*	3.210	3.596
BoW + GIST + RAND	70.59	*81.65*	81.47	68.33	2.829	*3.308*	3.263	3.590
BoW + HS	81.58	*84.18*	84.47	72.83	3.504	*3.572*	3.541	3.755
BoW + CNN	83.36	*86.60*	86.27	73.70	3.562	*3.611*	3.624	3.802
BoW + HS + CNN	83.75	*87.23*	87.95	74.96	3.661	*3.677*	3.750	3.840
BoW + GIST + RAND + HS + CNN	81.04	*87.34*	**87.98**	**75.03**	3.608	*3.690*	3.752	**3.841**

Table 2. Results on benchmarks with different fusion methods. We compare our method with Graph Fusion [35], Co-Indexing (Co-IDX) [36], and global weight tuning (Global), respectively. * indicates the case where classic BoW without Hamming Embedding [10] is used.

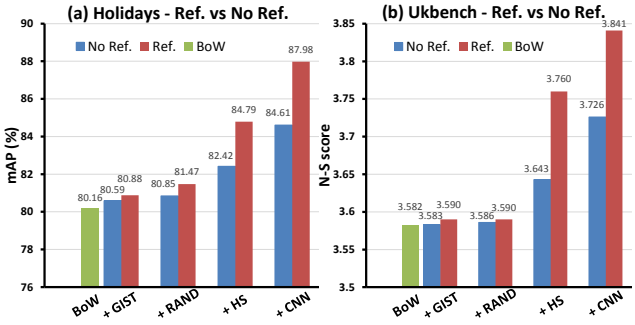

Figure 7. The impact of reference. We compare our method with the case where reference is not used. Four feature combinations are presented, *i.e.*, "BoW + GIST", "BoW + GIST + Random", "BoW + GIST + Random + HS", and "BoW + GIST + Random + HS + CNN". The green bar represents the BoW results, while blue and red bars show results by "No Reference" and "Reference", respectively.

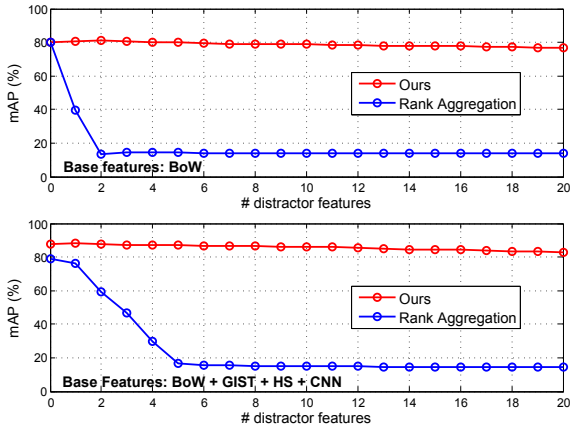

Figure 8. Impact of bad features on Holidays dataset. We plot mAP against a increasing number of random features. **Top:** random features are fused with BoW. **Bottom:** BoW + GIST + HS + CNN is used as baseline. We compare with Rank Aggregation [13].

more resistant to noise; but when k increases, more "bad" references can be introduced especially under small Q. We set $u = 10$ and $v = 400$ in our experiments.

When evaluating parameter Q, *i.e.*, the number of collected references, we present the results in Fig. 6. We find that the fusion accuracy increases steadily with Q. In fact, when we collect a large number of references (large Q), it is more likely to find among them a good approximation to the tail. Nevertheless, computational complexity also increases with Q. Considering this, we choose $Q = 1000$ in our experiments as a trade-off between speed and accuracy.

Impact of using reference. To demonstrate the effectiveness of reference selection, we compare with the case in which no reference is used (No Ref.). In other words, the score curve directly undergoes min-max normalization, and the resulting area is employed for feature weight estimation. The results are shown in Fig. 7. It is clear that, the usage of reference brings benefit for various feature combinations. On Holidays and Ukbench datasets, when all five features are fused, the usage of references brings improvement of +3.37% in mAP, and by +0.115 in N-S, respectively.

Comparison with global parameter tuning. For each feature, we assign to it a global weight $w^{(i)}$. Then, we manually tune $w^{(i)}, i = 1, ..., K$ for two datasets. When fusing five features, we use a step of 0.1 for manual tuning. The results are shown in Table 2. We can see that global tuning exceeds our results when two features are fused. But when using five features, our method is superior. In fact, we find in our experiment that global weighting tuning is very sensitive to weight change: a small change in feature weight would cause intensive accuracy change. Our method determines feature weights automatically, and produces competitive results compared with global tuning.

How about MANY bad features? When a large number of bad features are present, it is desirable that fusion result not be influenced too much. In our experiment, 20 random projection matrices are generated, so that we are provided with 20 random projection features.

We evaluate this property on Holidays dataset in Fig. 8. We compare our method with Rank Aggregation (RA) [13]. In RA, we compute the median rank of each candidate image over all rank lists obtained by different fea-

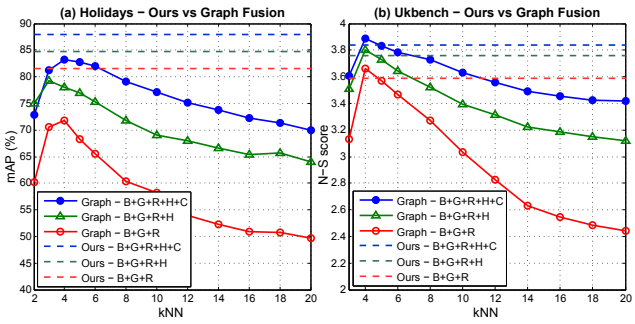

Figure 9. Comparison with graph fusion. On (a) Holidays and (b) Ukbench datasets, three feature combinations are tested. Abbreviations "B", "G", "R", "H", and "C" represent BoW, GIST, Random, HSV, and CNN, respectively. Dashed lines are the results of our method. "kNN" refers to the key parameter in graph fusion.

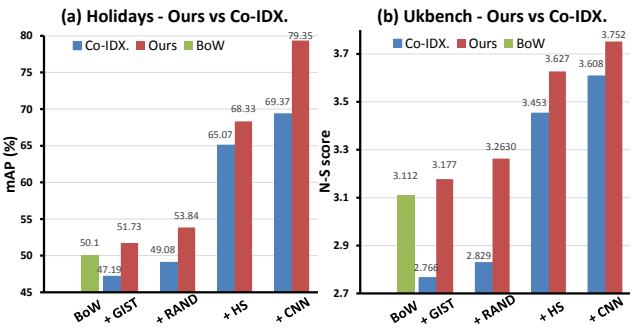

Figure 10. Comparison with co-indexing [36]. Four feature combinations are presented as specified in Fig. 7. The green bar represents the BoW results, while blue and red bars show results by co-indexing and our method, respectively.

tures. We can see that when the number of random features increases, mAP of our method drops very slowly, but that of RA decreases dramatically. When as many as 20 "bad" features are used, mAP of our method drops from 80.16% to 76.58%, and from 87.98% to 82.91% for the two base-feature settings, respectively. In comparison, RA yields an mAP of 13.85% and 14.29%, respectively. Therefore, our method is very robust to "bad" features.

Comparison with other fusion schemes. In order to further verify the strength of our method, results of two state-of-the-art fusion schemes, *i.e.*, Graph Fusion [35] and Co-Indexing [36] are presented in Table 2, Fig. 9, and Fig. 10.

For graph fusion, we use the code released by [35]. Except for the kNN value (different from the kNN in our method), we use the default parameters. Results in Fig. 9 indicate that graph fusion is sensitive to parameter kNN, which, in order to obtain fine accuracy, should be consistent with the average number of groundtruth images in the dataset. For comparison convenience, we plot the corresponding results of our method as the dashed lines.

On Holidays dataset, for each feature combination, our method outperforms graph fusion. On Ukbench, our result is lower than graph fusion only when kNN = 4, which is the ideal parameter setting on Ukbench. Nevertheless, when kNN is set to other values, the performance of graph fusion drops. Moreover, when "bad" features, such as GIST and Random are used, graph fusion does not have a "fall back" mechanism (in fact, it treats all the features as equally important), and the resulting performance could be worse than BoW. Considering that our method is robust to parameters (see Fig. 5 and Fig. 6), we speculate that our method yields more stable and accurate performance than graph fusion.

We also compare our method with co-indexing [36] in Fig. 10. This method is similar to graph fusion in that both require to query all the database images in an offline manner. In our implementation, we choose the optimal parameters, kNN = 3 and 4 for Holidays and Ukbench respectively,

Stage	BoW	Glob. Feat.	Ref. Selection
Avg. Time (s)	1.95	0.96	0.01

Table 3. Average query time of different steps on Holidays + 1M, feature extraction and quantization time excluded.

and the weighting parameter is set to 0.2 as in [36]. For fair comparison, we use the classic BoW model without Hamming Embedding as the baseline.

In Fig. 10, it is clear that on both datasets, our method outperforms co-indexing. Specifically, when bad features such as GIST and Random are fused, our method still yields stable improvement while co-indexing fails.

These results reveals the robustness of our method to the inclusion of bad features. Moreover, since both graph fusion and co-indexing require extensive offline computation, they are not tailored for database updating. In contrast, our method only needs to collect a number of references, which is independent on the test data. As a result, our method suits well to database updating.

Large-scale experiments. We perform large-scale experiment by combining the MirFlickr1M dataset [18] with Holidays dataset. As noted in Section 4.3, we down-sample the initial score lists and references to a length of 1000. Moreover, dimension of all four global features are reduced to 128-D by PCA. The results are shown in Table 2. When five features are combined, we achieve the best mAP of 75.03% on Holidays + 1M dataset. Our experiments are performed on a server with 3.46 GHz CPU and 128 GB memory. CNN features are extracted with a GTX 780 Ti GPU. As shown in Table 3, our method adds little extra time in reference selection. Moreover, the storage of the reference codebook costs only 7.63MB extra memory. Therefore, our method is efficient in terms of memory and time cost.

Comparison with the state-of-the-arts. In Table 4, we compare our results with the state-of-the-art methods. It is shown that our method yields competitive results. Specifically, we achieve **mAP = 88.0%**, **mAP = 75.0%**, and **N-S = 3.84** on Holidays, Holidays + 1M, and Ukbench datasets,

Methods	Ours	[40]	[4]	[35]	[13]	[24]	[28]	[12]	[25]	[11]
Ukbench, N-S score	3.84	**3.85**	3.75	3.77	3.68	3.67	3.56	3.55	-	3.64
Holidays, mAP(%)	**88.0**	85.8	0.847	84.6	-	-	-	84.8	80.1	84.8
Holidays + 1M, mAP(%)	75.0	69	**79.4**	-	-	-	76	42.3	-	77

Table 4. Performance comparison with the state-of-the-art methods. Note that we use the same 1M dataset as [40].

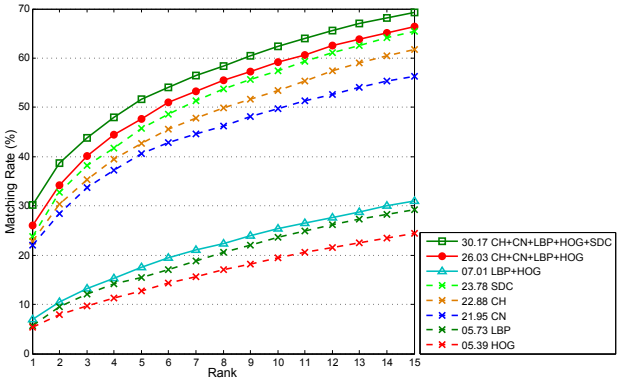

Figure 11. Performance on VIPeR dataset. Results by single features and feature combinations are drawn in dashed and solid lines, respectively. Rank-1 matching rate is shown before the name of each method.

respectively. On Holidays, our result compares favorably with the state-of-the-arts. On Holidays + 1M, we outperform [40] which uses the same 1M dataset. We also find that, on Ukbench, our result is slightly lower than [40] by 0.01 in N-S score. This is because [40] is built on a much higher BoW result (N-S = 3.72 in [40], N-S = 3.58 in this paper). Nevertheless, when combining our method and [40] using the provided code, we achieve N-S = 3.88 on Ukbench.

5.2. Person Re-identification Results

Person re-identification can be viewed as a special case of image search. In this section, we apply the proposed fusion method on this task.

Dataset and evaluation protocol. We use the VIPeR dataset [6] to evaluate our method on person re-identification. This dataset is composed of 632 persons and each has two images captured from two cameras. Persons in this dataset undergo extensive variances in viewpoint, pose, illumination, *etc* and are normalized to 128×48 pixels. VIPeR is randomly divided into two equal halves, one for training, and the other for testing. Each half contains 316 persons. For each person, one image is taken as query, and search is performed in the cross-camera gallery. We use the Cumulative Match Characteristic (CMC) curve as measurement, records the accumulated expectation of correct match at rank-k. Evaluation is repeated for 10 times.

Features. We employ the Bag-of-Words representation [38]. The codebook size is set to 350. Local features are extracted by dense sampling: 4×4 image patches with step of 4. The final descriptor is 5600-dim for each image. We refer the readers to our project page for more details.

In the BoW model, four types of features are separately used for each image patch, *i.e.*, *1) 20-dim H-S histogram (HS)*, *2) 11-dim Color Names (CN)*, *3) LBP*, and *4) HOG*. Moreover, we employ the *5) eSDC [37]* similarity.

Methods	$r = 1$	$r = 5$	$r = 10$	$r = 20$
PRDC [41]	15.66	38.42	53.86	70.09
ELF [7]	12.00	31.00	41.00	58.00
PCCA [19]	19.27	48.89	64.91	**80.28**
SDALF [5]	19.87	38.89	49.37	65.73
eSDC_svm [37]	23.78	45.70	57.48	71.08
Ours	**30.17**	**51.60**	**62.44**	73.81

Table 5. Comparison with the state-of-the-arts on VIPeR dataset. Rank-1, 5, 10, 20 matching rates (%) are presented.

Results. We fuse the aforementioned five features on VIPeR, and results are presented in Table 5 and Fig. 11. We observe consistent improvement multiple features are combined. Specifically, although LBP and HOG yield low matching rate, the fusion of both features still improves recognition accuracy. When eSDC [37] is fused, we obtain rank-1 accuracy of 30.17%. We speculate that when other state-of-the-art systems are integrated [3, 23], our system is capable of achieving even higher results. One issue that should be paid attention to is that since VIPeR has only one relevant image for each query, our method is more effective in improving search accuracy at small r.

6. Conclusion

This paper proposes a score-level fusion scheme featured by two advantages. First, our method estimates the effectiveness of each to-be-fused feature in an unsupervised, query-adaptive manner. This enables "safe" fusion in that ineffective features are unlikely to exert negative impact on the overall accuracy. Second, the offline steps associated with our method are independent on the test database. This makes the fusion scheme compatible to dynamic databases. Experiments on three benchmark datasets demonstrate the strength of our method, and we report competitive results compared with the state-of-the-arts.

Our work highlights the feasibility of score-level fusion with an unsupervised training. In the future, we will further explore the probabilistic nature of score distributions.

Acknowledgement This work was supported in part by the National High Technology Research and Development Program of China (863 program) under Grant 2012AA011004 and in part by the National Science and Technology Support Program under Grant 2013BAK02B04. The work was supported in part to Dr. Qi Tian by ARO grant W911NF-12-1-0057 and Faculty Research Award by NEC Laboratories of America, Inc. This work was supported in part by National Science Foundation of China (NSFC) 61429201.

References

[1] F. M. Alkoot and J. Kittler. Experimental evaluation of expert fusion strategies. *Pattern Recognition Letters*, 20(11):1361–1369, 1999.

[2] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color-and texture-based image segmentation using em and its application to content-based image retrieval. In *ICCV*, 1998.

[3] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In *CVPR*, 2015.

[4] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao. Visual reranking through weakly supervised multi-graph learning. In *ICCV*, 2013.

[5] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010.

[6] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *IEEE International workshop on performance evaluation of tracking and surveillance*. Citeseer, 2007.

[7] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*. 2008.

[8] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005.

[9] A. K. Jain and A. Ross. Learning user-specific parameters in a multibiometric system. In *ICIP*, 2002.

[10] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

[11] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[12] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 2010.

[13] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *PAMI*, 32(1):2–11, 2010.

[14] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org/`, 2013.

[15] F. S. Khan, J. van de Weijer, and M. Vanrell. Modulating shape features by color attention for object recognition. *IJCV*, 2012.

[16] J. Kittler, M. Hatef, R. P. Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):226–239, 1998.

[17] W. Liu, Y.-G. Jiang, J. Luo, and S.-F. Chang. Noise resistant graph ranking for improved web image search. In *CVPR*, 2011.

[18] B. T. Mark J. Huiskes and M. S. Lew. New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In *ACM MIR*, 2010.

[19] A. Mignon and F. Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012.

[20] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain. Likelihood ratio-based biometric score fusion. *PAMI*, 30(2):342–347, 2008.

[21] D. Niester and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[22] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[23] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, 2015.

[24] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, 2011.

[25] D. Qin and C. W. L. van Gool. Query adaptive similarity for large scale object retrieval. In *CVPR*, 2013.

[26] A. Relja and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[27] F. Shahbaz Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez. Color attributes for object detection. In *CVPR*, 2012.

[28] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *CVPR*, 2012.

[29] O. R. Terrades, E. Valveny, and S. Tabbone. Optimal classifier fusion in a non-bayesian probabilistic framework. *PAMI*, 31(9):1630–1644, 2009.

[30] Q. Tian, N. Sebe, M. S. Lew, E. Loupias, and T. S. Huang. Content-based image retrieval using wavelet-based salient points. In *Photonics West 2001-Electronic Imaging*, 2001.

[31] Q. Tian, J. Yu, Q. Xue, and N. Sebe. A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval. In *ICME*, 2004.

[32] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu. Multimodal graph-based reranking for web image search. *Image Processing, IEEE Transactions on*, 21(11):4649–4661, 2012.

[33] C. Wengert, M. Douze, and H. Jégou. Bag-of-colors for improved image search. In *ACM MM*, 2011.

[34] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009.

[35] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *ECCV*, 2012.

[36] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian. Semantic-aware co-indexing for near-duplicate image retrieval. In *ICCV*, 2013.

[37] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013.

[38] L. Zheng, L. Shen, L. Tian, S. Wang, J. Bu, and Q. Tian. Person re-identification meets image search. *arXiv preprint arXiv:1502.02171*, 2015.

[39] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Lp-norm idf for large scale image search. In *CVPR*, 2013.

[40] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Packing and padding: Coupled multi-index for accurate image retrieval. In *CVPR*, pages 1947–1954, 2014.

[41] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, 2011.