Fusion of Global and Local Features with Multi-Inverted Indices for Efficient Image Retrieval

Li Weng

***Abstract*—Feature fusion is an effective solution for improving image retrieval performance. Although the more feature types, the better accuracy, complexity also increases. Applications in practice typically afford a limited number of feature types. Due to the strong complementarity, global and local features form an ideal combination for many fusion applications. However, the two kinds of features are intrinsically different in nature, thus cannot be fused in a straightforward way. In this work, we propose an integrated image retrieval and feature fusion framework for global and local features. It is based on inverted index fusion, a technique for efficient image retrieval. The core idea is to rank candidates by weighted voting during candidate selection, which is named pre-ranking. This procedure takes place before re-ranking, and is potentially superior to conventional late fusion. Extensive experiments on three public datasets show that the light-weight pre-ranking stage significantly contributes to accuracy, and brings substantial improvement when used together with re-ranking. Our method is robust and versatile, and can be applied to any scenario where inverted indexing is used. It is a promising technique for multimedia retrieval in the big data era.**

***Index Terms*—Image retrieval, feature fusion, inverted index, ranking, global and local features.**

I. Introduction

IMAGE retrieval is about finding relevant images in a database according to a query image [1]–[3]. It is one of the basic research topics in the multimedia community [4]. Today, the fast growth of computing devices and communication infrastructures (Internet of Things, 5G, social networks, etc.) has made it extremely easy to generate multimedia content in terms of images, audio, texts, and video. Many new applications have emerged, such as augmented reality, digital humanities, visual localization [5], the Metaverse, etc. Although multimedia has become a part of daily life, and there are massive amounts of available data, we still need more effective ways of utilizing big data. Image retrieval technology is one of the step stones towards this goal.

The performance of image retrieval is characterized by accuracy and efficiency. They are related to the feature representation and the similarity (distance) metric of features, which are the core components of a basic image retrieval pipeline (Fig. 1a). Since applications often favour standard similarity metrics (e.g. Euclidean distance, cosine distance, Hamming distance), the performance essentially depends on the feature. In the last decades, many novel features have been proposed, ranging from hand-crafted features [6]–[9] to deep learning features [10]–[14]. Although new features are still being investigated, one of the lessons learnt so far is that a single feature is not enough to achieve the best accuracy; multiple features are needed to better characterize relevance, which is either perceptual or semantic [15]–[17].

Combining multiple features for performance gain is known as feature fusion [18]. For image retrieval, this is particularly useful for boosting accuracy. Efficiency, on the other hand, is typically neglected or impacted in a negative way, due to extra data processing. Recently, with the emergence of large-scale applications, such as visual localization and autonomous driving, the focus of image retrieval is shifting from accuracy to efficiency. This trend not only needs compact-feature solutions, but also calls for studying the efficiency aspects of feature fusion. In this paper, we focus on feature fusion with efficiency, an area which has not been widely investigated.

For large-scale image retrieval, even if the feature representation is compact, it is impractical to perform linear search (Fig. 1a), i.e., comparing the query with each database item. Efficient solutions typically use indexing structures on top of the conventional pipeline (Fig. 1b). A widely used one is the inverted index [19]. For a database of N images, an inverted index with M cells partitions the feature space into M clusters (aka buckets). On average each bucket contains N/M items. Typically, only items in one or a few buckets are compared with the query, leading to a reduced time complexity of $O(N/M)$. On the other hand, the increased space complexity is $O(M)$. The inverted index can be extended to multi-dimensional cases, aka inverted multi-index (IMI) [20]. If each index dimension corresponds to a different feature, then IMI-like structures can be used for feature fusion, aka coupled indexing [21]. For a two-dimensional IMI with dimensions M_1 and M_2, the time complexity is reduced to $O(N/M_1/M_2)$, but the space complexity is increased to $O(M_1M_2)$. Although IMI is one of the most efficient indexing mechanisms, the space complexity grows exponentially with the dimensionality, so it is not suitable when the number of features is large. A more balanced solution is called fusion of inverted indices (FII) [22], which breaks a high-dimensional indexing structure into linear combinations of single indices. For the two-dimensional case, FII has time complexity $O(N/M_1+N/M_2)$ and space complexity $O(M_1+M_2)$, which can be more easily extended to higher dimensions. Efficient feature fusion should take indexing into account. Based on this motivation, we propose a feature fusion scheme that is integrated within the FII framework. This is the fundamental distinction between our work and most existing solutions.

FII has been used for combining multiple local features [18].

L. Weng is with Zhejiang Financial College, 310018 Hangzhou, China (e-mail:lweng@zfc.edu.cn). This research was partly done at Hangzhou Dianzi University and supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY19F030022.

However, it is not straight-forward to extend the same approach to general cases, which involve both global and local features. In this paper, the main contribution is a more general FII-based fusion scheme for global and local features. Another contribution is an extended three-stage framework for efficient image retrieval. We show that when local features are available, it is possible to rank candidates before re-ranking. This property is not surprisingly new, but has not been thoroughly studied, especially in the context of feature fusion and efficient image retrieval. We name the novel fusion stage *pre-ranking*, and define corresponding performance measures to evaluate its utility. Through extensive experiments, we show that: 1) the proposed fusion method is effective; 2) our pre-ranking method outperforms conventional late fusion schemes that typically take place at re-ranking; 3) the novel three-stage pipeline exhibits superior retrieval performance. In addition, the proposed framework also facilitates flexible trade-offs between accuracy and efficiency. Given reasonable accuracy, the light-weight pre-ranking stage can be used alone in exchange for extra efficiency.

Without loss of generality, we only consider the fusion between two kinds of features – global and local features. We argue that such a combination is versatile and suitable for most application scenarios, mainly for the following reasons:

- Global and local features naturally provide strong complementarity (thus improved accuracy);
- When more than two feature types are used, feature selection might be required (with extra complexity) [23];
- Local features are useful for object-level retrieval, geometric verification, and pose estimation;
- Global features are relatively compact and highly representative;
- Local features enable effective pre-ranking when combined with global ones.

Among these reasons, the third one is particularly important for novel applications such as augmented reality, digital humanities, and visual localization [24]; the last one is interesting for any efficiency-oriented retrieval application, as shown in the paper.

Recently, deep learning features (aka CNN features) are more and more often used as off-the-shelf general-purpose features for various vision applications [12]. Due to their fine representative and discriminative power, we only consider CNN features as candidate global descriptors for our experiments, such as AlexNet [10], VGG [11], and ResNet [13]. On the other hand, the widely used local features are still the classic ones, such as SIFT [7], SURF [8], and ORB [25]. Therefore, we use them as candidate local descriptors for experiments. Apparently, the possibilities of feature combinations are limitless. Although the results in the paper are not exhaustive, they suffice to make a general conclusion.

The rest of the paper is organized as follows: Section II is a brief overview of related work on feature fusion; Section III presents the problem formulation, and gives related definitions; Section IV describes the proposed scheme; Section V is about experiment results, analysis, and discussion; Section VI concludes the work.

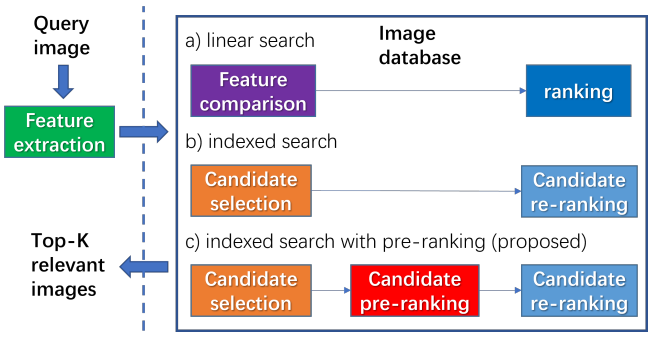

Fig. 1. A schematic diagram of image retrieval. Three different pipelines are illustrated (with increasing scalability).

II. Related Work

Feature fusion is also known as multi-modal or multi-view fusion. In addition to general image retrieval, it is also used for object, video, concept, cross-modal retrieval [14], [26]–[29], etc. Existing approaches can be generally divided into early fusion and late fusion [30]. Early fusion typically takes place at the feature level, and late fusion typically takes place at the decision level [31]. There is also hybrid fusion which utilizes both [32].

The goal of early fusion is to derive a common feature vector from all available features (e.g. [33], [34]). The advantage is the maximum amount of available information, including the correlation among features. According to the information processing inequality [35], the later the fusion, the less information is available. Therefore, early fusion is naturally more advantageous to improving accuracy than late fusion. However, early fusion also introduces huge complexity, which is prohibitive for big data, so large-scale applications typically favor late fusion.

The goal of late fusion is to aggregate the decision results from each feature (e.g. [36], [37]). The advantage is the ease of parallel processing, because each feature type can be individually processed. For retrieval applications, the decision process is about ranking the candidates, so late fusion can be further divided into similarity fusion and rank fusion. The former aggregates similarity scores, and the latter aggregates ranking lists. The basic forms of similarity fusion include linear fusion [38] and non-linear fusion [39], where the overall similarity score is the weighted average of individual scores from all features. In addition to uni-modal similarity scores, cross-modal similarity scores can be included in linear fusion to achieve cross-modal fusion. A cross-modal similarity vector of one modality can be obtained from the similarity vector of another modality through a transformation [40].

A possible extension to similarity fusion is graph-based fusion [41]. If a graph is built for top items according to a query, a retrieval process can be modeled as a random walk [42] or a general diffusion process [43] on the graph. A recent example is [44], where constrained dominant sets are used for similarity fusion. Another example is [45], where local affinity graphs are non-linearly fused and refined by diffusion.

Similarity matrices can also be fused by mapping them to a common latent space, using dimension reduction techniques such as partial least squares [46], [47], and canonical correlation analysis [48].

Rank fusion is another late fusion category, which is highly flexible but also challenging [49]. An early example is CombSUM [50], where the overall relevance score is obtained by summing up scores of sub-queries (similar to linear fusion). It is also common to aggregate rank scores. Representative examples include Borda count fusion [51], Condorcet fusion [52], and reciprocal rank fusion [53]. A recent method uses the Cartesian product of ranked references to build a neighbourhood graph for iterative candidate re-ranking [54]. It is extended to model high-order relevance by incorporating a hypergraph-based similarity measure [55]. In [23], rankers are selected according to effectiveness and correlation measure. In [56], ranked references are combined with supervised ranking algorithms for concept retrieval.

In practice, graph-related feature fusion approaches represent a particular category, which has a fine accuracy level but relatively high complexity. For big data, it is impractical to build complete graphs, so small neighbourhood graphs are typically used (e.g. [44], [45], [54], [55], [57]). These methods can be considered as data-oriented, whereas the other unsupervised methods are data-independent.

The proposed method is data-independent, light-weight, and versatile. Compared with related work, it is novel in a few aspects: 1) it fuses global and (un-aggregated) local features at an early stage; 2) it belongs to late fusion, but takes place before re-ranking; 3) furthermore, it is integrated with multi-inverted indexing, so efficiency is also taken into account.

An inverted index [19] is a data structure with M "buckets". Each bucket is a linear list of image IDs, and associated with a vector $\mathbf{c}_i, i = 0, 1, ..., M-1$. Assume a database of N images and let $\mathbf{f}_i$ denote the feature representation of the ith image. Typically, $\mathbf{c}_i$ is a codeword (cluster center) obtained by applying vector quantization [58] to $\{\mathbf{f}_i\}_{i=0}^{N-1}$.

III. Problem Formulation

Efficient image retrieval schemes typically avoid the naive linear search and adopt a divide-and-conquer strategy. Currently, the de facto image retrieval pipeline is essentially a two-stage framework (Fig. 1b), including the following steps:

- Candidate selection;
- Candidate re-ranking.

During candidate selection, a subset of database items is selected as candidates. Candidates are likely to be relevant to the query, but are typically not ranked at this stage. During candidate re-ranking, the selected candidates are ranked according to a similarity (distance) metric of the feature representation. Finally, the top K candidates are returned to the user.

The number of selected candidates is an important parameter. We denote it by L. Since the number of comparison operations (during re-ranking) is proportional to L, the overall time complexity can be controlled by limiting L. On the other hand, there is a trade-off between efficiency and accuracy. A smaller L also lowers the recall performance. Nevertheless, for large-scale applications, such a compromise is a necessary balance.

A. Utilizing the Inverted Index

A main approach for candidate selection is to use the inverted index [19]. During retrieval, the query's feature $\mathbf{f}_q$ is compared with $\{\mathbf{c}_i\}_{i=0}^{M-1}$ to find the ID k of the nearest bucket (cluster).

$$k = \arg\min_i d(\mathbf{f}_q, \mathbf{c}_i), \tag{1}$$

where $d(\cdot)$ is a distance measure (e.g. Euclidean distance). On average, each bucket contains $L \approx N/M$ image IDs. Once a bucket is identified, the corresponding images are selected as candidates.

If L turns out to be too small, it is possible to check multiple buckets (aka multi-probing [59]) for increasing L. For example, when Hamming distance is used, all the buckets within a small Hamming radius might be checked [60]. Nevertheless, this strategy is not in the scope of this paper, because we focus on the opposite scenario when L is too large. Therefore, we assume only one bucket is checked for each query.

In practice, additional data, such as other feature descriptors or cues, can be embedded in the inverted index (together with image IDs) for *refined* re-ranking, such as Hamming embedding [61], binary embedding [21], semantic-aware co-indexing [57], and IR embedding [29]. The embedded information is typically compact (e.g. binary) to save memory cost. Such an approach can be added to any inverted index system as an extension, but is not used in this work for clarity.

B. The Need for Pre-ranking

As the scale of data continues to grow, the conventional two-stage framework may not suffice. Since L is proportional to the database size, it could happen that L becomes too large for later processing. Therefore, a mechanism is needed to reduce L with fine granularity.

A simple solution is to select the first T ($T < L$) candidates out of L ones, which is almost equivalent to random selection. A more sophisticated solution is to coarsely sort candidates before re-ranking. We define this special sorting procedure as *pre-ranking*. It is different from re-ranking in a few aspects:

- Pre-ranking uses different criteria;
- Pre-ranking should be light-weight;
- Pre-ranking need not be highly accurate.

The basic goal of pre-ranking is to select a subset of candidates in a way that is better than random selection. In particular, it should cost a small time complexity.

The design of a pre-ranking stage introduces new challenges. A naive approach is to use a simplified feature representation in pre-ranking and a complete feature representation in re-ranking. For example, if multiple features are available, one can use a simple feature in pre-ranking and a complex feature in re-ranking. This approach inevitably incurs additional feature comparison, whose time complexity is $O(L)$. Therefore, more efficient solutions are needed.

C. The Complexity of Pre-ranking

According to the properties of pre-ranking, its memory complexity is typically smaller than the one of re-ranking, and is not a critical parameter for systems with adequate storage. In the following, we focus on the time complexity. Assume a general ranking procedure of N items consists of two parts: distance computation and sorting, and the time complexities are $O_1(N)$ and $O_2(N\log_2 N)$ respectively. After adding pre-ranking, the increase in time complexity is an important performance measure, defined as the *complexity gain*:

$$\text{complexity gain} \tag{2}$$

$$= \frac{\text{complexity}\{\text{pre-ranking(L) + re-ranking(T)}\}}{\text{complexity}\{\text{re-ranking(L)}\}} \tag{3}$$

$$= \frac{O_1(L) + O_2(L\log_2 L) + O_1'(T) + O_2(T\log_2 T)}{O_1'(L) + O_2(L\log_2 L)} \tag{4}$$

$$\approx \frac{O_1(L) + O_1'(T)}{O_1'(L)}\ , \tag{5}$$

where L is the length of the initial candidate list for pre-ranking; $T < L$ is the length of the reduced candidate list for re-ranking (complexity denoted by O_1'); the last approximation is derived by neglecting the small complexities of sorting.

Since pre-ranking is simpler than re-ranking, we should have $O_1(L) \ll O_1'(L)$ in the above formulation. If $O_1(L) \leq O_1'(L-T)$, then complexity gain ≤ 1, which is an ideal situation. In practice, even if the complexity gain is larger than one, it is worth using pre-ranking if the accuracy can be improved. In the worst case, when T is the largest possible number of candidates for re-ranking, $L-T$ candidates have to be thrown away, and the complexity gain is upper-bounded by

$$[O_1(L) + O_1'(T)]/O_1'(T)\ , \tag{6}$$

which is used in our experiments.

IV. The Proposed Scheme

We propose a three-stage image retrieval framework for multi-feature scenarios (Fig. 1c). It is mainly based on inverted index fusion [18], [22] and a novel pre-ranking method (Fig. 2). Consider a database of N images, represented by a global feature and a local feature. Denote the feature representations by $\{\mathbf{f}_i^G\}_{i=1}^N$ and $\{\mathbf{f}_i^L\}_{i=1}^N$ respectively. A basic difference between the two is that, for each image, there is typically one global feature representation (vector), whereas there can be many local feature representations (vectors). Therefore, let $\mathbf{f}_i^L = \{\mathbf{f}_{i,j}^L\}_{j=1}^{p_i}$, where $\mathbf{f}_{i,j}^L$ is the jth local feature vector of the ith image, which has p_i local feature vectors in total.

A. Inverted Index Fusion

In inverted index fusion, an inverted index is built for each feature type. Assume the global feature's index has M_G buckets, denoted by $\{\mathbb{B}_i^G\}_{i=1}^{M_G}$; the local feature's index has M_L buckets, denoted by $\{\mathbb{B}_i^L\}_{i=1}^{M_L}$. Let $\{\mathbf{c}_i^G\}_{i=1}^{M_G}$ denote the codewords (cluster centres) for the global feature, and $\{\mathbf{c}_i^L\}_{i=1}^{M_L}$ denote the codewords for the local one.

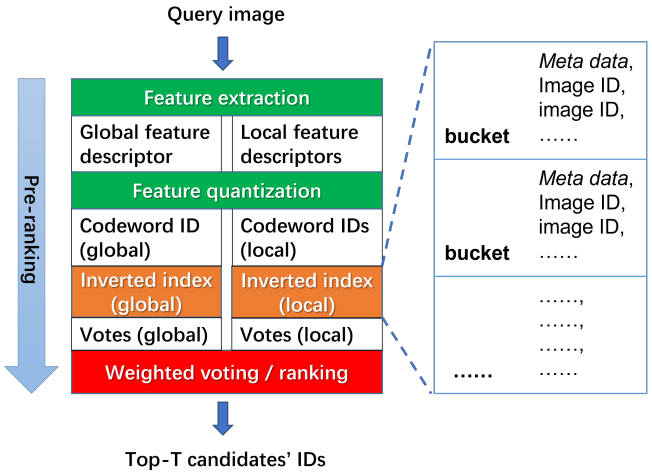

Fig. 2. A schematic diagram of the proposed pre-ranking method.

During retrieval, global and local feature representations are computed from the query. They are denoted by $\mathbf{f}_q^G$ and $\mathbf{f}_q^L = \{\mathbf{f}_{q,j}^L\}_{j=1}^{p_q}$ respectively. The corresponding bucket IDs are found by

$$I^G = \arg\min_i d(\mathbf{f}_q^G, \mathbf{c}_i^G), \tag{7}$$

$$I_j^L = \arg\min_i d(\mathbf{f}_{q,j}^L, \mathbf{c}_i^L), j = 1, \cdots, p_q\ . \tag{8}$$

Finally, the items in the union of identified buckets are selected as candidates:

$$\mathbb{B}_{I^G}^G \cup (\mathbb{B}_{I_1^L}^L \cup \cdots \cup \mathbb{B}_{I_{p_q}^L}^L)\ . \tag{9}$$

This formulation can be extended to more features. It is also possible to use other selection strategies instead of union, such as intersection. Since the union strategy maximizes the recall and enables a fast implementation (see Section V-F), we use this in our experiments. In the following, we model candidate selection as a voting procedure.

B. Pre-ranking by Voting

Since the retrieval procedure involves many feature vectors, each pointing to a bucket of image IDs, we can think of this as a voting procedure. Each feature vector casts a vote to some images. The more votes an image receives, the more likely it is a candidate. This is the basis of our pre-ranking method. In addition, each vote could bear a different *weight*, for two reasons:

- If a codeword occurs more frequently across the database, it should have a lower weight;
- A global feature's codeword should have a higher weight than a local feature's codeword.

The first reason leads to a commonly used weighting strategy, known as the inverse document frequency [19].

$$idf_i = \log(1/f_i) \tag{10}$$

where f_i is the normalized frequency of codeword $\mathbf{c}_i$. The second reason is also intuitive, but this principle has not been used in inverted index fusion. Assume a vote from the global

feature has weight w^G and a vote from the local feature has weight w^L. Taking both into account, we define the weighted votes received by the ith image by

$$v_i = \sum_{j=1}^{M_G} w_j^G \perp(I^G, j) + \sum_{k=1}^{p_q} \sum_{j=1}^{M_L} w_j^L \perp(I_k^L, j) \qquad (11)$$

where $w_j^G = idf_j^G \cdot w^G$, $w_j^L = idf_j^L \cdot w^L$, and $\perp(I, j)$ is an indicator function which equals 1 when $I = j$, otherwise 0. For simplicity, we may let $w^G = 1$ and adjust w^L as a constant parameter.

C. Query-adaptive Voting

The above formulation in (11) is relatively comprehensive. It can adapt to different inverted index structures and different feature combinations. However, one piece of knowledge is still missing: a vote from the same codeword can have a varying weight according to the query, which (if successful) makes a query-adaptive image retrieval framework. In many cases, being query-adaptive can be useful, but solutions are typically designed case by case, due to different scenarios.

In this work, we propose a general query-adaptive strategy by extending the voting framework. It is based on the following motivation:

- If an image has more local feature descriptors, then each of them should give a lower-weighted vote.

This principle is formulated as

$$w^L = 1/N_f \ , \qquad (12)$$

where N_f is the number of local feature regions (points, blobs, etc) detected from a query image.

Since N_f varies from query to query, we achieve query-adaptive voting by combining (11) and (12). In practice, it is necessary to clip w^L when N_f is too large or too small. Therefore, a more sophisticated form of (12) is

$$w^L = \begin{cases} 1/N_{max} & \text{if } N_f > N_{max} \\ 1/N_{min} & \text{if } N_f < N_{min} \\ 1/N_f & \text{otherwise} \end{cases} \qquad (13)$$

where N_{min} and N_{max} are constant parameters. In our experiments, they are empirically set to 20% and 80% quantile points of N_f respectively, according to the statistics of the database.

D. Re-ranking

After pre-ranking, the top T candidates are selected for re-ranking, where more complete feature representations are used for comparison. Any re-ranking method can be used at this stage. Note that, even single-feature based methods can be used here, because re-ranking does not necessarily use all available features. Nevertheless, in order to achieve high accuracy, it is normal to utilize all features. As a baseline, we consider the following distance metric

$$d = d_G(\mathbf{f}_q^{G'}, \mathbf{f}_i^{G'}) \cdot w_f^G + d_L(\mathbf{f}_q^{L'}, \mathbf{f}_i^{L'}) \cdot w_f^L, \quad i = 1, \cdots, T \qquad (14)$$

where $d_G()$, $d_L()$ are two distance metrics, $\mathbf{f}^{G'}$, $\mathbf{f}^{L'}$ denote global and local feature representations for re-ranking respectively, and the indices q, i refer to the query image and the ith candidate image; w_f^G and w_f^L are constant weights that addresses the difference in importance. Note that $\mathbf{f}^{G'}$ is not necessarily the same as $\mathbf{f}^G$ used in pre-ranking, and so is $\mathbf{f}^{L'}$ with respect to $\mathbf{f}^L$. This formulation completes the three-stage image retrieval framework. It is flexible, and also simple enough to reveal the properties of pre-ranking.

In our experiments, $d_G()$ and $d_L()$ are both the cosine distance, $\mathbf{f}^{G'}$ is the same as $\mathbf{f}^G$, and $\mathbf{f}^{L'}$ is the bag-of-words [19] representation of $\mathbf{f}^L$.

The proposed image retrieval framework is summarized in Algorithm 1.

Algorithm 1 Image retrieval with pre-ranking.

Input: a query image.
Output: top K candidates of highest relevance (similarity).
Pre-requisite: an image database with inverted indices, codebooks for quantizing global and local features.
Protocol:

1: Feature extraction (from query):
- Compute query's global feature descriptor $\mathbf{f}_q^G$;
- Compute query's local feature descriptors $\mathbf{f}_{q,j}^L$;
- Obtain query's aggregated local feature descriptor $\mathbf{f}_q^{L'}$.

2: Candidate selection:
- Find bucket IDs with (7), (8);
- Select candidates from the buckets with (9).

3: Candidate pre-ranking:
- Vote the candidates with (11);
- Choose top T candidates according to the votes.

4: Candidate re-ranking:
- Compute query-candidate distances with (14);
- Output top K candidates according to the distances.

E. Evaluation

An image retrieval method is typically evaluated by the precision-recall curve or the mean average precision (mAP). In this work, we mainly use mAP, but focus on the performance gain brought by pre-ranking. In order to justify the three-stage framework, we define "naive pre-ranking" as selecting the first T candidates for re-ranking. A conventional two-stage retrieval scheme can be compared with the proposed one after adding a naive pre-ranking stage. In the following, a few customized performance measures are given. These metrics are defined for the worst case, where only T candidates are allowed, i.e., pre-ranking must be used.

First, we consider the gain in mAP compared with the case of naive pre-ranking:

$$\text{mAP gain} = \frac{\text{mAP}\{\text{pre-ranking + re-ranking}\}}{\text{mAP}\{\text{naive pre-ranking + re-ranking}\}} \ . \qquad (15)$$

where the same re-ranking process is used. It measures how much pre-ranking improves mAP. Second, we consider how much pre-ranking contributes to the overall mAP:

$$\text{mAP contribution} = \frac{\text{mAP}\{\text{pre-ranking}\}}{\text{mAP}\{\text{pre-ranking + re-ranking}\}} \ . \tag{16}$$

It measures the importance of pre-ranking, with respect to a particular way of re-ranking. We argue that, sometimes a pre-ranking strategy with a large mAP contribution might be used alone as a fast ranking method. Finally, we define pre-ranking's *figure of merit*:

$$\text{figure of merit} = \frac{\text{mAP gain}}{\text{complexity gain}} \ , \tag{17}$$

where complexity gain is defined in (6). In general, a good (feasible) retrieval method with a pre-ranking strategy should have its figure of merit larger than one (the larger, the better).

The above measures are also functions of L, T, and K, thus we use notions like $\{\text{mAP@}K\}$ for given parameters.

V. Experiments

The proposed pre-ranking based fusion method has been evaluated with some public datasets. In this section, the datasets, the extracted features, and the baseline methods are first described in details, then followed by extensive experiment results. The experiments are organized in three parts: 1) the effects of fusion compared with single features; 2) comparison with baseline methods; 3) effects of pre-ranking compared with re-ranking. It is worth emphasizing that our goal is not to obtain the highest mAP for a dataset, but to reveal insights of the fusion method and quantify the performance gain.

A. The Datasets

Three classic public datasets are used in the experiments. They are described in the following.

Holidays [61] This dataset contains 1491 images of various scenes. They are divided into 500 queries and 991 corresponding relevant images.

Paris [62] This dataset contains 6392 images of some landmarks in Paris. There are 55 queries of 11 landmarks.

Oxford Buildings [63] This dataset contains 5063 images of some landmarks in Oxford. There are 55 queries of 11 landmarks.

The first dataset is often used for global level image retrieval, whereas the other two are often used for object level image retrieval. They together provide complementary evaluation of the proposed scheme.

B. The Features

A few well-known feature representations (descriptors) are used in the experiments. They are divided into global and local, as listed in Table I. Due to the recent success of convolutional neural networks (CNNs) in many vision applications, we use pre-trained CNN models as feature extractors. All the global features are the outputs of CNNs. On the other hand, the adopted local features are the classic ones, which are still widely used in relevant applications. In particular, we consider AlexNet as a representative global feature and SIFT as a representative local feature.

TABLE I
The extracted features.

feature name	category	dimensionality
AlexNet [10]	global	9216
VGG-16 [11]	global	25088
ResNet-18 [13]	global	1000
SIFT [7]	local	128
SURF [8]	local	64
ORB [25]	local	32

For the global features, the descriptors are directly used in re-ranking. For the local features, the bag-of-words representations of their descriptors are used in re-ranking. In practice, feature descriptors can be made more compact by dimension reduction, e.g. PCA. This additional step might improve the efficiency of retrieval, but is not used here for clarity.

C. The Baselines

Since our fusion process is in the middle of an image retrieval pipeline, it cannot be tested alone when compared with other baselines. In effect, we implement the proposed three-stage framework to examine the fusion performance.

For our fusion method, we consider the same retrieval pipeline with single features as baselines. For example, if feature A and feature B are used for fusion, then two baselines are defined by using A and B separately. Specifically, an inverted index is built for a single feature; candidates are selected by querying the inverted index, and afterwards re-ranked by feature distance computation.

We also consider a few well-known late fusion methods as baselines, which are divided into three categories. The first category is based on similarity/distance fusion, namely linear [38] and non-linear [39] similarity fusion. The second category is based on rank fusion, namely Borda count fusion [51] and reciprocal rank fusion [53]. The third category is based on graphs, namely Cartesian rank product [54], constrained dominant sets [44], and topology correlation [45]. They all have the local neighborhood size set to 20. These methods are listed in Table II with their properties commented.

All these methods are applied at the re-ranking stage, after naive pre-ranking. They are compared with our three-stage image retrieval framework, which is realized by combining pre-ranking and linear similarity fusion. In most experiments, the parameter T, i.e. the maximum number of candidates for re-ranking, is fixed at 200.

D. The Effects of Feature Fusion

In this section, we apply the proposed fusion method to selected features, and compare the retrieval performance with single feature based counterparts. First, we test with the Holidays dataset using the global feature AlexNet and the local feature SIFT. In Table III, mAP values are listed for various

TABLE II
BASELINE METHODS VS. PROPOSED METHOD.

method name	property
single feature	use a single feature
linear fusion [38]	$d = a \cdot d_1 + b \cdot d_2$, $a + b = 1$
non-linear fusion [39]	$d = d_1^a + d_2^b$, $a + b = 1$
Borda count fusion [51]	$s = s_1 + s_2$, $s_i = L - \text{rank}_i + 1$
reciprocal rank fusion [53]	$s = s_1 + s_2$, $s_i = 1/\text{rank}_i$
Cartesian rank product [54]	graph-based methods
constrained dominant sets [44]	
topology correlation [45]	
proposed method	pre-ranking + linear fusion

K, which is the number of retrieved images; AlexNet+SIFT means the two features are fused, otherwise a single feature is used. According to different codebook sizes, the results for two configurations are shown. One observation is that AlexNet works much better than SIFT, which confirms the strong representation power of neural network based features. In order for a BoW representation to perform better, typically a much larger dimensionality (i.e. codebook size) is needed. More importantly, it is clear that the mAP after fusion is significantly higher than using a single feature. Compared with AlexNet, the relative improvement in mAP varies from 10% to 30%.

The same test is also carried out for the Paris dataset and the Oxford dataset, using different feature combinations. The results are shown in Table IV and V respectively for various configurations. These two datasets are generally more difficult than Holidays, and the obtained mAP values are relatively lower. Nevertheless, feature fusion consistently outperforms single features. Since in our experiments global features always outperform local features, we only consider relative improvement in mAP with respect to global features. Sometimes the improvement is significant: for example, in Table IV, from VGG (200) to VGG+SIFT (200, 1000), the relative improvement in mAP varies from 40% to 60%. Sometimes the improvement is not obvious (especially for small K), but becomes more noticeable for large K, e.g. from VGG (500) to VGG+SIFT/SURF (500, 2000). In Table V, the results are more diverse: for ResNet+ORB, the increase in mAP is only a few percent, typically below 0.02. If ORB is replaced with SIFT or SURF, the increase in mAP is larger, up to 0.048 (21%). Therefore, although it is not the focus of the paper, we can roughly say that SIFT/SURF works better than ORB when combined with ResNet (at least for our configuration).

There are other observations from the results. For example, a larger codebook size does not necessarily give a large mAP. This is typical with inverted indices. Although increasing the codebook size generally improves the re-ranking performance, it reduces the number of candidates L per bucket. Without using multi-probing, the recall can be negatively impacted by a large codebook size. In addition, since global features outperform local ones for the same dataset, we have the implication that global features should be given a higher weight.

TABLE III
THE EFFECTS OF FEATURE FUSION (HOLIDAYS DATASET).

	mAP@K			
feature (codebook size)	K=1	K=5	K=10	K=50
AlexNet (200)	.422	.580	.588	.589
SIFT (500)	.077	.108	.112	.116
AlexNet+SIFT (200, 500)	.469	.639	.649	.652
AlexNet (500)	.328	.401	.406	.407
SIFT (1000)	.077	.108	.112	.115
AlexNet+SIFT (500, 1000)	**.397**	**.525**	**.531**	**.535**

TABLE IV
THE EFFECTS OF FEATURE FUSION (PARIS DATASET).

	mAP@K			
feature (codebook size)	K=1	K=5	K=10	K=50
VGG (200)	.005	.024	.039	.103
SIFT (1000)	.001	.004	.007	.018
VGG+SIFT (200, 1000)	**.008**	**.034**	**.056**	**.143**
VGG (500)	.008	.036	.065	.201
SIFT (2000)	.001	.004	.007	.017
VGG+SIFT (500, 2000)	**.008**	**.038**	**.071**	**.230**
SURF (1000)	.001	.004	.007	.024
SURF (2000)	.001	.004	.007	.023
VGG+SURF (200, 1000)	**.008**	**.034**	**.056**	**.148**
VGG+SURF (500, 2000)	**.008**	**.038**	**.070**	**.235**

E. Comparison with Baselines

In this section, we compare the proposed fusion method with some other baseline methods (see Table II). These baselines perform late fusion at the re-ranking stage, thus use the same feature representations. Recall that our fusion method focuses on pre-ranking and does not involve re-ranking. In order to make comprehensive comparison, we examine two versions of our method. The first version is built by incorporating linear fusion as our re-ranking stage after pre-ranking (the same as in the previous section). The second version is simpler: results after pre-ranking are directly

TABLE V
THE EFFECTS OF FEATURE FUSION (OXFORD DATASET).

	mAP@K			
feature (codebook size)	K=1	K=5	K=10	K=50
ResNet (200)	.059	.135	.165	.238
ORB (1000)	.005	.019	.031	.052
ResNet+ORB (200, 1000)	**.059**	**.136**	**.170**	**.246**
ResNet (500)	.059	.140	.168	.225
ORB (2000)	.005	.018	.027	.048
ResNet+ORB (500, 2000)	**.059**	**.140**	**.176**	**.238**
SIFT (1000)	.005	.016	.024	.051
SIFT (2000)	.005	.015	.022	.049
ResNet+SIFT (200, 1000)	**.059**	**.142**	**.173**	**.258**
ResNet+SIFT (500, 2000)	**.059**	**.155**	**.188**	**.273**
SURF (1000)	.005	.025	.040	.070
SURF (2000)	.005	.023	.038	.069
ResNet+SURF (200, 1000)	**.059**	**.143**	**.172**	**.258**
ResNet+SURF (500, 2000)	**.059**	**.149**	**.185**	**.271**

TABLE VI
COMPARISON WITH BASELINE METHODS (HOLIDAYS DATASET).

features: AlexNet+SIFT weights: 0.7, 0.3	mAP@K			
codebook sizes: 500, 500	K=1	K=5	K=10	K=50
linear fusion	.123	.175	.178	.179
non-linear fusion	.126	.173	.176	.177
Borda count fusion	.107	.148	.152	.154
reciprocal rank fusion	.110	.151	.154	.157
Cartesian rank product	.117	.158	.162	.165
constrained dominant set	.105	.145	.149	.153
topology correlation	.070	.124	.127	.129
proposed method	**.382**	**.502**	**.508**	**.511**
proposed method (w/o re-ranking)	**.139**	**.191**	**.200**	**.213**

TABLE VII
COMPARISON WITH BASELINE METHODS (PARIS DATASET).

features: VGG+SURF weights: 0.8, 0.2	mAP@K			
codebook sizes: 500, 1000	K=1	K=5	K=10	K=50
linear fusion	.001	.006	.009	.030
non-linear fusion	.001	.006	.009	.030
Borda count fusion	.001	.005	.009	.028
reciprocal rank fusion	.001	.005	.009	.029
Cartesian rank product	.001	.004	.006	.020
constrained dominant set	.001	.005	.009	.027
topology correlation	.001	.003	.005	.015
proposed method	**.008**	**.037**	**.067**	**.181**
proposed method (w/o re-ranking)	**.005**	**.017**	**.030**	**.097**

TABLE VIII
COMPARISON WITH BASELINE METHODS (OXFORD DATASET).

features: ResNet+ORB weights: 0.6, 0.4	mAP@K			
codebook sizes: 200, 2000	K=1	K=5	K=10	K=50
linear fusion	.007	.023	.035	.059
non-linear fusion	.007	.023	.036	.059
Borda count fusion	.006	.024	.036	.058
reciprocal rank fusion	.006	.024	.037	.063
Cartesian rank product	.006	.022	.037	.065
constrained dominant set	.007	.022	.038	.064
topology correlation	.006	.021	.033	.059
proposed method	**.059**	**.136**	**.172**	**.248**
proposed method (w/o re-ranking)	**.019**	**.038**	**.056**	**.114**

considered as final output, i.e., no re-ranking at all.

We first test with the Holidays dataset. The results are shown in Table VI, which examines AlexNet and SIFT features. The weights for fusion are empirically set to 0.7 and 0.3 for all methods that use weights. It is clear that our method outperforms the baselines by substantial margins. The relative increase in mAP is up to 200%. More surprisingly, even our simplified version beats all the baselines, which implies that pre-ranking is more effective than re-ranking.

More results are shown in Table VII and VIII for Paris and Oxford datasets, where different features and fusion weights are used. The advantage of pre-ranking becomes more obvious: the mAP of our method is approximately an order of magnitude larger than the baselines'. In addition, even without re-ranking, our method still exhibits superior performance. Therefore, the same conclusion can be drawn.

From the results, we can also observe that in spite of small differences, the baselines generally have similar performance. That means the advantage of our method is systematic and by design. The large performance gains are brought by the three-stage architecture, where pre-ranking plays an important role.

Another observation is that, in most cases, graph-based methods perform equally or worse than other baselines. This is contradictory to the impression that they can achieve the highest level of accuracy. One possible explanation is that they might need more than two kinds of features to precisely characterize local neighborhood structures in a latent space. In the original papers, they actually have not been tested with only two feature types. In this sense, fusing two kinds of features is sometimes more challenging than fusing multiple ones.

F. The Role of Pre-ranking

In this section, more experiment results are presented to evaluate the utility of the proposed method. Is it useful for practical applications? This can be measured by the *figure of merit* (17) defined in Section IV-E. In addition, more insights about pre-ranking are revealed by the defined *mAP contribution* (16). In the following, the implementation of pre-ranking is introduced, then more test results are presented.

We find that the union operation in (9) incurs substantial computational cost, so the actual implementation of pre-ranking takes a different approach, while (9) is only a conceptual formulation. In our implementation, each image ID has a counter in a fixed-sized array, which increments according to the votes in (11). This is more efficient, because almost all images get some votes (especially from local features). After voting, top T candidates are selected for re-ranking. This implementation is much faster and used in our experiments. If an intersection-like alternative is used instead of union, we can expect two things: 1) some consensus check is needed to see if a candidate appears in enough votes; 2) it occurs more often that the number of eligible candidates is less than T, which leads to a reduced recall. Therefore, the expected figure of merit would decrease.

We first test with the Holidays dataset. The results are shown in Table IX. Recall that the figure of merit is the ratio between the mAP gain (15) and the complexity gain (2), which should be computed first. The mAP gain is about 3. The complexity gain is 1.16, which is measured on a Laptop with an Intel i5-7300HQ CPU (2.5GHz) and 8GB memory. Compared with the increase in mAP, the extra complexity is almost negligible. The figure of merit also reflects this: it is about 2.5 (larger than one), meaning that the gain in mAP is "worth" the gain in complexity. On the other hand, the mAP contribution of pre-ranking is about 0.4. This is interesting, for a significant part of the final mAP is already achieved before re-ranking.

TABLE IX
THE EFFECTS OF PRE-RANKING (HOLIDAYS DATASET).

features: AlexNet+SIFT weights: 0.7, 0.3	proposed method			
codebook sizes: 500, 500	K=1	K=5	K=10	K=50
mAP gain	3.11	2.87	2.85	2.85
mAP contribution	.36	.38	.39	.42
complexity gain	1.16			
figure of merit	**2.68**	**2.47**	**2.46**	**2.46**

TABLE X
THE EFFECTS OF PRE-RANKING (PARIS DATASET).

features: VGG+SURF weights: 0.8, 0.2	proposed method			
codebook sizes: 500, 1000	K=1	K=5	K=10	K=50
mAP gain	8.00	6.17	7.44	6.03
mAP contribution	.64	.47	.46	.57
complexity gain	1.15			
figure of merit	**6.96**	**5.37**	**6.47**	**5.24**

More results are shown in Table X and XI for Paris and Oxford datasets respectively. We can observe higher mAP gains, and the complexity gain is similar. Thus the figure of merit also increases, which is on average between 5 and 6. Therefore, the improvement brought by our method is substantial. The mAP contribution is still significant – roughly between 0.3 and 0.6. This implies a new trade-off: in occasions where the retrieval speed is a priority and a lower mAP can be tolerated, we might only use pre-ranking without re-ranking.

In addition, we test different weight combinations for global and local features, and also different values of T. Figure 3 shows the variation of mAP for the Holidays dataset. There is a common trend for a fixed T, and $(0.7, 0.3)$ appears to be a reasonable choice. When T is increased, the mAP also increases, which is a sign of pre-ranking functioning properly.

To conclude, our method is not only effective, but also feasible for practical applications. In particular, its usage is flexible: it can be used either with a re-ranking stage, or alone without re-ranking. From the results, we can see that the largest figure of merit typically appears at $K = 1$, which means our method tends to improve retrieval accuracy for small K. This is a desired property for most applications.

TABLE XI
THE EFFECTS OF PRE-RANKING (OXFORD DATASET).

features: ResNet+ORB weights: 0.6, 0.4	proposed method			
codebook sizes: 200, 2000	K=1	K=5	K=10	K=50
mAP gain	8.43	5.91	4.91	4.20
mAP contribution	.31	.28	.32	.46
complexity gain	1.07			
figure of merit	**7.88**	**5.52**	**4.59**	**3.93**

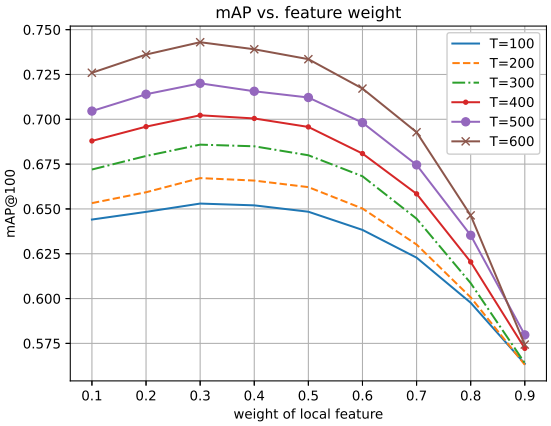

Fig. 3. mAP vs. feature weight for various T (Holidays dataset, AlexNet+SIFT, codebook sizes: 200, 1000).

G. Discussion

The performance of image retrieval depends on many factors. With multiple features, it is hardly possible to find a globally optimal configuration. For our feature fusion study, the goal is to derive a versatile method that offers the benefits of multiple features as easily as possible. In our experiments, sometimes the absolute mAP is small. Two possible reasons are: 1) K is set to small values; 2) raw outputs of pre-trained neural networks are used as global features without post-processing. Both make it sufficiently challenging. However, these settings help to isolate the problem of feature fusion from feature representation, and reflect pure effects of fusion in a critical scenario.

The extensive experiment results have shown the effectiveness, versatility, and robustness of our method. Nevertheless, our solution is not optimal, because (12) or (13) is only an approximate rule that we have discovered. We have observed higher mAP values by slightly tweaking the computed voting weights, but the trials do not lead to particular rules. In practice, there are other parameters to optimize, e.g. the weights for linear fusion, the codebook size, etc., which might require a higher level of data modeling. Additionally, the local feature representation for re-ranking can be changed. For example, the BoW representation can be replaced by the VLAD [64] representation. These options could be of interest for future study, but they seem to be parallel to our method, as indicated by the design principle and the current experiment results.

In the evaluation of our method, naive pre-ranking with parameter T plays an important role. This is a critical step that mimics the bottleneck in reality and influences the performance of baselines. Current experiments are performed under the condition that T is typically small. When T is increased, the advantage of pre-ranking might be diluted, because the ratio of computation in re-ranking is increased. This mechanism is also worth future investigation, which might be generalized to the problem of optimally allocating computation between pre-ranking and re-ranking.

VI. Conclusion

The fusion of global and local features represents a cost-effective approach to boost image retrieval performance. In this work, we propose a novel feature fusion method and a corresponding image retrieval framework based on inverted index fusion. In contrast to existing late fusion methods, we perform fusion before re-ranking. In order to utilize local feature descriptors without aggregation, the fusion process is modeled as an adaptive voting procedure named pre-ranking, where the number of local descriptors per image is used as a natural weight indicator. The method's effectiveness is verified by extensive experiments, where it outperforms several state-of-the-art baselines in various conditions. It is light-weight, robust, and versatile, thus offers a fine starting point towards future image retrieval.

References

[1] Y. Rui, T. S. Huang, and S. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *J. Vis. Commun. Image Represent.*, vol. 10, no. 1, pp. 39–62, 1999. [Online]. Available: https://doi.org/10.1006/jvci.1999.0413

[2] Y. Liu, D. Zhang, G. Lu, and W. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognit.*, vol. 40, no. 1, pp. 262–282, 2007. [Online]. Available: https://doi.org/10.1016/j.patcog.2006.04.045

[3] S. Li, Z. Tu, Y. Chen, and T. Yu, "Multi-scale attention encoder for street-to-aerial image geo-localization," *CAAI Transactions on Intelligence Technology*, vol. 8, no. 1, p. 166–176, jan 2022. [Online]. Available: https://doi.org/10.1049/cit2.12077

[4] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, May 2018.

[5] M. Humenberger, Y. Cabon, N. Pion, P. Weinzaepfel, D. Lee, N. Guérin, T. Sattler, and G. Csurka, "Investigating the role of image retrieval for visual localization," *Int. J. Comput. Vis.*, vol. 130, no. 7, pp. 1811–1836, 2022. [Online]. Available: https://doi.org/10.1007/s11263-022-01615-7

[6] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 404–417.

[9] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008, pp. 1–8.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[11] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014, p. 12.

[12] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, June 2014, pp. 512–519.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[14] F. Liu, C. Gao, Y. Sun, Y. Zhao, F. Yang, A. Qin, and D. Meng, "Infrared and visible cross-modal image retrieval through shared features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 11, pp. 4485–4496, Nov 2021.

[15] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, and A. G. Hauptmann, "Multi-feature fusion via hierarchical regression for multimedia analysis," *IEEE Trans. Multim.*, vol. 15, no. 3, pp. 572–581, 2013. [Online]. Available: https://doi.org/10.1109/TMM.2012.2234731

[16] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "Multi-stream CNN: learning representations based on human-related regions for action recognition," *Pattern Recognit.*, vol. 79, pp. 32–43, 2018. [Online]. Available: https://doi.org/10.1016/j.patcog.2018.01.020

[17] H. Xu, J. Ma, J. Jiang, X. Guo, and H. Ling, "U2fusion: A unified unsupervised image fusion network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 502–518, 2022. [Online]. Available: https://doi.org/10.1109/TPAMI.2020.3012548

[18] N. Bhowmik, L. Weng, V. Gouet-Brunet, and B. Soheilian, "Cross-domain image localization by adaptive feature fusion," in *Proc. of Joint Urban Remote Sensing Event*, 2017, p. 4.

[19] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, Oct 2003, pp. 1470–1477.

[20] A. Babenko and V. Lempitsky, "The inverted multi-index," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1247–1260, June 2015.

[21] L. Zheng, S. Wang, and Q. Tian, "Coupled binary embedding for large-scale image retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3368–3380, Aug 2014.

[22] N. Bhowmik, R. González V., V. Gouet-Brunet, H. Pedrini, and G. Bloch, "Efficient fusion of multidimensional descriptors for image retrieval," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 5766–5770.

[23] L. P. Valem and D. C. G. Pedronette, "Unsupervised selective rank fusion for image retrieval tasks," *Neurocomputing*, vol. 377, pp. 182–199, 2020. [Online]. Available: https://doi.org/10.1016/j.neucom.2019.09.065

[24] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," in *Proc. of International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 667–674.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 2564–2571.

[26] D. Zhong and S.-F. Chang, "An integrated approach for content-based video object segmentation and retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1259–1268, Dec 1999.

[27] C. Lacoste, J.-H. Lim, J.-P. Chevallet, and D. T. H. Le, "Medical-image retrieval based on knowledge-assisted text and image indexing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 889–900, July 2007.

[28] Y.-H. Yang, W. H. Hsu, and H. H. Chen, "Online reranking via ordinal informative concepts for context fusion in concept detection and video search," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 12, pp. 1880–1890, Dec 2009.

[29] K. Liao, H. Lei, Y. Zheng, G. Lin, C. Cao, M. Zhang, and J. Ding, "IR feature embedded BOF indexing method for near-duplicate video retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 12, pp. 3743–3753, Dec 2019.

[30] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *ACM International Conference on Multimedia*, New York, NY, USA, 2005, pp. 399–402. [Online]. Available: http://doi.acm.org/10.1145/1101149.1101236

[31] I. Gialampoukidis, E. Chatzilari, S. Nikolopoulos, S. Vrochidis, and I. Kompatsiaris, *Big Data Analytics for Large-Scale Multimedia Search*. Wiley, 2019, ch. Multimodal Fusion of Big Multimedia Data, pp. 121–156.

[32] Z. Lan, L. Bao, S. Yu, W. Liu, and A. G. Hauptmann, "Multimedia classification and event detection using double fusion," *Multim. Tools Appl.*, vol. 71, no. 1, pp. 333–347, 2014. [Online]. Available: https://doi.org/10.1007/s11042-013-1391-2

[33] J. Magalhães and S. M. Rüger, "An information-theoretic framework for semantic-multimedia retrieval," *ACM Trans. Inf. Syst.*, vol. 28, no. 4, pp. 19:1–19:32, 2010. [Online]. Available: https://doi.org/10.1145/1852102.1852105

[34] J. C. Caicedo, J. Ben-Abdallah, F. A. González, and O. Nasraoui, "Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization," *Neurocomputing*, vol. 76, no. 1, pp. 50–60, 2012. [Online]. Available: https://doi.org/10.1016/j.neucom.2011.04.037

[35] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 2006.

[36] E. Younessian, T. Mitamura, and A. G. Hauptmann, "Multimodal knowledge-based analysis in multimedia event detection," in *International Conference on Multimedia Retrieval (ICMR)*. ACM, 2012, p. 51. [Online]. Available: https://doi.org/10.1145/2324796.2324855

[37] I. Kitanovski, G. Strezoski, I. Dimitrovski, G. Madjarov, and S. Loskovska, "Multimodal medical image retrieval system," *Multim. Tools Appl.*, vol. 76, no. 2, pp. 2955–2978, 2017. [Online]. Available: https://doi.org/10.1007/s11042-016-3261-1

[38] P. K. Atrey, M. A. Hossain, A. El-Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multim. Syst.*, vol. 16, no. 6, pp. 345–379, 2010. [Online]. Available: https://doi.org/10.1007/s00530-010-0182-0

[39] B. Safadi, M. Sahuguet, and B. Huet, "When textual and visual information join forces for multimedia retrieval," in *International Conference on Multimedia Retrieval (ICMR)*, 2014, p. 265. [Online]. Available: https://doi.org/10.1145/2578726.2578760

[40] J. Ah-Pine, M. Bressan, S. Clinchant, G. Csurka, Y. Hoppenot, and J. Renders, "Crossing textual and visual content in different application scenarios," *Multim. Tools Appl.*, vol. 42, no. 1, pp. 31–56, 2009. [Online]. Available: https://doi.org/10.1007/s11042-008-0246-8

[41] I. Gialampoukidis, A. Moumtzidou, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Retrieval of multimedia objects by fusing multiple modalities," in *ACM International Conference on Multimedia Retrieval (ICMR)*, 2016, pp. 359–362. [Online]. Available: https://doi.org/10.1145/2911996.2912068

[42] W. H. Hsu, L. S. Kennedy, and S. Chang, "Video search reranking through random walk over document-level context graph," in *International Conference on Multimedia*, 2007, pp. 971–980. [Online]. Available: https://doi.org/10.1145/1291233.1291446

[43] J. Ah-Pine, G. Csurka, and S. Clinchant, "Unsupervised visual and textual information fusion in CBMIR using graph-based methods," *ACM Trans. Inf. Syst.*, vol. 33, no. 2, pp. 9:1–9:31, 2015. [Online]. Available: https://doi.org/10.1145/2699668

[44] L. T. Alemu and M. Pelillo, "Multi-feature fusion for image retrieval using constrained dominant sets," *Image Vis. Comput.*, vol. 94, p. 103862, 2020. [Online]. Available: https://doi.org/10.1016/j.imavis.2019.103862

[45] Y. Li, X. Kong, H. Fu, and Q. Tian, "Node-sensitive graph fusion via topo-correlation for image retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3777–3787, Oct 2020.

[46] B. Siddiquie, B. White, A. Sharma, and L. S. Davis, "Multi-modal image retrieval for complex queries using small codes," in *International Conference on Multimedia Retrieval (ICMR)*, 2014, p. 321. [Online]. Available: https://doi.org/10.1145/2578726.2578767

[47] I. Gialampoukidis, A. Moumtzidou, D. Liparas, T. Tsikrika, S. Vrochidis, and I. Kompatsiaris, "Multimedia retrieval based on non-linear graph-based fusion and partial least squares regression," *Multim. Tools Appl.*, vol. 76, no. 21, pp. 22 383–22 403, 2017. [Online]. Available: https://doi.org/10.1007/s11042-017-4797-4

[48] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *International Conference on Multimedia*, 2010, pp. 251–260. [Online]. Available: https://doi.org/10.1145/1873951.1873987

[49] S. Wei, Y. Zhao, Z. Zhu, and N. Liu, "Multimodal fusion for video search reranking," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 8, pp. 1191–1199, 2010. [Online]. Available: https://doi.org/10.1109/TKDE.2009.145

[50] E. A. Fox and J. A. Shaw, "Combination of multiple searches," in *Proceedings of The Second Text REtrieval Conference (TREC)*, ser. NIST Special Publication, D. K. Harman, Ed., vol. 500-215. National Institute of Standards and Technology (NIST), 1993, pp. 243–252. [Online]. Available: http://trec.nist.gov/pubs/trec2/papers/ps/vpi.ps

[51] J. A. Aslam and M. H. Montague, "Models for metasearch," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001, pp. 275–284. [Online]. Available: https://doi.org/10.1145/383952.384007

[52] M. H. Montague and J. A. Aslam, "Condorcet fusion for improved retrieval," in *ACM CIKM International Conference on Information and Knowledge Management*, 2002, pp. 538–548. [Online]. Available: https://doi.org/10.1145/584792.584881

[53] G. V. Cormack, C. L. A. Clarke, and S. Büttcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 758–759. [Online]. Available: https://doi.org/10.1145/1571941.1572114

[54] L. P. Valem, D. C. G. Pedronette, and J. Almeida, "Unsupervised similarity learning through cartesian product of ranking references," *Pattern Recognit. Lett.*, vol. 114, pp. 41–52, 2018. [Online]. Available: https://doi.org/10.1016/j.patrec.2017.10.013

[55] D. C. G. Pedronette, L. P. Valem, J. Almeida, and R. da Silva Torres, "Multimedia retrieval through unsupervised hypergraph-based manifold ranking," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5824–5838, 2019. [Online]. Available: https://doi.org/10.1109/TIP.2019.2920526

[56] H. F. Yang, K. Lin, and C. S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 437–451, Feb. 2018.

[57] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, "Semantic-aware co-indexing for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2573–2587, 2015. [Online]. Available: https://doi.org/10.1109/TPAMI.2015.2417573

[58] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[59] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-probe LSH: Efficient indexing for high-dimensional similarity search," in *Proc. of 33rd International Conference on Very Large Data Bases (VLDB)*, Vienna, Austria, 2007, pp. 950–961.

[60] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet, "A privacy-preserving framework for large-scale content-based information retrieval," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 152–167, 2015.

[61] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*, 2008, pp. 304–317.

[62] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. [Online]. Available: https://doi.org/10.1109/CVPR.2008.4587635

[63] ——, "Object retrieval with large vocabularies and fast spatial matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. [Online]. Available: https://doi.org/10.1109/CVPR.2007.383172

[64] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 3304–3311.