Dynamic Match Kernel with Deep Convolutional Features for Image Retrieval

Jufeng Yang, Jie Liang, Hui Shen, Kai Wang, Paul L. Rosin, Ming-Hsuan Yang

***Abstract*—For image retrieval methods based on bag of visual words, much attention has been paid to enhancing the discriminative powers of the local features. Although retrieved images are usually similar to a query in minutiae, they may be significantly different from a semantic perspective, which can be effectively distinguished by convolutional neural networks (CNN). Such images should not be considered as relevant pairs. To tackle this problem, we propose to construct a dynamic match kernel by adaptively calculating the matching thresholds between query and candidate images based on the pairwise distance among deep CNN features. In contrast to the typical static match kernel which is independent to the global appearance of retrieved images, the dynamic one leverages the semantical similarity as a constraint for determining the matches. Accordingly, we propose a semantic-constrained retrieval framework by incorporating the dynamic match kernel, which focuses on matched patches between relevant images and filters out the ones for irrelevant pairs. Furthermore, we demonstrate that the proposed kernel complements recent methods such as Hamming embedding, multiple assignment, local descriptors aggregation and graph-based re-ranking, while it outperforms the static one under various settings on off-the-shelf evaluation metrics. We also propose to evaluate the matched patches both quantitatively and qualitatively. Extensive experiments on five benchmark datasets and large-scale distractors validate the merits of the proposed method against the state-of-the-art methods for image retrieval.**

***Index Terms*—Content based image retrieval, semantic-constrained framework, deep representation, dynamic match kernel**

I. Introduction

Recent years have witnessed significant advances in content based image retrieval (CBIR) [1], [2] with numerous applications. The goal of CBIR is to efficiently find the most relevant images of the given query from a huge amount of candidate corpus [3]. Different lines of existing retrieval frameworks calculate their search criteria with different image representing and indexing schemes. For representing the query and candidate images, both local features which are robust to depict low-level image contents, and global attributes reflecting semantical meanings, are independently well exploited. For instance, the state-of-the-art bag-of-words (BoW) model [4], [5] uses local descriptors to encode image regions of interest, *e.g.*, SIFT [6] and color names [7]. Recently, visual features derived from a convolutional neural network (CNN) are leveraged to enhance the discriminative capacity of the retrieval system [8], [9], where off-the-shelf CNN features are extracted from pre-trained models and used as a generic image representation to tackle image retrieval problems. The Hamming embedding (HE) based algorithms construct an inverted index file with a codebook quantizing the local descriptors into visual words, and images are matched using a weighted similarity function [10], [11] constrained by a given threshold. In addition, various post-processing methods [12]–[15] have been developed to refine the relevance of retrieved images.

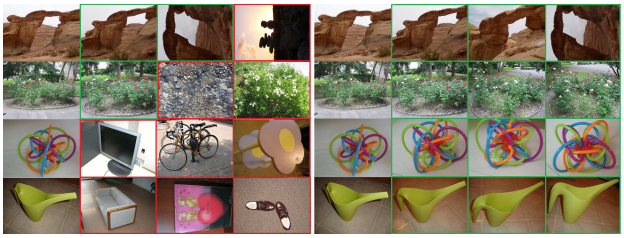


Fig. 1. Comparison of retrieved results using a static match kernel (**left**) and the proposed dynamic match kernel (**right**), respectively. Images in the first column of each part are query images. Relevant and irrelevant results in ground-truth are marked with green and red boxes, respectively. In each group, images in the first two rows are from the Holidays dataset [4], while the last two rows are from the UKBench database [16].

Given query and candidate images, the traditional local-based CBIR framework first detects local interest patches of each image. Then, it discovers all matched pairs by determining whether the distance between two patches is less than a given and fixed threshold [4]. The similarity score between the query and each candidate image is calculated as the quantity of matched components, followed by ranking the candidate images of this query accordingly [17]. Therefore, determining the number of matches between a pair of images is crucial for an effective retrieval system, which can be significantly influenced by the value of the selected threshold.

However, images of the same object or scene which should be considered to be similar may have variations due to various factors, *e.g.*, different illuminations or views. Also, since the traditional BoW model does not consider the spatial structure of local patches and loses information during quantization [18], non-relevant images might share many local descriptors *e.g.*, sharing blue sky by birds and planes, which may lead to false positive matches [19]. Therefore, traditional local-based static matching methods with a fixed threshold can hardly be optimal for various applications [4]. Moreover,

Manuscript received August 09, 2017; revised January 06, 2018 and May 10, 2018; accepted May 26, 2018.

J. Yang, J. Liang, H. Shen and K. Wang are with School of Computer Science and Control Engineering, Nankai University, Tianjin 300350, China (E-mail: yangjufeng@nankai.edu.cn; liang27jie@163.com; jhonjoe.c@gmail.com; wangk@nankai.edu.cn).

P.L. Rosin is with School of Computer Science and Informatics, Cardiff University, Wales, UK (E-mail: Paul.Rosin@cs.cf.ac.uk).

M.-H. Yang is with School of Engineering, University of California, Merced, CA 95343, USA (E-mail: mhyang@ucmerced.edu).

the retrieval framework should not only search for candidates sharing similar local contents but also encourage a fusion of constraint on their semantic similarities.

Deep CNNs provide discriminative features which are widely used in the vision community [20], [21]. The features from a high-level layer of a CNN model is effective at capturing a compact and holistic representation of an image. Inspired by the independent successes of local-based matching schemes and deep representations using CNNs, in this paper we propose a semantic-constrained retrieval framework to merge the advantages of both modules, which is expected to explore the shared similarity structures of both local and global representations. Specifically, we first calculate the semantic distance between two samples via high-level layers of a CNN model [22], together with the Hamming distance using a local descriptor. Then, we conduct an adaptive transformation on the global semantic distance to combine both cues of low-level image contents and semantical meanings. Consequently, we construct a dynamic match kernel for each query image to detect the matched candidates, which focuses on the positive matches and filters out the negative ones. Fig. 1 shows several examples of retrieved images based on the static match kernel (left) and the proposed dynamic one (right).

The contributions of this work are summarized as follows. First, rather than fixing the threshold when detecting the matched patches, we propose to calculate an adaptive threshold for each image pair according to the global similarity derived from deep CNN representations. For each query, the dynamic match kernel incorporates a relative similarity (reflected by semantic distance) among all candidates, which can be effectively measured by off-the-shelf deep CNN models. It provides a preference on allowing more local matches for relevant candidates, while rejecting most matches for irrelevant ones. Then, based on the dynamic match kernel, we propose a semantic-constrained retrieval framework which leverages both the local features describing low-level image contents and the global similarities reflecting semantic meanings. Extensive experiments on five benchmark datasets, *i.e.*, Holidays [4], UKBench [16], Paris6K [23], Oxford5K [5] and DupImages [24], show that the proposed dynamic match kernel outperforms the state-of-the-art methods with static ones. We also conduct experiments on large-scale distractors which combine the aforementioned datasets with 1 million [25] or 100 thousand [5] images and validate the generality of the proposed method.

II. Related Work and Problem Context

To put this work in context, we review the methods most relevant to the proposed algorithm regarding match kernels based on local descriptors [26], [27], as well as retrieval approaches using deep features [8], [28]. We also review several hybrid methods in this section.

A. Static Match Kernels with Local Descriptors

Numerous image retrieval methods based on local descriptors have been proposed [29], [30]. Image retrieval methods typically contain four components including feature extraction, quantization, indexing, and ranking [31]–[35], where most works concentrate on the improvement of feature extraction and the indexing scheme.

In particular, Niestér and Stewénius [16] use a visual vocabulary tree to hierarchically quantize SIFT features [6] into visual words. A local descriptor is assigned to its nearest k visual word, and the corresponding term frequency with image label is stored in the entry [31]. The hierarchical construction of the visual vocabulary tree facilitates storing a large amount of visual words and efficient search. However, detailed information of local features is not retained since choosing k is a compromise between efficiency and the quality of the descriptors [4]. To handle this problem, Jégou *et al.* [4] propose the Hamming Embedding method which uses a random matrix to encode the position of each descriptor within the Voronoi cells. Specifically, local descriptors are projected onto another space with a random matrix and binarized by the mean value learned from a training dataset. The similarity between a query and each candidate image in the database is computed by counting the matched local patches of both images weighted with the TF-IDF frequency. To determine whether the matching exists between a pair of patches, a static match kernel is employed which returns true if the distance between two patches is less than a given and fixed threshold.

Recently, Tolias *et al.* [36] aggregate local descriptors which are assigned to the same visual word into a single vector, and binarize it using the Hamming embedding scheme [4]. Here, aggregation denotes that all local descriptors assigned to the same visual word are averaged. Experimental results show that aggregation is critical in image retrieval as it encodes local descriptors effectively and removes noise [36]. Both texture and color cues have been used for image retrieval [18]. Local regions where both texture and color cues are sufficiently close are considered as a true match by incorporating two Kronecker delta function with SIFT and color name (CN) [7] descriptors.

B. Image Retrieval with Deep Covolutional Features

In recent years, several methods have exploited deep CNNs for image retrieval [37]–[40], thanks to their excellent property of capturing semantics and forming discriminative high-level representations which are robust to natural variations. Babenko *et al.* [41] extensively evaluate the performance of deep features. A descriptor of each image is extracted using a CNN model with fine-tuning and compressed using principal component analysis. Experimental results show that the neural codes outperform numerous state-of-the-art hand-crafted features for image retrieval. Gong *et al.* [42] concatenate the activations of a fully-connected layer with the vector of locally aggregated descriptors (VLAD) coding scheme applied to local patch responses at finer scales. Retrieval is performed using the Euclidean distance of the feature vectors. In contrast, Razavian *et al.* [43] extract activations from different resolutions and positions and find the minimum distance match between each query sub-patch and reference image sub-patch. Retrieved results are ranked by the average distance of the query sub-patch matches for each database image. Paulin *et al.* [44] detect regions by the Hessian-Affine detector [45].

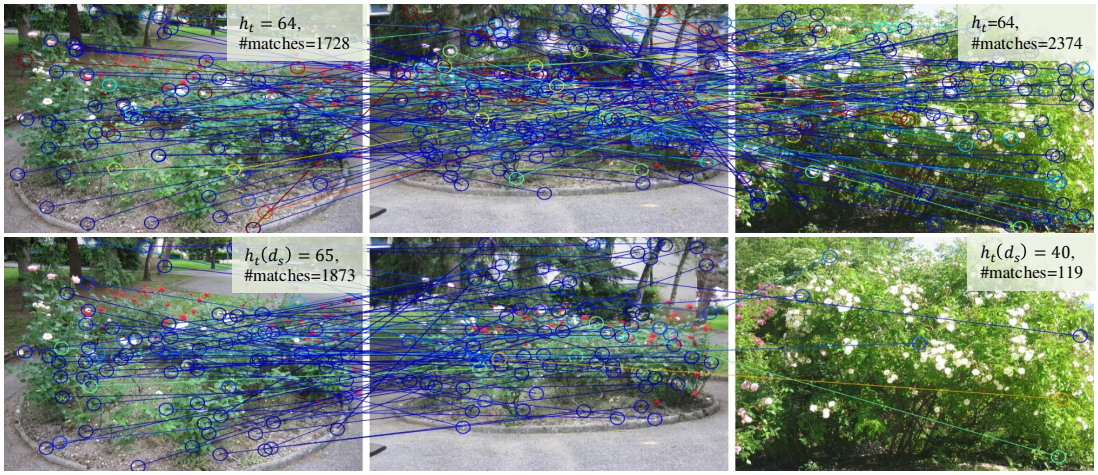


Fig. 2. Visualization of matched patches with both static **(top)** and dynamic **(bottom)** match kernels. In each row, a query image is shown in the middle while the relevant and irrelevant images are shown on its left- and right-hand sides, respectively. We use the default threshold $h_t = 64$ for the static match kernel, and compute $h_t(d_s)$ for the dynamic match kernel using Eq. 5. Here, $h_t(d_s)$ of relevant and irrelevant images equals 65 and 40, respectively. "#matches" denotes the number of feature matches for each candidate image. Circles of the same color indicate that corresponding local features are assigned to a same visual word. For presentation clarity, the number of feature matches for each image is reduced to one-fiftieth of the original.

Deep features are extracted from the affine normalized regions using convolutional kernel networks [46], and aggregated with the VLAD scheme. Recently, Babenko and Lempitsky [47] evaluated aggregation approaches for the output of the last layer of a deep network for image retrieval. They found the sum pooling method to work best.

As discussed, most recent methods use deep features for image retrieval in two ways. On one hand, a large number of image patches are fed into the CNN model for feature extraction [48] followed by the traditional indexing framework to finish the retrieval. Alternatively, the global features are extracted [49] in a single pass using pre-trained or fine-tuned CNN models followed by the approximate nearest neighbor method to generate the ranking result. Different from the previous works, we employ deep features to estimate the global semantic distance between images and use the adaptive threshold to construct dynamic match kernels. The dynamic threshold for each image pair can be considered as a semantic prior for the prediction of whether the two images are relevant or not.

C. Image Retrieval with Hybrid CNN-SIFT Features

There are several methods using both CNN features and SIFT features. Zhang *et al.* [50] propose a semantic-aware co-indexing scheme to fuse two cues into the inverted indexes: the SIFT features for delineating low-level image contents and the deep CNN features for revealing image semantic similarity. They use the semantic attributes to enrich the discriminative descriptors by inserting semantically similar images into the initial inverted index set built with SIFT features. Recently, Zhou *et al.* [3] construct an image retrieval scheme which improves the indexing ability of both a SIFT feature based module and a CNN [20] feature based module. They define separate codebooks for the two modules, and propose a collaborative index embedding algorithm in which two images should become more similar in one feature space if they are neighbors in the other feature space. This is achieved by modifying the positions of images in the feature space, alternating optimization in the two spaces. After enhancing the indexing ability of both high-level and low-level features, the embedded CNN index is used to generate the result of the retrieval. Generally, these methods preserve the indexing schemes of both SIFT and CNN features to enhance each other iteratively, which lead to a heavy burden for calculation. In this paper, we use the CNN features to calculate the dynamic threshold between image pairs. Meanwhile, we only employ the SIFT feature based indexing scheme which is constrained with the dynamic threshold generated from deep features. As a result, the proposed method is still efficient for retrieval although we consider both SIFT and CNN features, which is also validated in the experiments.

III. Semantic-Constrained Image Retrieval

Given a query image $\boldsymbol{I}_q$ and a database $\boldsymbol{D}$, the goal of content-based image retrieval (CBIR) is to estimate a ranking of candidate images based on their visual similarity with the query. We denote the candidate images as either relevant or irrelevant to the query image in the remainder. In this section, we first briefly review the traditional retrieval method followed by illustrating the proposed dynamic match kernel and the semantic-constrained retrieval framework.

A. Baseline Framework for Image Retrieval

1) BoW model based method: For determining the match of each candidate image $\boldsymbol{I}_c$, a BoW system first detects the local interest regions for both $\boldsymbol{I}_q$ and $\boldsymbol{I}_c$, *i.e.*, $\{\boldsymbol{x}_i\}_{i=1}^{n_q} \in \boldsymbol{I}_q$ and $\{\boldsymbol{y}_j\}_{j=1}^{n_c} \in \boldsymbol{I}_c$, where n_q and n_c denote the number of patches for the query and candidate image, respectively. Then, the system represents each patch using the SIFT descriptor, followed by vector-quantization, *e.g.*, Hamming code [4], using a large-scale visual vocabulary which aims to generate

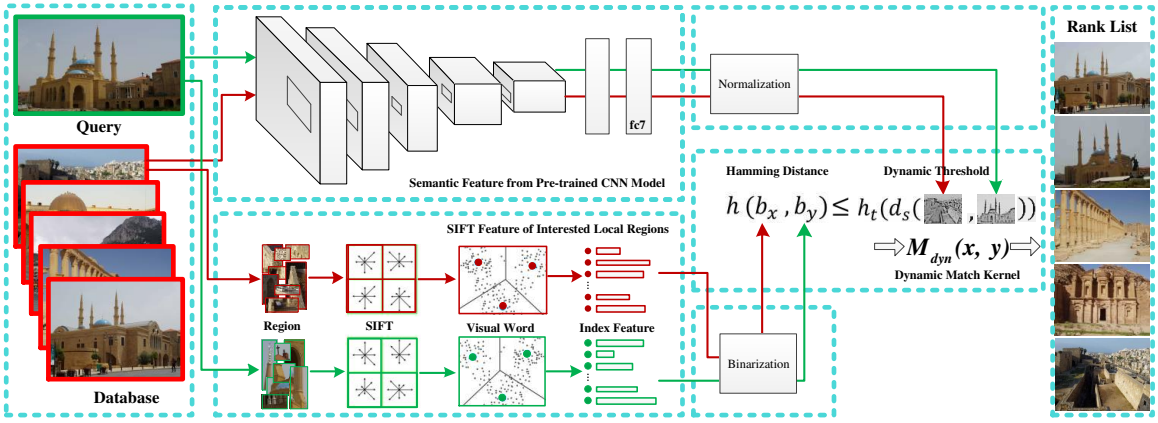


Fig. 3. Main steps of the proposed semantic-constrained image retrieval algorithm. Different from prior work based on local descriptors with static match kernels, we calculate an adaptive threshold $h_t(d_s)$ using a pre-trained CNN to construct the dynamic match kernel. The proposed dynamic match kernel function is shown in Eq. 6 where $h(\boldsymbol{b}_x, \boldsymbol{b}_y)$ and $h_t(d_s)$ are computed with local invariant features and deep features, respectively.

a distinctive signature of each local region. After that, it calculates the ranking of candidate images using a similarity metric on the obtained signatures. Specifically, a similarity score $S(\boldsymbol{I}_q, \boldsymbol{I}_c)$ is computed as the quantity of matched patches between $\boldsymbol{I}_q$ and $\boldsymbol{I}_c$, *i.e.*,

$$S(\boldsymbol{I}_q, \boldsymbol{I}_c) = \sum_{\boldsymbol{x}_i \in \boldsymbol{I}_q} \sum_{\boldsymbol{y}_i \in \boldsymbol{I}_c} M_{sta}(\boldsymbol{x}_i, \boldsymbol{y}_i) \times f_{\text{TF-IDF}}(\boldsymbol{x}_i, \boldsymbol{y}_j), \quad (1)$$

where $f_{\text{TF-IDF}}(\boldsymbol{x}_i, \boldsymbol{y}_j)$ is the TF-IDF weights of $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$. Here, h_t denotes a fixed threshold which is given as a prior and $M_{sta}()$ denotes the static match kernel defined as:

$$M_{sta}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \delta_{v(\boldsymbol{x}), v(\boldsymbol{y})}, & h(\boldsymbol{b}_x, \boldsymbol{b}_y) \leqslant h_t, \\ 0, & \text{otherwise}, \end{cases} \quad (2)$$

where $v(\cdot)$ is a quantization function for local descriptors and δ is the Kronecker delta function. In addition, $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ denote the binarized vectors of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. The Hamming distance $h(\boldsymbol{b}_x, \boldsymbol{b}_y)$ is computed between the binarized features $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$, and h_t is a given threshold so that $0 \leqslant h_t \leqslant l_H$ where l_H denotes the length of Hamming code of a local descriptor in the inverted table. Finally, $S(\boldsymbol{I}_q, \boldsymbol{I}_c)$ is used to rank the candidate images for $\boldsymbol{I}_q$ [17].

2) Semantic Representation: To measure the semantic distance between query and candidate images, we adopt a compact and holistic representation derived from a deep CNN model [20], which is composed of five sequential convolutional layers followed by three fully connected layers. The employed CNN model is pre-trained on the ImageNet dataset, which takes color images as input and outputs a feature vector $\boldsymbol{z} \in \mathbb{R}^{4096}$ from the fully connected layers. Then, we translate $\boldsymbol{z}$ into a unit vector $\widehat{\boldsymbol{z}}$ via the ℓ_2-normalization, *i.e.*,

$$\widehat{z}_i = \frac{z_i}{\sqrt{\sum_{i=1}^{4096} z_i^2}}, \quad (3)$$

where each z_i denotes the i-th entry of $\boldsymbol{z}$.

B. Dynamic Match Kernel

We construct the dynamic match kernel in this subsection. Eq. 1 demonstrates that the match kernel plays a pivotal role in measuring image similarity. Given a set of local descriptors for both query I_q and a retrieved image I_c, *i.e.*, $\{x_i\}_{i=1}^{n_q} \in I_q$ and $\{y_j\}_{j=1}^{n_c} \in I_c$, the traditional CBIR framework matches image patches by determining whether a distance $d(x, y)$ is less than a given and fixed threshold h_t [4]. However, a static match kernel with h_t fixed can hardly be optimal for different applications with different illuminations or views since the given threshold is independent of the holistic relationship between local patches.

To tackle this problem, we propose to construct a dynamic match kernel with an adaptive threshold, which is calculated based on the semantic distance between $\boldsymbol{I}_q$ and each $\boldsymbol{I}_c$. Specifically, for a given query $\boldsymbol{I}_q$ and each candidate image $\boldsymbol{I}_c$, we extract their deep representations from CNN model as stated in Sec. III-A2, *i.e.*, $\boldsymbol{z}_q$ and $\boldsymbol{z}_c$, respectively. We then apply ℓ_2-normalization using Eq. 3 to generate $\widehat{\boldsymbol{z}}_q$ and $\widehat{\boldsymbol{z}}_c$, followed by calculating the semantic distance d_s as follows:

$$d_s = \|\widehat{\boldsymbol{z}}_q - \widehat{\boldsymbol{z}}_c\|_2^2. \quad (4)$$

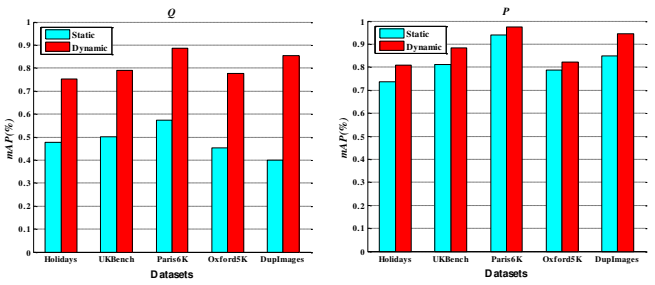
Then, to *bridge the gap* between the different domains, *i.e.*,

Algorithm 1 : Semantic-Constrained Image Retrieval

Input: The query $\boldsymbol{I}_q$, the database $\boldsymbol{D}$ with N candidates.

1: Detect the interested local regions for $\boldsymbol{I}_q \cup \boldsymbol{D}$;
2: Extract the semantic representations for $\boldsymbol{I}_q \cup \boldsymbol{D}$;
3: Normalize the representations using Eq. 3;
4: **for** $i = 1 : N$ **do**
5: Generate the Hamming distance between patches of $\boldsymbol{I}_q$ and $\boldsymbol{I}_c^i$ as in Section III-A1;
6: Calculate the semantic distance between $\boldsymbol{I}_q$ and $\boldsymbol{I}_c^i$;
7: Compute the adaptive threshold by transforming the semantic distance using Eq. 5;
8: Construct the dynamic match kernel $M_{dyn}(\cdot, \cdot)$ using Eq. 6;
9: Calculate the similarity score between $\boldsymbol{I}_q$ and $\boldsymbol{I}_c^i$ using Eq. 7;
10: **end for**
11: Compute the ranking order for the query $\boldsymbol{I}_q$ using Eq. 8.

Output: Ranking order $\boldsymbol{r}_q$ of the candidates.



(a) (b)

Fig. 4. Comparison between the static and dynamic match kernels in terms of quantity and quality scores, where Q and P measures the quantity and quality of positive matches, respectively.

the Hamming and semantic distances, we design the following transformation system with both linear and non-linear operations to calculate the adaptive threshold $h_t(d_s)$:

$$h_t(d_s) = \lambda \times l_H \times \mathrm{e}^{-\frac{d_s}{2}}, \tag{5}$$

where l_H is the length of Hamming code of a local descriptor in the inverted table, and $\lambda > 0$ is a scaling parameter of the dynamic threshold. We apply the non-linear exponential operation on $-\frac{d_s}{2}$ to map the Euclidean distance into the exponent space with lower growth rate, since we have $-\frac{d_s}{2} \leq 0$. We evaluate the parameter λ through experiments to linearly control the scale of $h_t(d_s)$, of which the robustness is validated in Section IV-B. Accordingly, for interest patches $\boldsymbol{x} \in \boldsymbol{I}_q$ and $\boldsymbol{y} \in \boldsymbol{I}_c$, we propose the dynamic match kernel $M_{dyn}(\boldsymbol{x}, \boldsymbol{y})$ defined as:

$$M_{dyn}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} \delta_{v(\boldsymbol{x}),v(\boldsymbol{y})} \cdot f_s(h(\boldsymbol{b}_x, \boldsymbol{b}_y)), & h(\boldsymbol{b}_x, \boldsymbol{b}_y) \leqslant h_t(d_s), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where d_s denotes the semantic distance calculated by Eq. 4 and $h_t(d_s)$ denotes the adaptive threshold calculated by Eq. 5.

We denote the matches between relevant and irrelevant images to be positive and negative matches, respectively. The adaptive threshold calculated based on the holistic deep representations incorporates the relationship between the semantics of the candidate image and the query. For each pair of images, a smaller d_s denotes that the two images are more semantically relevant, which leads to a larger value of the adaptive threshold $h_t(d_s)$ in Eq. 5. Consequently, the dynamic match kernel enlarges the quantitative gap between positive and negative matches, which leads to a better performance than the static match kernel.

Fig. 2 shows how the dynamic match kernels affect the number of matched local regions against the static one. Given the query image in the middle of each row, we show the number of matched patches using both kernels on relevant (*left*) and irrelevant (*right*) images. For the static match kernel, we use the default value $h_t = 64$ in prior work [4]. Note we verify the effectiveness of the static threshold in Section IV-B. We can see from the first row of Fig. 2 that more negative matches are found than positive ones with the static match kernel. In contrast, according to the semantic variation detected by deep representations, we compute different thresholds $h_t(d_s)$ for the relevant and irrelevant images using Eq. 5, *i.e.*, $h_t(d_s) = 65$ and $h_t(d_s) = 40$, respectively. As a result, the number of non-zero matches for $S(\boldsymbol{I}_q, \boldsymbol{I}_c)$ tends to increase if $\boldsymbol{I}_c$ is relevant to $\boldsymbol{I}_q$ and decrease otherwise. Meanwhile, since the majority of the database is irrelevant to the query, it simplifies the calculation due to the significant reduction of the feature matches.

C. Semantic-Constrained Image Retrieval Algorithm

Fig. 3 and Algorithm 1 depict the main steps of the proposed semantic-constrained retrieval algorithm. For a retrieval task, the SIFT features of local regions are extracted first from both query and database images, then the Hamming distance $h(\boldsymbol{b}_x, \boldsymbol{b}_y)$ is calculated as described in Section III-A1. Simultaneously, we train a neural network and the transformation system in Eq. 5 to construct an adaptive threshold $h_t(d_s)$ for each query. We use the CNN model [20] trained on the ImageNet dataset to extract semantic features. The semantic distance d_s between two images is the squared Euclidean distance after ℓ_2-normalization. For each pair of images, a smaller d_s denotes that the two images are more semantically relevant, which leads to a larger value of the adaptive threshold $h_t(d_s)$ in Eq. 5. We then construct the dynamic match kernel $M_{dyn}(\cdot, \cdot)$ using Eq. 6 followed by calculating the similarity score via:

$$S(\boldsymbol{I}_q, \boldsymbol{I}_c) = \sum_{\boldsymbol{x}_i \in \boldsymbol{I}_q} \sum_{\boldsymbol{y}_i \in \boldsymbol{I}_c} M_{dyn}(\boldsymbol{x}_i, \boldsymbol{y}_i) \times f_{\text{TF-IDF}}(\boldsymbol{x}_i, \boldsymbol{y}_j). \tag{7}$$

Here, $M_{dyn}(\cdot, \cdot)$ integrates global and pairwise semantic relations between the query and candidate images, which is considered as a constraint on the similarity estimation. Finally, for each $\boldsymbol{I}_c^i$ in the database $\boldsymbol{D}$, we calculate the ranking order $\boldsymbol{r}_q$ by

$$r_i = \text{SORT}(\boldsymbol{I}_c^i | \boldsymbol{D}), \tag{8}$$

where $\text{SORT}(\boldsymbol{a}|\boldsymbol{A})$ denotes a function which returns the ranked index of $\boldsymbol{a}$ against $\boldsymbol{A}$.

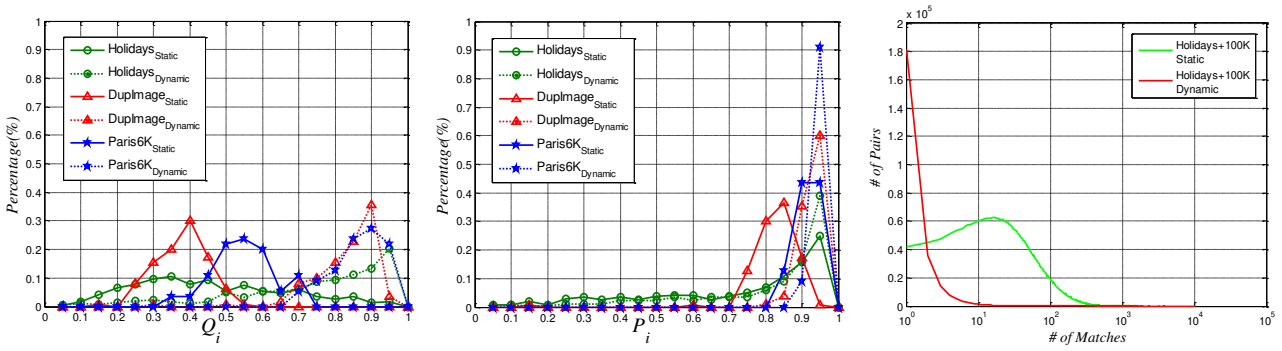
D. Evaluation Metrics on Positive Matches

In this work, we use the state-of-the-art match function [36] in Eq. 2, *i.e.*, the static match kernel, as the baseline. For validating the effectiveness of the proposed dynamic match kernel, which induces the large gap between the number/score of the positive and negative matches, we propose to evaluate the corresponding feature matches both quantitatively and qualitatively.

1) Average Quantity of Feature Matches: Given m query images $\{\boldsymbol{I}_q^i\}_{i=1}^m$, we define the average score as $Q = \frac{1}{m}\sum_{i=1}^N Q_i$, where Q_i is the quantity score of the matched patches for $\boldsymbol{I}_q^i$ calculated by

$$Q_i = \frac{\sum_{j=1}^{K_i} n_{i,j}^+}{\sum_{j=1}^{K_i} n_{i,j}^+ + \sum_{j=1}^{K_i} n_{i,j}^-}. \tag{9}$$

Here, K_i is the number of relevant images of $\boldsymbol{I}_q^i$ in the database, and we select the top K_i irrelevant images from the rank list to be negative samples. For easy illustration, we assign temporary indexes ranging from 1 to K_i for both the



(a) (b) (c)

Fig. 5. Comparison of quantity **(a)** and quality **(b)** of positive matches when the static match kernel **(solid lines)** and dynamic match kernel **(dotted lines)** are applied on three benchmarks (The curves of other two datasets are similar). Q_i and P_i are defined in Eq. 9 and Eq. 10, respectively. The vertical axes in both figures represent the percentages of feature matches falling into the corresponding quantity and quality bins. **(c)** shows the statistics of the non-zero matches between image pairs in Holidays dataset with 100K distractors. The horizontal axis indicates the number of matches, and the vertical axis represents the number of image pairs which have the corresponding match number. More details can be found in Section III-D.

relevant images and the selected irrelevant images. Therefore, $n^+_{i,j}$ denotes the number of matches in the j-th relevant image for the query image $\boldsymbol{I}^i_q$, and $n^-_{i,j}$ is the number of matches in the j-th irrelevant image. As a result, a larger Q represents more positive matches determined by Eq. 1 and Eq. 7, which improves the performance of image retrieval.

2) Average Quality of Feature Matches: Similarly, we define the average quality score as $P = \frac{1}{m}\sum_{i=1}^{m} P_i$, where each P_i for a query $\boldsymbol{I}^i_q$ is calculated by

$$P_i = \frac{\sum_{j=1}^{K_i} n^+_{i,j}\overline{M}^+_{i,j}}{\sum_{j=1}^{K_i} n^+_{i,j}\overline{M}^+_{i,j} + \sum_{j=1}^{K_i} n^-_{i,j}\overline{M}^-_{i,j}}, \qquad (10)$$

where $\overline{M}^+_{i,j}$ is the mean matching score of the matches in the j-th relevant image. and $\overline{M}^-_{i,j}$ is the mean match score of the matches in the j-th irrelevant image. A larger P implies that more positive matches occurred with higher match scores.

We note that the proposed two evaluation metrics are calculated directly with the number and quality of positive and negative matches, which are comprehensive and critical for the performance of a retrieval system. The proportion of positive matches among the whole set reflects the accuracy of a matching system. Meanwhile, higher Q and P with smaller denominators indicates both better efficiency and effectiveness of the retrieval framework.

Fig. 4(a) shows that on all five datasets, the proportion of positive matches obtained with the proposed dynamic match kernel increases significantly in most cases compared to that obtained by the static match kernel, which in turn improves the similarity scores of relevant images (See Section IV). Meanwhile, Fig. 4(b) shows that the average quality score of the feature matches is also increased when the proposed dynamic match kernel is used. These results demonstrate the effectiveness of the proposed match kernel both quantitatively and qualitatively.

We further examine the distributions of Q_i and P_i on all benchmark datasets and analyze the contribution of the dynamic match kernel. Fig. 5(a) and 5(b) demonstrate that curves for both Q_i and P_i with the dynamic match kernel shift to the right, which means more positive matches are included while most negative ones are excluded in our system. Note that, as defined in Eq. 9 and Eq. 10, Q_i and P_i represent the proportion of positive matches and scores, respectively. Hence, with other well-designed modules (*e.g.*, ranking), a retrieval method using the proposed dynamic match kernel will be further enhanced with more positive matches and fewer negative ones.

TABLE I
COMPARISON OF RETRIEVAL PERFORMANCE USING DIFFERENT SEMANTIC REPRESENTATIONS FROM DIFFERENT FULLY-CONNECTED LAYERS IN THE CNN. THREE KINDS OF DEEP FEATURES ARE EMPLOYED TO CALCULATE THE ADAPTIVE THRESHOLDS.

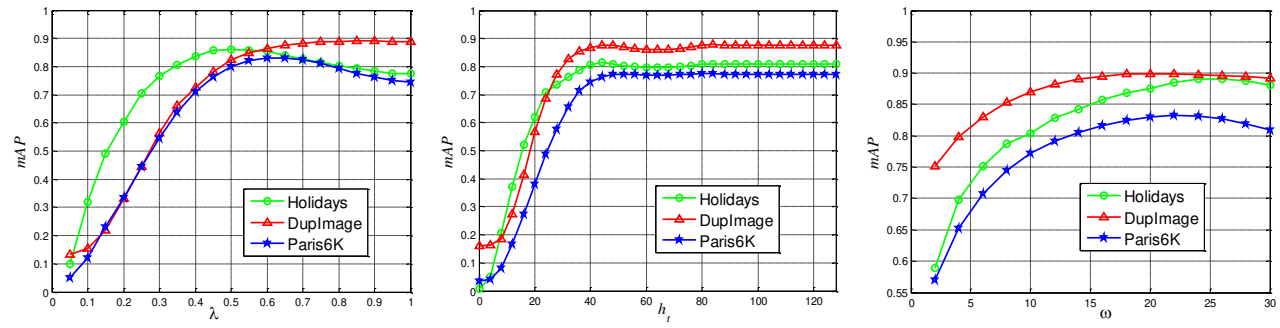
Layers	Holidays	UKBench	Paris6K	Oxford5K	DupImages
	mAP (%)	N-S	mAP (%)	mAP (%)	mAP (%)
fc_6	87.78	**3.82**	82.94	80.78	88.85
fc_7	**87.92**	**3.82**	**84.92**	**83.05**	**89.43**
fc_8	81.03	3.43	72.55	70.80	82.06

In Fig. 5(c), we present the histogram of matches between image pairs with or without applying the dynamic match kernel. The experiments are conducted on the combination of the Holidays and 100K distractors dataset from Flickr website. The curves indicate that the overall number of matches reduces drastically by applying the proposed dynamic match kernel, which makes the calculation of similarity more efficient. Specifically, most image pairs have less than 5 matches according to the dynamic match kernel, since most images in the database are irrelevant to the query, *e.g.*, in Holidays+100K dataset, 2 images are relevant to each query and 101488 images are irrelevant. As a contrast, there are about half of the image pairs with 10 to 1000 feature matches when static match kernel is applied. Therefore, it turns out that the proposed dynamic threshold filters most of the negative matches.

IV. Experimental Results and Analysis

A. Setup

1) Datasets and Evaluation Metrics: We evaluate the proposed algorithm against the state-of-the-art image retrieval



(a) (b) (c)

Fig. 6. Parameter analysis on Holidays, Paris6K and DupImage datasets. **(a)** Effect of the parameter λ which linearly scales the dynamic threshold in Eq. 5. We set $\lambda = 0.6$ since it leads to the best performance. **(b)** The mAP of the static match kernel against the fixed threshold h_t in Eq. 2. The mAP tends to be stable when $h_t > 40$, where detailed analysis of λ and h_t can be found in Section IV-B. **(c)** Influence of the parameter ω in Section IV-F which controls the intensity of the exponent when the dynamic selective match function [51] is applied instead of the dynamic threshold. We set $\omega = 22$ in this paper.

methods on the following five benchmark datasets: Holidays [4], UKBench [16], Paris6K [23], Oxford5K [5] and DupImages [24]. The performance on all datasets can be measured by the mean average precision (mAP) [31] expressed as percentages, where the UKBench dataset can also be evaluated using the N-S score (maximum 4) [52].

We also combine three benchmark datasets with large sets of distractors to evaluate the generality ability of the proposed approach. Following the previous literature [1], [53], for the holidays dataset, we merged it with MIR Flickr 1M (1 million) images [25], so that the size of the final dataset is $1,001,491$. For Paris6K and Oxford5K, we added 100K (100 thousands) distractors as in [5]. The features of the distractors are extracted in the same manner as for the benchmark datasets which will be inserted into the inverted file system.

2) Local Features: Unless stated otherwise, we first use the modified Hessian-Affine detector similar to the work proposed by Perdoch *et al.* [54] with default parameters to detect regions of interest. Then, SIFT features are collected from the detected regions. As the root-SIFT descriptors have been demonstrated to perform well, we use component-wise square-root and ℓ_2-normalization in the experiments.

3) Vocabularies: The Approximate K-Means method [5] is used to generate visual words. The vocabulary size is 65K in all datasets except for DupImages in which the vocabulary size is set to 4K [18]. For the Holidays, Oxford5K and Paris6K datasets, the vocabularies are trained using an independent dataset from Flickr as was done in prior work [51].

4) Multiple Assignment: We employ the multiple assignment (MA) scheme [4] in which the 5 nearest neighbors of a query descriptor are used.

5) Aggregation: We employ the aggregation operation proposed by Tolias *et al.* [36], where local features with a same visual word are aggregated into a single descriptor. For an arbitrary image, if the set of local features $\{\boldsymbol{x}_i\}_{i=1}^{n_j}$ are all assigned to the visual word $\boldsymbol{v}_j$, then we simply aggregate those local features into a single feature $\bar{\boldsymbol{x}} = \frac{1}{n_j}\Sigma_{i=1}^{n_j}\boldsymbol{x}_i$.

6) Hamming Embedding: We employ HE [4] to compress local features into binary codes. Concretely, considering an arbitrary local feature $\boldsymbol{x}$ and its corresponding projection $\boldsymbol{z}$, if $\boldsymbol{x}$ is assigned to the visual word $\boldsymbol{v}_j$, then each entry of the corresponding Hamming code $\boldsymbol{b}$ satisfies that $\boldsymbol{b}_i = 1$ if $\boldsymbol{z}_i > \tau_{j,i}$ and 0 otherwise. Here, the parameter $\tau_{j,i}$ denotes the i-th mean value of the visual word $\boldsymbol{v}_j$, which is learned from the training dataset.

7) Deep Features: The AlexNet [20] pre-trained on ImageNet [55] is employed to extract deep semantic features. Features are extracted from the fully-connected layer without aggregation or additional transformations, and the dimension of the deep features is 4096. We apply ℓ_2-normalization on deep features as empirically it performs well for image retrieval.

8) Performance of Comparative Methods: For the tables in Sections IV-D, IV-E and IV-F, the results are derived from either the original papers or our evaluation with released codes. For the latter, we employ the same baseline framework as used in the proposed method unless stated otherwise, and replace the corresponding operation for fair comparison. We employ the same architecture and parameters as reported in the original paper.

B. Impact of Parameters

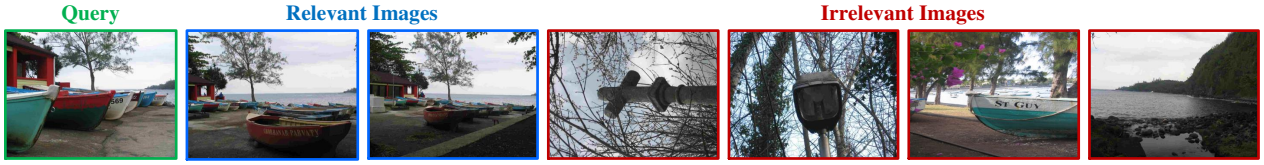
1) Parameter λ: We first evaluate the effect of parameter λ in Eq. 5. The parameter λ induces a linear scaling of the dynamic threshold $h_t(d_s)$, by which the semantic and the Hamming distance are projected into a common subspace. As shown in Fig. 6(a), the proposed method performs well within a wide range of λ values on three benchmark datasets, which indicates that the transformation process is robust. We set $\lambda = 0.6$ in all the experiments for performance evaluation against the state-of-the-art methods.

2) Parameter h_t: We then evaluate the effect of parameter h_t for static match kernel in Eq. 2. Fig. 6(b) shows that the performance of the static match kernel rises rapidly when h_t increases from 0 to 40. Once it is beyond 40, the performance becomes stable. Considering both the consistency with previous work [4], [18], [36] and the effectiveness of our experiments, we set $h_t = 64$ for the static match kernel.

3) Different layers in CNN: Deep features are used to calculate the semantic distance followed by constructing the dynamic match kernel for each query. To select a better

TABLE II
COMPARISON BETWEEN THE PROPOSED DYNAMIC MATCH KERNEL AND THE STATIC ONE ON FIVE BENCHMARK DATASETS WITH DIFFERENT COMBINATIONS OF COMPONENTS, *i.e.*, BAG OF WORDS (BOW), MULTIPLE ASSIGNMENT (MA, WITH 5 NEAREST NEIGHBORS), LOCAL DESCRIPTORS AGGREGATION (AGG), GRAPH-BASED RE-RANKING (RR). THE CHECKMARK DENOTES THE CORRESPONDING COMPONENT IS INCLUDED.

Kernel	BoW	MA	AGG	RR	Holidays	UKBench		Paris6K	Oxford5K	DupImages
					mAP (%)	N-S	mAP (%)	mAP (%)	mAP (%)	mAP (%)
Static	√				75.91	3.36	86.66	67.96	73.71	83.04
Dynamic	√				82.40 ↑	3.50 ↑	90.00 ↑	76.27 ↑	74.54 ↑	85.83 ↑
Static	√	√			73.44	3.39	87.62	68.75	76.15	72.97
Dynamic	√	√			84.04 ↑	3.59 ↑	92.22 ↑	79.31 ↑	77.87 ↑	82.36 ↑
Static	√		√		79.16	3.42	87.93	74.38	75.85	89.38
Dynamic	√		√		82.98 ↑	3.48 ↑	89.66 ↑	78.51 ↑	76.54 ↑	**89.43** ↑
Static	√	√	√		79.72	3.53	90.44	77.02	80.40	86.14
Dynamic	√	√	√		**87.92** ↑	**3.82** ↑	**96.97** ↑	**84.92** ↑	**83.05** ↑	88.98 ↑
Dynamic	√	√	√	√	**91.11**	**3.88**	**98.09**	**87.22**	**85.11**	**91.00**



Static	Rank	1	4	2	3	23	15
	Threshold	64	64	64	64	64	64
	#Matches	667	690	1487	1592	702	526
	Score	0.105	0.053	0.042	0.040	0.021	0.085
Dynamic	Rank	1	2	30	38	3	4
	Threshold	54	46	32	34	40	38
	#Matches	220	129	12	8	95	4
	Score	0.305	0.243	0.082	0.053	0.099	1.193

Fig. 7. Effectiveness of the semantic representations for calculating adaptive threshold for image retrieval. Images in the first row depict an example of retrieval on the Holidays dataset, including the query (with green box), relevant (blue) and irrelevant (red) images. The table reports quantitative results for each image, *i.e.*, the ranking, generated threshold, number of matches (#Matches) and mean match score (Score), when using **Static** and **Dynamic** match kernels.

representation for measuring the distance, we evaluate features extracted from different fully-connected layers in the CNN model. Table I demonstrates that the results using features from layer fc_7 are better than those using fc_6 or fc_8. These experimental results are consistent with the findings in prior work where features from layer fc_7 are shown to perform better than those using other fully-connected layers. We extract features from the fc_7 layer in the rest of the experiments.

C. Effectiveness of Dynamic Match Kernel

In this subsection, we demonstrate the significant influence of incorporating semantic representations for retrieval in Fig. 7, followed by validating the effectiveness of the proposed dynamic match kernel against the static one in Table II.

Fig. 7 visualizes an example when performing retrieval on the Holidays dataset, and compares the quantitative results with both static and dynamic thresholds. Note that the latter incorporates the semantic information into the indexing system. The figure shows the effectiveness of the dynamic threshold derived from the semantic representations against the static one in three ways.

First, the adaptive threshold calculated from the semantic distance using Eq. 5 reflects the visual similarity between the query and candidate images. The thresholds of all relevant images are larger than those of irrelevant images, which produces a positive bias on determining the matching relationships. Consequently, the proposed semantic-constrained retrieval framework not only detects the relevant candidates as the most closely related results, but provides a favorable rank on several negative candidates which are labeled as irrelevant but look semantically similar to the query. For example, the third irrelevant image with boat and trees is ranked as a top 3 candidate by the proposed method, while the static threshold based framework prefers the first irrelevant image due to the similar local patches such as the sky.

Second, by mapping the semantic and Hamming distances into a common subspace, the number of matches are consistent with the global similarity between two images, *i.e.*, relevant candidates have more local matches than others.

Third, as discussed in Section III-D, the adaptive thresholds for all candidates are smaller than the fixed ones, which eliminates most redundant matches and accelerates the indexing procedure, especially for irrelevant images.

We also calculate the Q and P of the selected query image as defined in Eq. 9 and Eq. 10, respectively. We have $Q = 0.306$ and $P = 0.458$ with static match kernel, while $Q = 0.779$ and $P = 0.874$ with the proposed dynamic match kernel. The dynamic match kernel produces better matching

TABLE III
COMPARISON TO STATE-OF-THE-ART METHODS WITHOUT POST-PROCESSING, WHICH ARE FOCUSED ON THE IMPROVEMENT OF THE REPRESENTATION OF IMAGE PATCHES AND THE INDEXING MODULE, RESPECTIVELY. "†" DENOTES THAT THE RESULT IS DERIVED FROM OUR EVALUATION AS DISCUSSED IN SECTION IV-A8. THE VALUES SHOWN IN BOLD CORRESPOND TO THE BEST SCORE PER DATASET.

Methods	Ours	Representation Based Methods				Indexing Based Methods							
		[51]	[52]	[18]	[56]	[53]	[57]	[58]	[59]	[60]	[61]	[62]	[63]
Holidays	87.92	82.2	79.6	84.0	80.9	75.8	78.7	82.1	**88.1**	75.8	81.3	83.9	73.2
UKBench	**3.82**	3.65†	3.60	3.71	3.60	3.50†	N/A	N/A	N/A	N/A	3.42	3.54	3.56
Paris6K	**84.92**	78.2	N/A	N/A	N/A	74.9	N/A	73.6	77.5	N/A	N/A	N/A	N/A
Oxford5K	**83.05**	81.7	N/A	N/A	68.7	74.2	77.0	78.0	80.4	67.7	61.5	64.7	59.0
DupImages	**89.43**	85.5†	87.1	87.6	N/A	82.3†	N/A	N/A	N/A	N/A	N/A	N/A	N/A

TABLE IV
COMPARISON TO STATE-OF-THE-ART METHODS WITH DIFFERENT POST-PROCESSING MODULES, *i.e.*, IMAGE LEVEL RE-RANKING (WHICH IS EMPLOYED IN THIS PAPER) OR OTHERS. "†" DENOTES THAT THE RESULT IS DERIVED BY OUR EVALUATION AS DISCUSSED IN SECTION IV-A8. "*" DENOTES THAT THE RESULT IS UNFAIR FOR COMPARISON AND WILL BE FURTHER INTERPRETED IN SECTION IV-D. THE VALUES SHOWN IN BOLD CORRESPOND TO THE BEST SCORE PER DATASET.

Methods	Ours	Graph-Based Re-Ranking					Others								
		[64]	[51]	[1]	[26]	[18]	[53]	[65]	[66]	[12]	[52]	[67]	[58]	[62]	[19]
Holidays	**91.11**	N/A	81.3	88.0	91.7*	85.8	75.8	N/A	N/A	89.2	85.2	88.3	80.1	84.8	78.0
UKBench	**3.88**	N/A	3.72†	3.84	N/A	3.85	3.63†	N/A	N/A	N/A	3.79	3.86	N/A	3.64	N/A
Parix6K	**87.22**	84.5	85.1	85.7†	N/A	N/A	82.4	76.5	80.5	N/A	N/A	84.9	85.5	N/A	N/A
Oxford5K	85.11	**87.7**	86.9	78.3†	74.1	N/A	84.9	80.9	82.7	73.7	N/A	83.3	85.0	68.5	77.3
DupImages	**91.00**	N/A	86.4†	85.6†	N/A	N/A	**84.1**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

capacity by incorporating the holistic similarity between images.

Table II validates the performance gain of the proposed dynamic match kernel over the static one on five benchmark datasets along with different components. For all kinds of combinations and benchmark datasets, the proposed dynamic match kernel improves the results of baseline methods which are based on static match kernels. In particular, on the Holidays and Paris6K datasets, the performance when combining the dynamic match kernel are improved by 8.20% and 7.90% respectively on top of the first three modules, *i.e.*, the bag of words, multiple assignment and aggregation. We also apply the graph-based re-ranking (RR) method [13] to refine the retrieval results, which further improves the performance of the proposed method.

D. Comparisons to State-of-the-Art Methods

We evaluate the proposed algorithm against the state-of-the-art methods in this subsection. For clear comparison, we categorize existing retrieval work into three technical approaches, *i.e.*, without or with post-processing, employing deep features. Each of them is further arranged based on the implementation details [22].

We first compare the proposed semantic-constrained retrieval framework against the state-of-the-art methods with no post-processing in Table III. Existing algorithms contribute to improvement in two different ways, *i.e.*, constructing more distinctive representations and calculating a more precise index. In this paper, we employ the baseline SIFT feature for image patches, and contribute by incorporating the deep semantic relationships during the indexing stage. On the UKBench, Paris6K, Oxford5K and DupImages datasets, we achieve the best performance. The proposed approach outperforms the second best method by 0.11 in terms of N-S score and $6.72\%, 1.35\%, 1.83\%$ in terms of mAP on these datasets, respectively. The method in [59] achieves better performance by around 0.2% on Holidays dataset but is not robust when applied to other datasets. The dynamic threshold calculated by Eq. 5 introduces a semantic cue into the retrieval system, where the matches to negative candidates are limited with a small threshold.

In addition, we evaluate the proposed algorithm with the graph-based re-ranking (RR) against the state-of-the-art results, all of which incorporate various post-processing schemes including RR and others, *e.g.*, query expansion [64], and spatial verification [36]. Table IV shows that the performance of the proposed algorithm with post-processing is further improved, and consistently better than the other methods with post-processing on the Holidays, UKBench, Paris6K and DupImages datasets. The kernelized SLEM [26] shows better performance on the Holidays dataset with the well designed NetVLAD [74], but the performance declines to 72.9% with AlexNet [20] feature (which is used in the proposed framework). The HGP method proposed in [64] outperforms us on the Oxford5K dataset, which uses complicated post-processing techniques and cannot easily be extended to other applications.

In Table V, we compare the proposed semantic-constrained retrieval framework against recent methods which employ deep neural networks. We categorize these approaches in two ways: some consider deep activations as global descriptors, while others combine multiple cues at the feature level or index level [18] for image retrieval. In the proposed method, deep features are used to construct a dynamic match kernel

TABLE V
COMPARISON TO STATE-OF-THE-ART METHODS EMPLOYING DEEP FEATURES IN THREE WAYS, *i.e.*, UTILIZING IT AS DEEP DISCRIMINATIVE REPRESENTATIONS, AND CONSTRUCTING A JOINT MODEL FOR RETRIEVAL USING FUSION SCHEMES. THE VALUE SHOWN IN BOLD CORRESPOND TO THE BEST SCORE PER DATASET.

Methods	Ours	Deep Representation								Fusion Scheme						
		[37]	[41]	[44]	[49]	[42]	[43]	[47]	[68]	[38]	[39]	[69]	[70]	[3]	[63]	[1]
Holidays	**91.11**	84.0	78.9	79.3	89.1	80.2	84.3	80.2	89.9	85.7	89.7	89.1	85.8	90.3	84.5	88.0
UKBench	3.88	N/A	3.55	3.76	N/A	N/A	N/A	3.65	3.89	3.76	N/A	3.88	3.53	**3.91**	N/A	3.84
Paris6K	**87.22**	69.4	N/A	N/A	87.1	N/A	79.5	N/A	N/A	81.2	85.3	N/A	N/A	N/A	N/A	85.7†
Oxford5K	**85.11**	64.9	55.7	56.5	83.1	N/A	68.0	65.7	N/A	N/A	84.4	83.5	N/A	N/A	67.5	78.3†
DupImages	**91.00**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	85.6†

TABLE VI
PERFORMANCE OF THE PROPOSED METHOD ON THREE DATASETS AND THEIR CORRESPONDING LARGE SCALE EXTENSIONS COMPARED TO OTHER STATE-OF-THE-ART METHODS. "†" DENOTES THAT THE RESULT IS DERIVED BY OUR EVALUATION AS DISCUSSED IN SECTION IV-A8. "*" DENOTES THAT THE RESULT IS UNFAIR FOR COMPARISON AND WILL BE FURTHER DISCUSSED IN SECTION IV-E. THE RESULTS IN THIS TABLE ARE OBTAINED WITHOUT ANY POST PROCESSING SUCH AS QUERY EXPANSION [64].

Methods	Holidays	Holidays + 1M	Paris 6K	Paris 106K	Oxford 5K	Oxford 105K
Tolias 2013 [36]	82.2	71.3†	78.2	70.5†	81.7	75.0
Mikulik 2013 [53]	75.8	69.4†	74.9	67.5	74.2	67.4
Qin 2013 [58]	82.1	N/A	73.6	N/A	78.0	72.8
Shi 2015 [59]	88.1	N/A	77.5	N/A	80.4	68.9
Jegou 2010 [61]	81.3	N/A	N/A	N/A	61.5	51.6
Zheng 2015 [1]	88.0	75.0	81.2†	72.5†	76.2†	71.1†
Babenko 2015 [47]	80.2	N/A	N/A	N/A	65.7	64.2
Filip 2016 [71]	79.5	N/A	83.8*	76.4*	79.7	73.9
Rezende 2017 [26]	86.3	N/A	N/A	N/A	64.8	62.5
Gordo 2016 [49]	86.7	N/A	87.1*	79.7*	83.1	78.6
Husain 2017 [63]	73.2	N/A	N/A	N/A	59.0	56.1
Tolias 2016 [51]	82.2†	70.0†	78.2†	69.5†	81.7†	72.3†
Razavian 2015 [39]	N/A	N/A	N/A	N/A	58.9	57.8
Babenko 2014 [41]	N/A	N/A	N/A	N/A	67.6	61.1
Zheng 2014 (1) [18]	80.2	69.6	N/A	N/A	N/A	N/A
Zheng 2014 (2) [52]	77.5	72.3	N/A	N/A	N/A	N/A
Do 2018 [72]	74.1	72.5	N/A	N/A	63.7	62.2
Tolias 2015 [73]	N/A	N/A	83.0	**75.7**	66.9	61.6
Ours	**89.4**	**77.2**	**83.2**	73.4	**83.1**	**79.5**

for each query, instead of being used as feature vectors. The dynamic match kernel produces a large gap between the number of positive and negative matches, which leads to the best performance on most datasets except for the UKBench dataset. CIE+ [3] outperforms the proposed method on the UKBench dataset, but it relies on a deep iterative process and a direct combination of the local and deep features, which incurs a heavy burden on memory and time consumption.

Note that, as aforementioned, we use a simple deep framework (AlexNet [20]) in this paper, while some recent methods, *e.g.*, VGGNet [75], GoogleNet [76] and ResNet [77], exploit more complicated and deeper networks which can also be integrated into the proposed dynamic match kernel framework for performance gain.

E. Extension to Large-Scale Image Retrieval

In this section, we extend the proposed semantic-constrained image retrieval framework with the dynamic match kernel to the large-scale datasets. Fig. 8 shows the comparison on image retrieval performance (mAP) between the static and dynamic match kernels on three datasets with increasing amounts of distractors. For both retrieval frameworks based on static and dynamic match kernels, the performance on three datasets decline as the number of distractors increases. Nevertheless, the proposed framework with dynamic match kernel always performs better compared to the methods with static one, which demonstrates that the dynamic match kernel as well as the proposed method are scalable on large scale extensions.

In table VI, we compare the proposed method with other state-of-the-art methods on the large-scale datasets. The performance of all three datasets drops with the increasing number of distractors, no matter what approaches were applied. For example, the mAP of the method in [59] drops from 80.4% on Oxford5K dataset to 68.9% on the Oxford105K datasets. In contrast, the proposed approach drops from 83.05% to 79.46%, which outperforms all the others. Both frameworks in [49] and [71] employ the VGG architecture [75] to obtain the reported results. However, as is reported in [71], the mAP result on Paris106K dataset with AlexNet architecture [20] is 61.8% (while the proposed method with AlexNet achieves mAP=73.41% on the same dataset). Therefore, the performance of these methods relies heavily on the quality of the feature engineering of the existing deep architecture.

F. Comparison with Selective Match Function

Tolias *et al.* [51] propose a selective match function f_s for Eq. 1 defined by $f_s(h(b_x, b_y)) = (g(h(b_x, b_y)))^\alpha$ where $h(\cdot,\cdot)$ denotes the Hamming distance, $g(\cdot)$ denotes a mapping function and the exponent α is fixed for all queries. The selective match function f_s can be used to weaken the effect of false correspondences but is only determined by the local spatial relationships.

In this section, we validate an alternative way to enlarge the gap between the influence of positive and negative matches. We extend the static selective function f_s in [51] to the dynamic version f_s^* by incorporating the global semantic relationship:

$$f_s^*(h(b_x, b_y)) = \left(1 - \frac{h(b_x, b_y)}{l_H}\right)^{\alpha(d_s)} \tag{11}$$

where the dynamic exponent $\alpha(d_s)$ is defined as

$$\alpha(d_s) = d_s \times \omega. \tag{12}$$

TABLE VII
COMPARISON TO THE DYNAMIC SELECTIVE MATCH FUNCTION WITH ("W/") OR WITHOUT "W/O") POST-PROCESSING. DYN-T DENOTES THE DYNAMIC THRESHOLD AND DYN-S REPRESENTS THE DYNAMIC SELECTIVE MATCH FUNCTION.

Datasets	W/O		W/	
	DYN-T	DYN-S	DYN-T	DYN-S
Holidays	87.92	**88.97**	91.11	**92.06**
UKBench	3.82	**3.83**	3.88	**3.89**
Parix6K	**84.92**	83.43	**87.22**	86.51
Oxford5K	**83.05**	79.66	**85.11**	80.87
DupImages	89.43	**90.22**	**91.00**	90.58

Here, d_s denotes the semantic distance between two images based on deep features, the parameter ω controls the intensity of the dynamic selective match function. We evaluate ω on the Holidays, DupImage and Paris6K datasets and show the results in Fig. 6(c), which inspires us to set $\omega = 22$ in rest of the experiments.

The dynamic exponent $\alpha(d_s)$ is considered as the counterpart of $h_t(d_s)$ in Eq. 5. Table VII reports the comparison between the dynamic selective match function in Eq. 11 and the dynamic threshold in Eq. 5 on five benchmarks. The post-processing indicates the graph-based re-ranking method proposed by Zhang *et al*. [13]. The proposed method achieves comparable performance to the dynamic selective match function in terms of both mAP or N-S score.

Furthermore, we carry out experiments on a large scale dataset, *i.e*., the Holidays dataset with 1M distractors, to compare the scalability and efficiency of the proposed dynamic threshold and the selective match function, of which the results are reported in Table VIII. While the performance on mAP or N-S score is also comparable, the feature matches of the proposed method is about $1/1000$ of that of the selective match function. Consequently, the query time of the selective match function based method [36], [51] is over 8 times that of the proposed method on same computer.

When the proposed dynamic threshold is applied, numerous false correspondences are removed so that less non-zero items are introduced into Eq. 7 to compute the similarities between a query and the candidate images. As discussed above, the number of matches is determined by the threshold h_t and $h_t(d_s)$ in Eq. 2 and Eq. 6, respectively. The method with the selective match function employs a fixed threshold h_t. Therefore, even if the ideal condition is reached, *i.e*., all negative matches are assigned the weight as $f_s^* = 0$, the algorithm still wastes time handling a large amount of false correspondences. In contrast, by dynamically estimating the threshold based on the semantic relationship, the proposed method aims to directly remove as many false correspondences as possible while retaining true ones for high efficiency.

G. Computation Time

In this paper, experiments are conducted on a computer with 64GB RAM, and the processor is Intel Xeon 2.40 GHz CPU. For extracting the feature of the fully connected layer from the AlexNet, we use a GeForce GTX980 with 4GB RAM.

TABLE VIII
COMPARISON ON THE EFFICIENCY OF THE SELECTIVE MATCH FUNCTION AND THE PROPOSED DYNAMIC MATCH KERNEL. FOR EACH QUERY OF THE HOLIDAYS DATASET WITH 1M DISTRACTORS, WE REPORT THE NUMBER OF MATCHES AND THE TIME CONSUMED BY BOTH METHODS.

Methods	#Matches per Query	Time per Query
Selective Match Function	67083788	7.33s
Dynamic Match Kernel	85152	0.89s

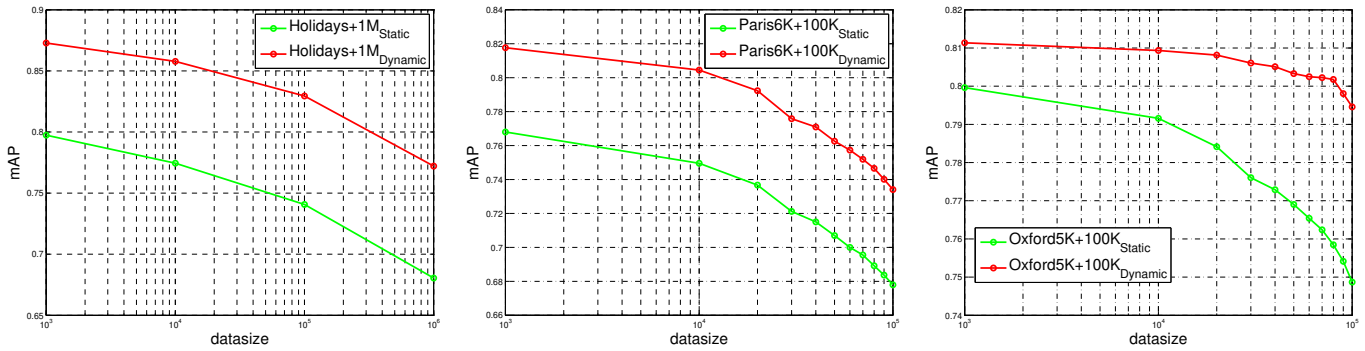
TABLE IX
COMPUTATION TIME FOR GENERATING DYNAMIC THRESHOLDS ON BENCHMARKS AND LARGE-SCALE EXTENSIONS IN SECONDS. "TOTAL" MEANS THE TIME ON THE WHOLE DATASET FOR EXTRACTING DEEP FEATURES AND CALCULATING DYNAMIC MATCH KERNELS. "AVERAGE" FOR DEEP FEATURE EXTRACTION DENOTES THE TIME FOR EACH IMAGE, WHILE "AVERAGE" FOR KERNEL CONSTRUCTION DENOTES THE PARAMETER CALCULATION TIME FOR EACH QUERY (*e.g*., EACH OF THE 500 QUERIES FOR HOLIDAYS DATASET).

Datasets	Deep Feature Extraction		Kernel Construction	
	Total	Average	Total	Average
Holidays	61.8138	0.0415	0.2777	0.0006
UKBench	308.8487	0.0303	27.5104	0.0108
Parix6K	342.7969	0.0536	0.2808	0.0051
Oxford5K	184.8187	0.0365	0.1993	0.0036
DupImages	21.9930	0.0199	0.0673	0.0006
Holidays+1M	25438.3212	0.0254	211.2747	0.4225
Paris106K	4841.9589	0.0455	1.4965	0.0272
Oxford105K	4683.9807	0.0445	1.2891	0.0234

We use a pre-trained CNN to extract deep features, which are sequentially used to calculate d_s and $h_t(d_s)$. Table IX illustrates the additional time for extracting deep features and calculating dynamic thresholds on five benchmark datasets and three large-scale datasets. The feature extraction process is quite efficient since it takes less than 0.06s per image for all datasets. The additional time for the calculation of the dynamic thresholds is also negligible. For instance, it takes 0.4225s to calculate $100,1490$ dynamic thresholds for each query on Holidays+1M dataset.

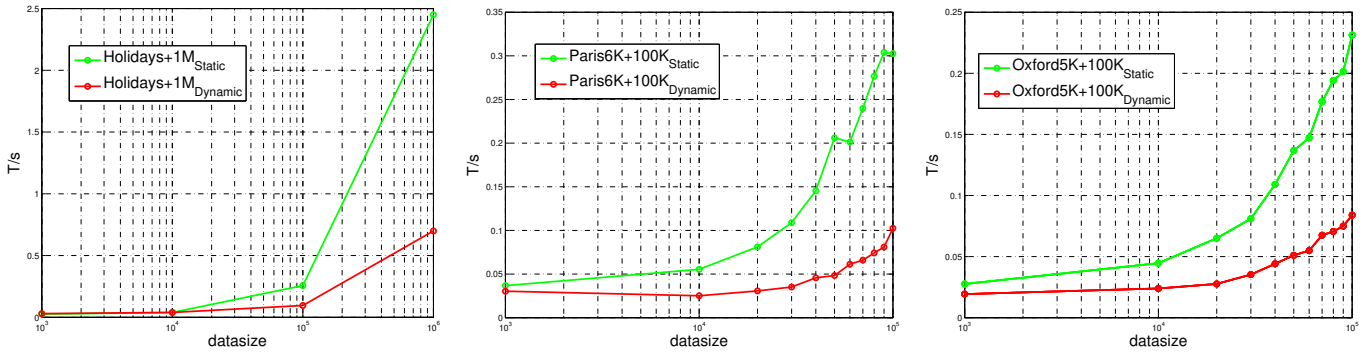
Calculating the semantic relationship of the database can be considered as a pre-processing for the algorithm, which should be done only once at the beginning of the retrieval system. Note although extra time is needed for constructing the dynamic match kernels, we reduce the overall query time of the retrieval process due to the substantial elimination of negative matches in the query stage which simplifies the calculation of the $S(\boldsymbol{I}_q, \boldsymbol{I}_c)$ in Eq. 7. As shown in Table X, we save over $1/3$ query time (about 1.3s for each query) on the Holidays+1M dataset. Considering the additional time required for calculating the dynamic match kernel (about 0.5s for each query as shown in Table IX), we save around 1.8s in the query stage, which considerably improves the efficiency.

Fig. 9 provides a clearer visualization of the average query time on three datasets with increasing numbers of distractors. The proposed method needs less time in the query stage than the static kernel in all situations. For instance, on the Holidays+1M dataset, the average query time for the static match kernel is near 2.5s, while the proposed dynamic match



(a) Holidays+1M (b) Paris6K+100K (c) Oxford5K+100K

Fig. 8. Comparisons on image retrieval performance (mAP) against the size of the datasets for both the static match kernel (**green**) and the proposed dynamic match kernel (**red**). The maximum number of distractors is 1M in figure (a) and 100K in (b) and (c). The proposed dynamic match kernel gets favorable performance against the static one on large-scale datasets. Detailed analysis can be found in Section IV-E.



(a) Holidays+1M (b) Paris6K+100K (c) Oxford5K+100K

Fig. 9. Average query time against the datasize for static match kernel (**green**) and proposed dynamic match kernel (**red**). The maximum number of distractors is 1M in figure (a) and 100K in (b) and (c). Analysis can be found in Section IV-G.

TABLE X
OVERALL QUERY TIME FOR EACH QUERY ON THREE LARGE-SCALE DATASETS IN SECONDS. THE PROPOSED DYNAMIC MATCH KERNEL SHOWS HIGHER EFFICIENCY IN THE QUERY STAGE. MORE DETAILED ANALYSIS CAN BE FOUND IN SECTION IV-G.

Dataset	Holidays+1M	Paris106K	Oxford105K
Static	3.8775	0.9814	0.8404
Dynamic	**2.5749**	**0.8593**	**0.7610**

kernel only takes about 0.7s on average for a query.

V. CONCLUSIONS

In this paper, we propose a semantic-constrained retrieval framework which incorporates holistic image representations with dynamic match kernel. In contrast to the static match kernel, the dynamic one filters out most negative matches from the initial set while retaining most of the positive ones. The proposed method leverages both local and global cues to calculate the similarity relationship between query and candidates, which can be easily combined with other state-of-the-art modules for image retrieval. Extensive experimental results show that the proposed algorithm outperforms the state-of-the-art methods on five benchmark datasets and the corresponding large-scale extensions.

ACKNOWLEDGMENT

This research was supported by NSFC (NO. 61620106008, 61572264, 61633021, 61525306, 61301238, 61201424), NSF CAREER (No. 1149783) and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR).

REFERENCES

[1] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1741–1750. 1, 7, 9, 10

[2] A. Iscen, M. Rabbat, and T. Furon, "Efficient large-scale similarity search using matrix factorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2073–2081. 1

[3] W. Zhou, H. Li, J. Sun, and Q. Tian, "Collaborative index embedding for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1154–1166, 2018. 1, 3, 10

[4] 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. 1, 2, 3, 4, 5, 6, 7

[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8. 1, 2, 6, 7

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. 1, 2

[7] F. S. Khan, R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell, and A. M. Lopez, "Color attributes for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3306–3313. 1, 2

[8] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan, "Cross-modal retrieval with CNN visual features A new baseline," *IEEE Transactions on Cybernetics*, vol. 47, no. 2, pp. 449–460, 2017. 1, 2

[9] F. Zhu, X. Kong, L. Zheng, H. Fu, and Q. Tian, "Part-based deep hashing for large-scale person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4806–4817, 2017. 1

[10] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477. 1

[11] L. Zheng, S. Wang, and Q. Tian, "ℓ_p-norm IDF for scalable image retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3604–3617, 2014. 1

[12] X. Li, Z. Lin, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5153–5161. 1, 9

[13] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific fusion for image retrieval," in *European Conference on Computer Vision*, 2012, pp. 660–673. 1, 9, 11

[14] J. Wan, P. Wu, S. C. Hoi, P. Zhao, X. Gao, D. Wang, Y. Zhang, and J. Li, "Online learning to rank for content-based image retrieval," in *International Joint Conference on Artificial Intelligence*, 2015, pp. 2284–2290. 1

[15] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Fast image retrieval: Query pruning and early termination," *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 648–659, 2015. 1

[16] D. Niestér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2161–2168. 1, 2, 6

[17] Z. Liu, H. Li, W. Zhou, R. Hong, and Q. Tian, "Uniting keypoints: Local visual information fusion for large-scale image search," *IEEE Transactions on Multimedia*, vol. 17, no. 4, pp. 538–548, 2015. 1, 4

[18] L. Zheng, S. Wang, Z. Liu, and Q.Tian, "Packing and padding coupled multi-index for accurate image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1939–1946. 1, 2, 7, 9, 10

[19] I. G. Diaz, M. Birinci, F. D. de Maria, and E. J. Delp, "Neighborhood matching for image retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 544–558, 2017. 1, 9

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105. 2, 3, 4, 5, 7, 9, 10

[21] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned CNN embedding for person re-identification," *ACM Transactions on Multimedia Computing*, vol. 14, no. 1, p. 13, 2017. 2

[22] 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, 2018. 2, 9

[23] 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*, 2008, pp. 1–8. 2, 6

[24] W. Zhou, H. Li, Y. Lu, and Q. Tian, "SIFT match verification by geometric coding for large-scale partial-duplicate web image search," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 9, no. 1, p. 4, 2013. 2, 7

[25] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative," in *International Conference on Multimedia Information Retrieval*, 2010, pp. 527–536. 2, 7

[26] R. Rezende, J. Zepeda, J. Ponce, F. Bach, and P. Perez, "Kernel square-loss exemplar machines for image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271. 2, 9, 10

[27] L. Chen, D. Xu, I. W.-H. Tsang, and X. Li, "Spectral embedded hashing for scalable image retrieval," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 1180–1190, 2014. 2

[28] Y. Cao, M. Long, J. Wang, and S. Liu, "Deep visual-semantic quantization for efficient image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 916–925. 2

[29] W. W. Ng, X. Tian, Y. Lv, D. S. Yeung, and W. Pedrycz, "Incremental hashing for semantic image retrieval in nonstationary environments," *IEEE Transactions on Cybernetics*, vol. PP, no. 99, pp. 1–13, 2017. 2

[30] L. Liu, M. Yu, and L. Shao, "Unsupervised local feature hashing for image similarity search," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2548–2558, 2016. 2

[31] W. Zhou, M. Yang, X. Wang, H. Li, Y. Lin, and Q. Tian, "Scalable feature matching by dual cascaded scalar quantization for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 159–171, 2016. 2, 7

[32] G. Tolias and O. Chum, "Asymmetric feature maps with application to sketch based retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6185–6193. 2

[33] Y. Cao, M. Long, J. Wang, and S. Liu, "Collective deep quantization for efficient cross-modal retrieval," in *The Association for the Advancement of Artificial Intelligence*, 2017, pp. 3974–3980. 2

[34] S. Li, S. Purushotham, C. Chen, Y. Ren, and C.-C. J. Kuo, "Measuring and predicting tag importance for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2423–2436, 2017. 2

[35] L. Zhang, L. Wang, and W. Lin, "Generalized biased discriminant analysis for content-based image retrieval," *IEEE Transactions on Cybernetics*, vol. 42, no. 1, pp. 282–290, 2012. 2

[36] G. Tolias, Y. Avrithis, and H. Jégou, "To aggregate or not to aggregate: Selective match kernels for image search," in *IEEE International Conference on Computer Vision*, 2013, pp. 1401–1408. 2, 5, 7, 9, 10, 11

[37] J. Y.-H. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2015, pp. 53–61. 2, 10

[38] Y. Liu, Y. Guo, S. Wu, and M. S. Lew, "Deepindex for accurate and efficient image retrieval," in *Internation Conference on Multimedia Retrieval*, 2015, pp. 43–50. 2, 10

[39] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016. 2, 10

[40] J. Yu, X. Yang, F. Gao, and D. Tao, "Deep multimodal distance metric learning using click constraints for image ranking," *IEEE Transactions on Cybernetics*, vol. 47, no. 12, pp. 4014–4024, 2017. 2

[41] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *European Conference on Computer Vision*, 2014, pp. 584–599. 2, 10

[42] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *European Conference on Computer Vision*, 2014, pp. 392–407. 2, 10

[43] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2014, pp. 806–813. 2, 10

[44] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronnin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *IEEE International Conference on Computer Vision*, 2015, pp. 91–99. 2, 10

[45] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004. 2

[46] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, "Convolutional kernel networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 2627–2635. 3

[47] A. Babenko and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," in *IEEE International Conference on Computer Vision*, 2015, pp. 1269–1277. 3, 10

[48] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marqués, and X. Giró-i Nieto, "Bags of local convolutional features for scalable instance search," in *International Conference on Multimedia Retrieval*, 2016, pp. 327–331. 3

[49] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *European Conference on Computer Vision*, 2016, pp. 241–257. 3, 10

[50] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, "Semantic-aware co-indexing for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 37, pp. 2573–2587, 2015. 3

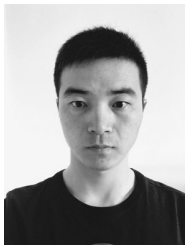
[51] G. Tolias, Y. Avrithis, and H. Jégou, "Image search with selective match kernels: aggregation across single and multiple images," *International Journal of Computer Vision*, vol. 116, no. 3, pp. 247–261, 2016. 7, 9, 10, 11

[52] L. Zheng, S. Wang, and Q. Tian, "Coupled binary embedding for large-scale image retrieval," *IEEE Transactions on Image Processing*, vol. 23, pp. 3368–3380, 2014. 7, 9, 10

[53] A. Mikulik, M. Perdoch, O. Chum, and J. Matas, "Learning vocabularies over a fine quantization," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 163–175, 2013. 7, 9, 10
[54] M. Perd'och, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 9–16. 7
[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. 7
[56] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, "Semantic-aware co-indexing for image retrieval," in *IEEE International Conference on Computer Vision*, 2013, pp. 1673–1680. 9
[57] R. Tao, E. Gavves, C. Snoek, and A. Smeulders, "Locality in generic instance search from one example," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2091–2098. 9
[58] D. Qin, C. Wengert, and L. van Gool, "Query adaptive similarity for large scale object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1610–1617. 9, 10
[59] M. Shi, Y. Avrithis, and H. Jégou, "Early burst detection for memory-efficient image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 605–613. 9, 10
[60] T.-T. Do, Q. D. Tran, and N.-M. Cheung, "FAemb: a function approximation-based embedding method for image retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3556–3564. 9
[61] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316–336, 2010. 9, 10
[62] H. Jegou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1169–1176. 9
[63] S. S. Husain and M. Bober, "Improving large-scale image retrieval through robust aggregation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1783–1796, 2017. 9, 10
[64] L. Xie, Q. Tian, W. Zhou, and B. Zhang, "Heterogeneous graph propagation for large-scale web image search," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4287–4298, 2015. 9, 10
[65] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2911–2918. 9
[66] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 889–896. 9
[67] F. Yang, B. Matei, and L. S. Davis, "Re-ranking by multi-feature fusion with diffusion for image retrieval," in *IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 572–579. 9
[68] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image classification and retrieval are ONE," in *International Conference on Multimedia Retrieval*, 2015, pp. 3–10. 10
[69] L. Zheng, S. Wang, J. Wang, and Q. Tian, "Accurate image search with multi-scale contextual evidences," *Internation Journal of Computer Vision*, vol. 120, no. 1, pp. 1–13, 2016. 10
[70] Y. Li, X. Kong, L. Zheng, and Q. Tian, "Exploiting hierarchical activations of neural network for image retrieval," in *ACM International Conference on Multimedia*, 2016, pp. 132–136. 10
[71] F. Radenović, G. Tolias, and O. Chum, "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples," in *European Conference on Computer Vision*, 2016, pp. 3–20. 10
[72] T.-T. Do and N.-M. Cheung, "Embedding based on function approximation for large scale image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 626–638, 2018. 10
[73] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," in *ICLR*, 2015. 10
[74] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307. 9
[75] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015. 10
[76] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9. 10
[77] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. 10

Jufeng Yang is an associate professor in the College of Computer and Control Engineering, Nankai University. He received the PhD degree from Nankai University in 2009. From 2015 to 2016, he was working at the Vision and Learning Lab, University of California, Merced. His research falls in the field of computer vision, machine learning and multimedia.

Jie Liang is currently a Master student with the College of Computer and Control Engineering, Nankai University. His current research interests include computer vision, machine learning, pattern recognition and optimization.



Hui Shen received the B.E. and M.S. degree in computer Science and Technology from Nankai University, China, in 2014 and 2017 separately. He is a researcher at Vision Technology Institute of Baidu. His current research interests include image retrieval and object detection.

Kai Wang is an associate professor in the College of Computer and Control Engineering, Nankai University. He received the PhD degree in control theory and control engineering from Nankai University in 2006. His research interests include computer vision and pattern recognition. He is a member of the ACM, the IEEE Computer Society and the CCF.

Paul L. Rosin is a professor at the School of Computer Science & Informatics, Cardiff University. His research interests include the representation, segmentation, and grouping of curves, knowledge-based vision systems, early image representations, low level image processing, machine vision approaches to remote sensing, methods for evaluation of approximation algorithms, medical and biological image analysis, mesh processing, non-photorealistic rendering and the analysis of shape in art and architecture.

Ming-Hsuan Yang is a professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in Computer Science from the University of Illinois at Urbana-Champaign in 2000. Yang has served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, Image and Vision Computing, Computer Vision and Image Understanding, and Journal of Artificial Intelligence Research. He received the NSF CAREER award in 2012 and the Google Faculty Award in 2009.