

# From Selective Deep Convolutional Features to Compact Binary Representations for Image Retrieval

THANH-TOAN DO\*, University of Liverpool, United Kingdom  
 TUAN HOANG\*, Singapore University of Technology and Design, Singapore  
 DANG-KHOA LE TAN, Singapore University of Technology and Design, Singapore  
 HUU LE, Queensland University of Technology, Australia  
 TAM V. NGUYEN, University of Dayton, United States  
 NGAI-MAN CHEUNG, Singapore University of Technology and Design, Singapore

In the large-scale image retrieval task, the two most important requirements are the discriminability of image representations and the efficiency in computation and storage of representations. Regarding the former requirement, Convolutional Neural Network (CNN) is proven to be a very powerful tool to extract highly-discriminative local descriptors for effective image search. Additionally, in order to further improve the discriminative power of the descriptors, recent works adopt fine-tuned strategies. In this paper, taking a different approach, we propose a novel, computationally efficient, and competitive framework. Specifically, we firstly propose various strategies to compute masks, namely *SIFT-mask*, *SUM-mask*, and *MAX-mask*, to select a representative subset of local convolutional features and eliminate redundant features. Our in-depth analyses demonstrate that proposed masking schemes are effective to address the burstiness drawback and improve retrieval accuracy. Secondly, we propose to employ recent embedding and aggregating methods which can significantly boost the feature discriminability. Regarding the computation and storage efficiency, we include a hashing module to produce very compact binary image representations. Extensive experiments on six image retrieval benchmarks demonstrate that our proposed framework achieves the state-of-the-art retrieval performances.

CCS Concepts: • **Computing methodologies** → **Visual content-based indexing and retrieval**;

Additional Key Words and Phrases: Content Based Image Retrieval, Image Hashing, Embedding, Aggregating, Deep Convolutional Features, Unsupervised

## 1 INTRODUCTION

Content-based image retrieval has been an active research field for decades and attracted a sustained attention from the computer vision/multimedia communities due to its wide range of applications, e.g., visual search, place recognition. Earlier works heavily rely on hand-crafted local descriptors, e.g., SIFT [41] and its variant [2]. Although a lot of great efforts have been made to improve performances of the SIFT-based image search systems, their performances are still limited. There

---

\* indicates equal contribution.

Authors' addresses: Thanh-Toan Do\*, University of Liverpool, United Kingdom, thanh-toan.do@liverpool.ac.uk; Tuan Hoang\*, Singapore University of Technology and Design, Singapore, nguyenanhuan\_hoang@mymail.sutd.edu.sg; Dang-Khoa Le Tan, Singapore University of Technology and Design, Singapore, letandang\_khoa@sutd.edu.sg; Huu Le, Queensland University of Technology, Australia, huu.le@qut.edu.au; Tam V. Nguyen, University of Dayton, United States, tamnguyen@udayton.edu; Ngai-Man Cheung, Singapore University of Technology and Design, Singapore, ngaiman\_cheung@sutd.edu.sg.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

XXXX-XXXX/2019/3-ART \$15.00

<https://doi.org/0000001.0000001>

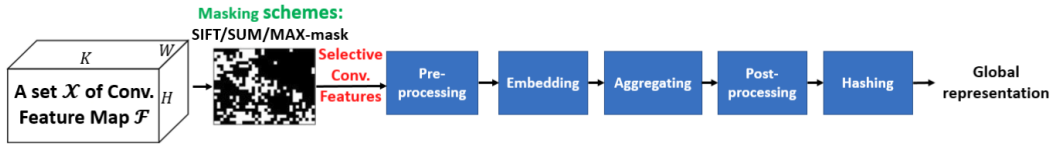


Fig. 1. The overview of our proposed framework to produce discriminative global binary representations.

are two main limitations with the SIFT features. The first and the most important one is the low-discriminability of SIFT features [4] which is necessary to emphasize the differences in images. Although the limitation have been relieved to some extent by embedding local features to a much higher dimensional space [10, 33, 35, 45, 54], the semantic gap between human understanding on objects/scenes and SIFT-based image representation is still considerable large [4]. Secondly, the *burstiness* effect [30], i.e., numerous descriptors are almost identical within an image, significantly degrades the quality of SIFT-based image representation [8, 30, 33].

Recently, deep Convolutional Neural Networks (CNN) achieve great success in various problems including image classification [24, 37, 53, 59], semantic segmentation [22, 39], object detection [14, 52] and image retrieval [1, 4, 36, 38, 62, 66]. While the output of the deeper layers, e.g., fully-connected, can be helpful for the image retrieval task [16]. Recent works [1, 4, 36, 38, 62, 66] show that using the outputs of middle layers, e.g., convolution layers, can help to enhance the retrieval performances by larger margins.

Even though the local convolutional (conv.) features are more discriminative than SIFT features [4], the burstiness issue, which may appear in the local conv. features, has not been investigated previously. In this paper, by delving deeper into the burstiness issue, we propose three different masking schemes to select *highly-representative* local conv. features and robustly eliminate redundant local features. The masking schemes are named as **SIFT-mask**, **SUM-mask**, and **MAX-mask**. The elimination of redundant local features results in more discriminative representation and efficient computation, we will further discuss these advantages in the experiment section. The fundamentals of our proposal are that we utilize SIFT detector [41] to produce SIFT-mask; additionally, we apply sum-pooling and max-pooling across all conv. feature maps to derive SUM-mask and MAX-mask, respectively. Note that our idea of using SIFT coordinate for CNN based image retrieval is novel. Our SUM-mask is also new. Previous works apply sum-pooling within a feature map; our mask is computed by summing across feature maps. Moreover, while max-pooling [62] gets the maximum value, our MAX-mask obtains the location of that value.

In addition, the majority of recent works, that work on local conv. features [36, 49, 62], do not utilize feature embedding and aggregating methods [10, 33, 35, 45], which are useful steps to boost the discriminability for SIFT features. In [4], the authors discussed that the deep conv. features are already discriminative enough for image retrieval task; hence, the embedding step is unnecessary. However, we find that applying the state-of-the-art embedding and aggregating [10, 33, 35, 45] can significantly help to enhance the discriminability of image representations. Therefore, by applying embedding and aggregating on our selective local conv. features, the aggregated representations improve image retrieval performance significantly.

Furthermore, in order to achieve compact binary codes, we cascade a state-of-the-art unsupervised hashing method, e.g., Iterative Quantization (ITQ) [15], Relaxed Binary Autoencoder (RBA) [12], Simultaneous Compression and Quantization (SCQ) [25], into the proposed system. The binary representations would help to achieve significant benefits in retrieval speed and memory consumption. Fig. 1 presents the overview of the proposed framework. In summary, in this work we make following contributions.

**Contributions.** A preliminary version of this work has been presented in [26]. In the preliminary version [26], we first propose various novel masking schemes, which are proven to effectively

eliminate redundant local conv. features. Secondly, we leverage the state-of-the-art embedding and aggregating methods to produce highly-discriminative global image representations. We comprehensively evaluate different components to build an efficient framework that achieves the state-of-the-art retrieval performance on standard benchmark datasets when using real-value image representations. In this current version, we introduce additional contributions as follows: Firstly, we conduct analysis to explain how various masking schemes work, both qualitatively and quantitatively. Secondly, we show that assembling information of different abstract levels is beneficial in the image retrieval task as this could help to produce more informative and discriminative representations. Thirdly, we then further optimize the framework to solve two crucial problems for large scale image search, i.e., searching speed and storage. Specifically, we propose to cascade a state-of-the-art unsupervised hash function into the framework to further binarize real-valued aggregated representations to binary representations. Note that binary representations would allow the fast searching and efficient storage which are very critical in large scale search systems. In addition, we also conduct very large scale experiments, i.e., on Flickr1M dataset [29], which consists of over one million images. The experiments on such kind of large scale dataset are necessary to confirm the effectiveness of the proposed method for real applications which usually have to deal with very large scale datasets. By the best of our knowledge, our work is the first deep learning-based retrieval method which conducts the evaluation on that kind of large scale dataset. We also conduct more experiments to deeply analyze the effectiveness of the proposed framework and to extensively compare to the state of the art. The extensive experiments on six benchmark datasets show that the proposed framework significantly outperforms the state of the art when images are represented by either real-valued representations or compact binary representations.

We organize the remainders of this paper as follows. Section 2 presents related works. Section 3 presents our main contributions of the proposed masking schemes. Section 4 presents the proposed framework for computing the final image representation from a set of selected local deep conv. features. Section 5 presents comprehensive experiments to evaluate our proposed framework. Section 6 concludes the paper.

## 2 RELATED WORK

In the last few years, image retrieval has witnessed an increasing of performance due to the use of better image representations, i.e., deep features obtained from pre-trained CNN models, which are trained on image classification task. The early CNN-based work [51] directly used deep fully-connected (FC) activations for the image retrieval. Instead of directly using features from the pre-trained networks for the retrieval as [51], other works apply different processings on the pre-trained features to enhance the discriminability. Gong *et al.* [16] proposed Multi-Scale Orderless Pooling to embed and aggregate CNN FC activations of image patches of an image at different scales. Hence, the final features are more invariant to the scale. However, as multiple patches (cropped and resized to a specific size) of an image are fed forward into the CNN, the method endures a high computational cost. Yan *et al.* [69] revisited the SIFT feature and suggested that SIFT and CNN FC features are highly complementary. Therefore, they proposed to integrate SIFT features with CNN FC features at multiple scales. Concurrently, Liu *et al.* [40] proposed ImageGraph to fuse various types of features, e.g., CNN FC features, BoW on SIFT [41] descriptors, HSV color histogram, and GIST features [44]. This method even though achieves very good performances, it requires very high-dimensional features. Furthermore, ImageGraph must be built on database images, which may be prohibitive on large scale datasets.

Recently, many image retrieval works shift the attention from FC features to conv. features. This is because outputs of lower layers contain more general information and spatial information is still preserved [3]. In this case, the conv. features are considered as local features. Hence, the

sum-pooling or max-pooling method is usually applied to achieve a single representation. Babenko and Lempitsky [4] demonstrated that by whitening the final image representation, sum-pooling can outperform max-pooling. Kalantidis *et al.* [36] proposed to learn weights for both feature channels and spatial locations which helps to enhance the discriminability of sum-pooling representation on conv. features. Tolias *et al.* [62] revisited max-pooling by proposing the strategy to aggregate the maximum activation over multiple spatial regions sampled on a output of a conv. layer using a fixed layout. Similarly, Jian Xu *et al.* [68] proposed to aggregate features which are weighted using probabilistic proposals.

Instead of using pre-trained features (with / without additional processing) for the retrieval task. In [5], Babenko *et al.* showed that fine-tuning an off-the-shelf network (e.g., AlexNet [37] or VGG [53]) can produce more discriminative deep features [5] for the image search task. However, collecting labeled training data is non-trivial [5]. Recent works tried to overcome this challenge by proposing unsupervised/weakly-supervised fine-tuning approaches which are specific for image retrieval. Arandjelovic *et al.* [1] proposed the NetVLAD architecture which can be trained in an end-to-end fashion. The author also proposed to collect from Google Street View Time Machine in a weakly-supervised process. Adopting a similar approach, Cao *et al.* [6] proposed to harvest data from Flickr with GPS information to form GeoPair dataset [60]. The dataset is afterward used to train the special Quartet-net architecture. Radenovic *et al.* [49], concurrently, proposed a different approach to fine-tune a pretrained CNN on classification task for image retrieval. The authors propose to use 3D reconstruction to obtain matching / non-matching image pairs in an unsupervised manner for fine-tuning process. Recently, Noh *et al.* [27] proposed the DEep Local Features (DELf) pipeline with attention-based keypoint selection for large scale image retrieval. The model is fine-tuned using their proposed Google Landmark dataset. However, the pipeline requires the geometric verification using RANSAC. Even though, features are compressed to very low dimensions, e.g., 40-D, for the trade-off between compactness, speed and discrimination, the process is still computation and memory intensive. Besides, the self-supervised approach [43, 58] to fine-tuning the models is also an interesting approach to enhance the discrimination power of the CNN models for the image retrieval task.

In regards to compact image representations, the earlier work [71] presented feature dimension selection on embedded high-dimensional features as a compression method to achieve compact representations. Radenovic *et al.* [49, 50] later introduced to learn the whitening and dimensionality reduction in the supervised manner resulting in better performances than the baseline PCA method. Albert *et al.* [19] made use of the product quantization [32] to compress image representations. This approach even though achieves good accuracy, it is not as efficient as the hashing approach, which we utilize in this paper, in term of retrieval time [13]. Do *et al.* [12] proposed to produce binary representations by simultaneously aggregating raw local features and hashing. Differentially, in this paper, we proposed various masking schemes in combination with a complete framework to produce more discriminative binary representations. Taking similar approach with [49] to mining the training datasets of matching / non-matching image pairs, Do *et al.* [9] proposed to directly learn the compact binary codes from input images. In BGAN [56], the authors utilize Generative Adversarial Networks (GAN) [17] to generate binary codes that can well represented for images in the retrieval task. Recently, Song *et al.* proposed Deep Region Hashing (DRH) [57] which computes binary codes for both global and local features. In which the global binary codes are used to obtain initial ranking, and the local binary codes are used for regional re-search (re-ranking). Similar to [62], re-ranking can help to improve performance; however, this approach results in significant increases in storage as all local binary codes are requires to be stored. Additionally, additional processing time is also required. Hence, this approach may not be scalable for very large-scale databases, e.g., millions or billions of images. In addition to the quantization and hashing methods,

in Quantization-Based Hashing (QBH) [55], the author proposed a novel approach to combine the advantages of quantization-based methods and hashing methods. We would like to refer readers to [72] for a more comprehensive survey on image retrieval.

### 3 SELECTIVE LOCAL DEEP CONVOLUTION FEATURES

In this section, firstly, we define the set of local deep conv. features which we work on throughout the paper (Section 3.1). We then propose in details the masking schemes to select a subset of discriminative local conv. features, including **SIFT-mask**, **SUM-mask**, and **MAX-mask** (Section 3.2). Finally, we provide in-depth analyses and experiments to qualitatively and quantitatively confirm the effectiveness of the proposed methods (Section 3.3).

#### 3.1 Local deep convolutional features

We consider a pre-trained CNN in which all fully connected layers are discarded. Given an input image  $I$  of size  $W_I \times H_I$  that is fed through a CNN, the 3D activation tensor of a conv. layer has the size of  $W \times H \times K$  dimensions, where  $K$  is the number of feature maps and  $W \times H$  is the spatial resolution of a feature map. We consider this 3D tensor as a set  $\mathcal{X}$  of  $(W \times H)$  local features; each of them has  $K$  dimensions. We denote  $\mathcal{F}^{(k)}$  as  $k$ -th feature map with size of  $W \times H$ .

#### 3.2 Selective features

Inspired by the concept of finding the interest keypoints in the input images in traditional designs of hand-crafted features, we propose to select discriminative local deep conv. features.

We now formally propose different methods to compute a selection mask, i.e., a set of unique coordinates  $\{(x, y)\}$  ( $1 \leq x \leq W$ ;  $1 \leq y \leq H$ ) in the feature maps where local conv. features are retained.

**3.2.1 SIFT-Mask.** In the image retrieval task, prior to the era of CNN, most previous works [8, 28, 30, 33, 35, 45, 61] rely on SIFT [41] features and its variant RootSIFT [2]. Although the gap between the SIFT-based representation and the semantic meaning of an image is still large, these early works have clearly demonstrated the capability of SIFT feature, especially in the potential of key-point detection. Fig. 2 - Row (2) shows local image regions which are covered by SIFT. We can observe that SIFT features mainly cover the salient regions, i.e., buildings. This means that SIFT keypoint detector is capable of locating important regions of images. Hence, we propose to take the advantage of SIFT detector in combination with highly-discriminative local conv. features. We will discuss more about the SIFT-mask in Section 3.3.

Specifically, let set  $\mathcal{S} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$  be SIFT feature locations extracted from an  $W_I \times H_I$  image;  $1 \leq x^{(i)} \leq W_I$ ,  $1 \leq y^{(i)} \leq H_I$ . Based on the fact that conv. layers still preserve the spatial information of the input image [62], we select locations on the spatial grid  $W \times H$  (of the feature map) which correspond to locations of SIFT key-points, i.e.,

$$\mathcal{M}_{\text{SIFT}} = \left\{ \left( x_{\text{SIFT}}^{(i)}, y_{\text{SIFT}}^{(i)} \right) \right\} \quad i = 1, \dots, n; \quad (1)$$

where  $x_{\text{SIFT}}^{(i)} = \text{round} \left( \frac{x^{(i)} W}{W_I} \right)$  and  $y_{\text{SIFT}}^{(i)} = \text{round} \left( \frac{y^{(i)} H}{H_I} \right)$ , in which  $\text{round}(\cdot)$  represents rounding to nearest integer. By keeping only locations  $\mathcal{M}_{\text{SIFT}}$ , we expect to remove “background” conv. features, while retaining “foreground” ones.

**3.2.2 MAX-Mask.** It is widely known that each feature map contains the activations of a specific visual structure [14, 70]. Hence, we propose to select the local conv. features which contain high activations for all visual contents. In other words, we select the local features that capture the most prominent structures in the input images. These features are highly desirable to differentiate scenes.

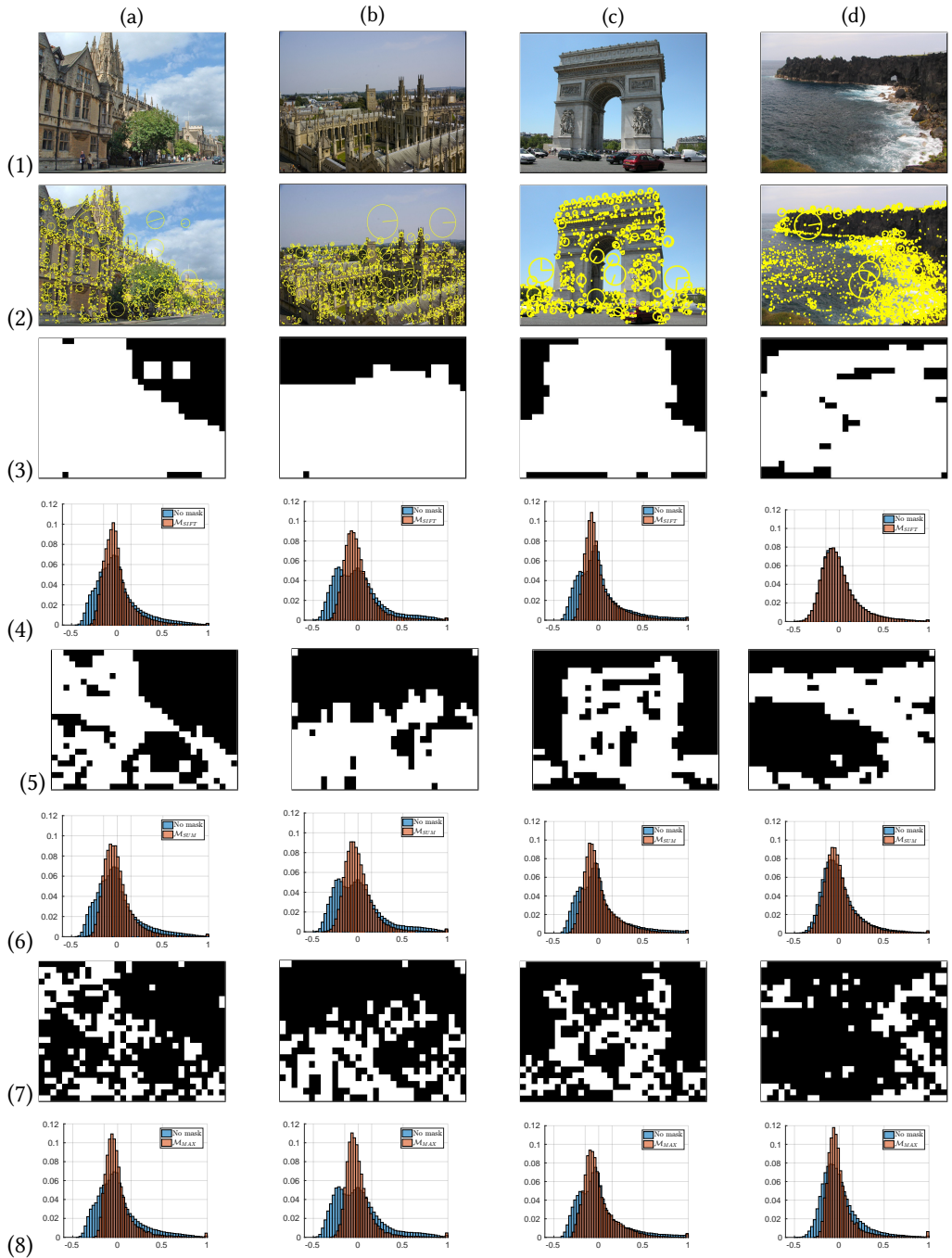


Fig. 2. Examples of SIFT/SUM/MAX-masks to select local conv. features. The first row shows the original images. The second row shows regions which are covered by SIFT features. The 3rd, 5th, and 7th rows respectively show the SIFT/SUM/MAX-masks of corresponding images (in the 1st row). The 4th, 6th, and 8th rows show the normalized histograms of covariances of sets of local conv. features with/without applying the SIFT/SUM/MAX-masks, respectively.

In specific, we assess each feature map and select the location corresponding to the max activation value on that feature map. We formally define the selected locations  $\mathcal{M}_{\text{MAX}}$  as follows:

$$\begin{aligned} \mathcal{M}_{\text{MAX}} &= \left\{ \left( x_{\text{MAX}}^{(k)}, y_{\text{MAX}}^{(k)} \right) \right\} \quad k = 1, \dots, K; \\ \left( x_{\text{MAX}}^{(k)}, y_{\text{MAX}}^{(k)} \right) &= \arg \max_{(x, y)} \mathcal{F}_{(x, y)}^{(k)}. \end{aligned} \quad (2)$$

**3.2.3 SUM-Mask.** Departing from the MAX-mask idea, we propose a different masking method based on the motivation that a local conv. feature is more *informative* if it gets excited in more feature maps, i.e., the sum on description values of a local feature is larger. By selecting local features having large values of sum, we can expect that those local conv. features are very informative about various local image structures [70]. The selected locations  $\mathcal{M}_{\text{SUM}}$  is defined as follows:

$$\begin{aligned} \mathcal{M}_{\text{SUM}} &= \left\{ (x, y) \mid \Sigma_{(x, y)}^{\mathcal{F}} \geq \alpha \right\}, \\ \Sigma_{(x, y)}^{\mathcal{F}} &= \sum_{k=1}^K \mathcal{F}_{(x, y)}^{(k)}, \quad \alpha = \text{median}(\Sigma^{\mathcal{F}}). \end{aligned} \quad (3)$$

### 3.3 Effectiveness of the proposed masking schemes

We now deeply analyze the effectiveness of the proposed masking schemes, in both qualitative and quantitative results.

SIFT detector [41] is designed to detect interesting points which are robust with variations in scale, noise and illumination; therefore, these interesting points usually locate on high-contrast regions of images, e.g., corners. These regions also usually contain detail structures of the scenes which are necessary in differentiating scenes. While smooth regions, e.g., sky, road surfaces, are ignored as these regions are mainly background and contribute very little information. Hence, by using the SIFT-mask, we expect to select local conv. features at higher-contrast, i.e. potentially informative regions. However, there are two main issues when using the SIFT-mask: (i) in cases of blurry images, the SIFT detector unsurprisingly fails to locate informative regions. (ii) However, having too many interesting points also causes unexpected outcomes, which is known as the burstiness effect [30], i.e., too many redundant local features are selected. For example, in Fig. 2-(2d)<sup>1</sup> and 2-(3d), SIFT-mask includes almost all local features of the sea regions, which are obviously redundant

On the other hand, SUM/MAX-masks perform much better when selecting just a few features at the sea region, i.e. Fig. 2-(5d) and 2-(7d) respectively, which are necessary to distinguish scenes with and without sea, and not to cause a serious burstiness effect which potentially makes the distinguishing different scenes with sea regions difficult. In fact, the burstiness effect is the main reason explaining why SIFT-mask underperforms SUM/MAX-mask rather than due to SIFT-mask fails to select important regions, which is also confirmed by the two facts: (i) SUM/MAX-masks are mainly subsets of SIFT-mask, and (ii) applying SIFT-mask helps to improve performances (compared to no mask) which means that important regions have been selected, otherwise performances will drop. Note that the empirical results will be presented in Section 5.2.1. It is worth noting that, the burstiness effect on local conv. features is expectedly less severe since local conv. features have much larger receptive fields than those of SIFT features. Specifically, a local conv. feature extracted from pool5 layer of AlexNet [37] and VGG16 [53] have the receptive fields of  $195 \times 195$  and  $212 \times 212$  respectively. We will further investigate this effect in Section 4.4.

<sup>1</sup>Row (2) and column (d) of Fig. 2.

Comparing SUM-mask and MAX-mask, which are computed from learned features, they both have the capability of detecting important regions based on the responded activation of regions. However, their principles of selecting local features are different. In particular, given prominent regions, the corresponding local conv. features of those regions are usually highly activated. As a result, the sums on those features are larger. This fact explains why SUM-mask more densely selects local conv. features at prominent regions. However, as the receptive fields of neighbouring features are largely overlapping, they are likely to contain similar information, i.e., redundant. On the other hand, MAX-mask only selects the features which have highest activation values. Hence, we expect MAX-mask can select the best features for representing the visual structures of prominent regions. As a result, we minimize the chance of selecting multiple similar local features.

Besides, we quantitatively evaluate the effectiveness of our proposed masking schemes in eliminating redundant local conv. features. Firstly, Fig. 3a shows the averaged percentage of the remaining local conv. features after applying our proposed masks on *Oxford5k* [47], *Paris6k* [48], and *Holidays* [31] datasets (Section 5.1). Note that local conv. features are extracted from pool5 layer of the pre-trained VGG [53] with the input image size of  $\max(W_I, H_I) = 1024$ . Apparently, SIFT/SUM/MAX-masks remove large numbers of local conv. features, about 25%, 50%, and 70% respectively.

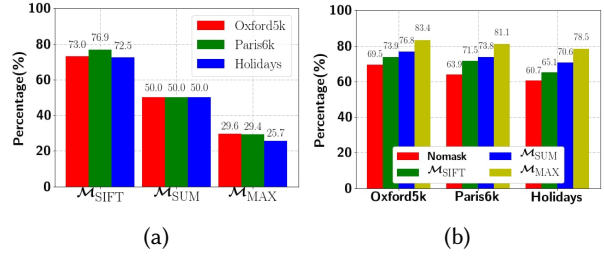


Fig. 3. Fig. 3a: The averaged percentage of remaining local conv. features after applying masks. Fig. 3b: The averaged percentage of the covariance values in the range of  $[-0.15, 0.15]$ .

In addition, we present the normalized histograms of covariances of selected local conv. features after applying different masks in Fig. 2-Row 4th, 6th, and 8th. To compute the covariances, we first  $l_2$ -normalize local conv. features, and then compute the dot products for all pairs of features. For easy comparison, the normalized histograms of covariances of all available local conv. features (i.e., before masking) are included. We can clearly observe that the distributions of covariances after applying masks have much higher peaks around 0 and have smaller tails than those without applying masks. This indicates that the masks are helpful in reducing correlation between features. Additionally, Fig. 3b shows the averaged percentage of  $l_2$ -normalized feature pairs whose dot products are within the range of  $[-0.15, 0.15]$ . The chart shows that the selected features are more uncorrelated. In summary, Fig. 3 shows that the proposed masking schemes are effective in removing a large proportion of redundant local conv. features. As a result, we can select a better representative subset of local conv. features. Furthermore, as the number of features is reduced, the computational cost is also considerably reduced, especially for the subsequent embedding and aggregating steps.

## 4 FRAMEWORK: EMBEDDING AND AGGREGATING ON SELECTIVE CONVOLUTION FEATURES

In this section, we introduce the completed framework which takes a set of local deep conv. features to compute the final image representation.

### 4.1 Pre-processing

Given a set  $\mathcal{X}_{\mathcal{M}} = \{\mathbf{x}_{(x,y)} \mid (x,y) \in \mathcal{M}_*\}$ , where  $\mathcal{M}_* \in \{\mathcal{M}_{\text{SUM}}, \mathcal{M}_{\text{MAX}}, \mathcal{M}_{\text{SIFT}}\}$ , of selective  $K$ -dimensional local conv. features belonged to the set, we apply the principal component analysis (PCA) to compress local conv. features to a lower dimension  $d$ :  $\mathbf{x}^{(d)} = M_{\text{PCA}}\mathbf{x}$ , where  $M_{\text{PCA}}$  is the



PCA-matrix. Applying PCA for dimension reduction can be very beneficial for two reasons. Firstly, using low-dimensional local features can help to produce compact final image representations as done in recent state-of-the-art image retrieval methods [4, 49, 62]. Secondly, applying PCA could be helpful in removing noise and redundancy; hence, enhancing the discrimination. The compressed features are subsequently  $l_2$ -normalized.

## 4.2 Embedding

We additionally aim to boost the discrimination power of the selective local conv. features. This task can be accomplished by embedding the local features to a high-dimensional space:  $\mathbf{x} \mapsto \phi(\mathbf{x})$ , using state-of-the-art embedding methods: *Fisher vector* – FV [45], *vector of locally aggregated descriptors* – VLAD [33], *triangulation embedding* – Temb [35], *function approximation-based embedding* – F-FAemb [10]. It is worth noting that while in [4], the authors mentioned that local conv. features are already discriminative; hence the embedding step is not necessary. However, in this work, we find that embedding the *selected* features to higher dimension significantly improves their discriminability.

## 4.3 Aggregating

Let  $\mathbf{V}_i = [\phi(\mathbf{x}_1^i), \dots, \phi(\mathbf{x}_{n_i}^i)]$  be an  $D \times n_i$  matrix that contains  $n_i$   $D$ -dimensional embedded local descriptors of  $i$ -th image. In earlier works, the two common methods to aggregate a set of local features to a single global one are max-pooling ( $\psi_m$ ) and sum/average-pooling ( $\psi_s/\psi_a$ ). Recently, H. Jégou et al. [35] introduced **democratic aggregation** ( $\psi_d$ ) method applied to image retrieval problem. The fundamental idea of democratic aggregation is to equalize the similarity between each local features and the aggregated representation. Note that, concurrently, Murray and Perronnin [42] proposed **Generalized Max Pooling (GMP)** ( $\psi_{GMP}$ ), which shares the similar idea with *democratic aggregation*. Democratic aggregation can be directly applied on various embedded features, e.g., FV [45], VLAD [33], Temb [35], F-FAemb [10]. Moreover, when working with embedded SIFT features, this aggregation method has been shown to clearly outperform both max-pooling and sum/average-pooling [35]. Noted that democratic aggregation requires local features to be  $l_2$ -normalized.

## 4.4 Post-processing

**Power-law normalization (PN).** The *burstiness* of visual elements [30] is the phenomenon that numerous descriptors are almost similar within an image. The burstiness can severely impact the similarity measure between two images. An effective solution to the burstiness issue is to apply PN [46] to and subsequently  $l_2$ -normalize [35] the aggregated features  $\psi$ . The PN formulation is defined as  $PN(x) = \text{sign}(x)|x|^\alpha$ , where  $0 \leq \alpha \leq 1$  [46].

By the best of our knowledge, no previous work has re-studied the *burstiness* phenomena on the local conv. features. Fig. 4 shows the effect of PN on local conv. features using various proposed masking schemes. The figure shows that the burstiness still happens on local conv. features ( $CNN + \phi_\Delta + \psi_d$ ), as the retrieval performance changes as  $\alpha$  varies. However, we additionally observe that the burstiness on conv. features is much weaker than on SIFT features

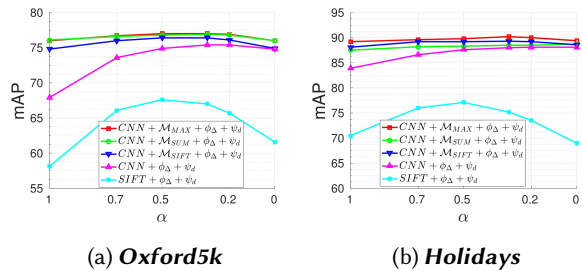


Fig. 4. Impact of power-law normalization factor  $\alpha$  on retrieval performance. Following the setting in [35], we set  $d = 128$  and  $|C| = 64$  for both SIFT and conv. features. The local conv. features are extracted from pool5 layer of the pre-trained VGG [53].

( $SIFT + \phi_\Delta + \psi_d$ ). More importantly, the proposed SIFT/SUM/MAX-masks clearly mitigate the burstiness phenomena: the performances achieved by  $CNN + \mathcal{M}_{MAX/SUM/SIFT} + \phi_\Delta + \psi_d$  are stable as  $\alpha$  varies. This confirms the effectiveness of the proposed masking schemes in removing redundant local features. Following previous works,  $\alpha$  is set at 0.5 for all later experiments, unless stated otherwise.

**Rotation normalization and dimension reduction (RN).** Besides the visual burstiness, frequent co-occurrences issue is also an important limitation. Fortunately, this effect can be easily addressed by whitening the data.

#### 4.5 Hashing function

In the large scale image retrieval problem, binary hashing, where images are represented by a  $L$ -bit binary codes, is an attractive approach because the binary representations allow the fast searching and sufficient storage.

There is a wide range of hashing methods have been proposed in the literature, in both unsupervised and supervised [20, 65]. Although supervised hashing methods usually outperform unsupervised hashing methods on some specific retrievals in which the data is labeled, they are not suitable for the general image retrieval (which is focused in this work). That is because in the general image retrieval, the label of an image is not well defined. Most of general image retrieval benchmarks, e.g., Holidays, Oxford5k, Paris6k, does not have labeled training data. On the other hand, the unsupervised hashing is well suitable for the general image retrieval task. Unsupervised hashing methods do not require the data label for training. Most of unsupervised hashing methods tries to preserve the geometric structure of data by using reconstruction criterion [7, 11, 12, 15] or directly preserving the distance similarity between samples [67]. By above reasons, we propose to cascade a state-of-the-art unsupervised hash function, i.e., Iterative Quantization (ITQ) [15], K-mean Hashing (KMH) [23], or Relaxed Binary Autoencoder (RBA) [12], into the framework to further binarize the real-valued aggregated representations to binary representations.

The overview of our proposed framework is shown in Fig. 1. In the next section, we will conduct extensive experiments to evaluate the framework in both cases: real-valued global representations (i.e., without the hash function in the framework), and binary global representations (i.e., with the hash function in the framework).

## 5 EVALUATION

In this section, we conduct a wide range of experiments to comprehensively evaluate the proposed framework on six standard image retrieval benchmark datasets, including Oxford5k dataset [47], Paris6k dataset [48], INRIA Holidays [31] dataset, Oxford105k dataset [47], Paris106k dataset [48], and Holidays+Flickr1M dataset [29].

### 5.1 Datasets, evaluation protocols, and implementation notes

**Oxford Buildings dataset:** The *Oxford5k* dataset [47] consists of 5,063 images of buildings and 55 query images corresponding to 11 distinct buildings in Oxford. Each query image contains a bounding box indicating the region of interest. Following the standard practice [10, 18, 35, 62], we use the cropped query images based on provided bounding boxes.

**Paris dataset:** The *Paris6k* dataset [48] consists of 6412 images of famous landmarks in Paris. Similar to *Oxford5k*, this dataset has 55 queries corresponding to 11 landmarks. We also use provided bounding boxes to crop the query images accordingly.

**INRIA Holidays dataset:** The *Holidays* dataset [31] contains 1,491 images corresponding to 500 scenes. The query image set consists of one image from each scene. Following [4, 5, 36], we manually rotate images (by  $\pm 90$  degrees) to fix the incorrect image orientation.

**Oxford105k and Paris106k datasets:** We additionally combine *Oxford5k* and *Paris6k* with 100k Flickr images [47] to form larger databases, named *Oxford105k* and *Paris106k* respectively. The new databases are used to evaluate retrieval performance at a larger scale.

**Holidays+Flickr1M:** In order to evaluate the retrieval on a very large scale, we merge *Holidays* dataset with 1M negative images downloaded from Flickr [29], forming the *Holidays+Flickr1M* dataset. This dataset allows us to evaluate real-like scenarios of the proposed framework.

**Evaluation protocols:** Follow the state of the art [1, 4, 10, 18, 35, 49], the retrieval performance is measured by mean average precision (**mAP**) over the query sets. Additionally, the *junk* images are removed from the ranking.

**Implementation notes:** In the image retrieval task, to avoid overfitting, it is important to use held-out datasets (training set) to learn all necessary parameters [4, 18, 49]. Following standard settings in the literature [4, 10, 35, 62], we use the set of 5,000 Flickr images [47]<sup>3</sup> as the training set to learn parameters for

*Holidays* and *Holidays+Flickr1M*. The *Oxford5k* is used as the learning set for *Paris6k* and *Paris106k*, while the *Paris6k* is used as the learning for *Oxford5k* and *Oxford105k*. For fair comparison, following recent works [4, 18, 49, 62], we use the pretrained VGG16 [53] (with Matconvnet toolbox [64]) to extract deep conv. features. In addition, all images are resized so that the maximum dimension is 1,024 while preserving aspect ratios before fed into the CNN. We utilize the VLFeat toolbox [63] for SIFT detector. For clarity, the notations are summarized in Table 1. The implementation of the proposed framework is available at <https://github.com/hnanhtuan/selectiveConvFeature>.

## 5.2 Effects of parameters

**5.2.1 Frameworks.** In this section, we conduct experiments to comprehensively evaluate various embedding and aggregating methods in combination with different proposed masking schemes. Note that, we follow [10] to decompose the embedding and aggregating steps of VLAD and FV methods. This allows us to utilize the state-of-the-art aggregations (e.g., democratic pooling [35]).

In order to have a fair comparison among different combinations, we empirically set the visual codebook size- $|C|$  and the number of retained PCA components- $d$  (of local conv. features) such that the produced final aggregation vectors of different methods have the same dimensionality- $D$ .

These parameters are presented in Table 2.

Table 1. Notations and their corresponding meanings.

Notations	Meanings	Notations	Meanings
$M_{\text{SIFT}}$	SIFT-mask	$\psi_a$	Average-pooling
$M_{\text{SUM}}$	SUM-mask	$\psi_s$	Sum-pooling
$M_{\text{MAX}}$	MAX-mask	$\psi_d$	Democratic-pooling [35]
$\phi_{\text{FV}}$	FV [45]	$\phi_{\text{VLAD}}$	VLAD [33]
$\phi_{\Delta}$	Temb [35]	$\phi_{\text{F-FAemb}}$	F-FAemb [10]
$C$	Codebook <sup>2</sup>	$d$	Retained PCA dim.
$D$	Final dim.		

Table 2. Configurations of different embedding methods.

Methods	$d$	$ C $	$D$
FV [45]	48	44	$2 \times d \times  C  = 4224$
VLAD [33]	64	66	$d \times  C  = 4224$
T-emb [35]	64	68	$d \times  C  - 128 = 4224$
F-FAemb [10] <sup>4</sup>	32	10	$\frac{( C  - 2) \times d \times (d + 1)}{2} = 4224$

<sup>3</sup>We randomly select 5,000 images from the 100k Flickr image set [47].

<sup>3</sup>For FV method, the codebook is learned by Gaussian Mixture Model. For VLAD, Temb, and F-FAemb methods, the codebooks learned by K-means.

<sup>4</sup>Instead of removing the first  $d(d + 1)/2$  components as in original design [10], we remove the first  $d(d + 1)$  components of the features after aggregating step (Section 4.3) as this generally achieves better performances.

Table 3. Comparison of different frameworks. For simplicity, we do not include the notations for post-processing steps (PN and RN). The “**Bold**” values indicate the best performances in each masking scheme and the “Underline” values indicate the best performances across all settings.

	Frameworks	$\mathcal{M}_{\text{MAX}}$	$\mathcal{M}_{\text{SUM}}$	$\mathcal{M}_{\text{SIFT}}$	None
<b>Oxford5k</b>	$\phi_{\text{FV}} + \psi_{\text{a}}$	67.8	65.1	65.5	59.5
	$\phi_{\text{FV}} + \psi_{\text{d}}$	72.2	71.8	72.0	69.6
	$\phi_{\text{VLAD}} + \psi_{\text{s}}$	66.3	65.6	66.4	65.1
	$\phi_{\text{VLAD}} + \psi_{\text{d}}$	69.2	70.5	71.3	69.4
	$\phi_{\Delta} + \psi_{\text{d}}$	<u>75.8</u>	<b>75.7</b>	<b>75.3</b>	73.4
	$\phi_{\text{F-FAemb}} + \psi_{\text{d}}$	75.2	74.7	74.4	<b>73.8</b>
<b>Paris6k</b>	$\phi_{\text{FV}} + \psi_{\text{a}}$	78.4	76.4	75.8	68.0
	$\phi_{\text{FV}} + \psi_{\text{d}}$	84.5	82.2	82.4	76.9
	$\phi_{\text{VLAD}} + \psi_{\text{s}}$	77.7	74.5	76.0	73.2
	$\phi_{\text{VLAD}} + \psi_{\text{d}}$	80.3	79.5	81.3	79.3
	$\phi_{\Delta} + \psi_{\text{d}}$	<b>86.9</b>	84.8	85.3	<b>83.9</b>
	$\phi_{\text{F-FAemb}} + \psi_{\text{d}}$	86.6	<b>85.9</b>	<b>85.6</b>	82.9
<b>Holidays</b>	$\phi_{\text{FV}} + \psi_{\text{a}}$	83.2	80.0	81.5	78.2
	$\phi_{\text{FV}} + \psi_{\text{d}}$	87.8	86.7	87.1	85.2
	$\phi_{\text{VLAD}} + \psi_{\text{s}}$	83.3	82.0	83.6	82.7
	$\phi_{\text{VLAD}} + \psi_{\text{d}}$	85.5	86.4	87.5	86.1
	$\phi_{\Delta} + \psi_{\text{d}}$	<b>89.1</b>	88.1	<b>88.6</b>	<b>87.3</b>
	$\phi_{\text{F-FAemb}} + \psi_{\text{d}}$	88.6	<b>88.4</b>	88.5	86.4

We report the comparative results on *Oxford5k*, *Paris6k*, and *Holidays* datasets in Table 3. The main observations from Table 3 are: (i) democratic pooling is clearly better than sum/max-pooling, (ii) our proposed masking schemes consistently boost performance for all embedding and aggregating frameworks, and finally (iii) the MAX-mask outperforms the SUM/SIFT-masks, while the performance gains of SUM-mask and SIFT-mask are comparable. At the comparison dimensionality of  $4224 - D$ , the two frameworks  $\phi_{\Delta} + \psi_{\text{d}}$  and  $\phi_{\text{F-FAemb}} + \psi_{\text{d}}$  achieve comparable performances for various masking schemes and datasets. Hence, we choose  $\mathcal{M}_{*} + \phi_{\Delta} + \psi_{\text{d}}$  as our default framework for analyzing other parameters.

5.2.2 *Final feature dimensionality.* Since our framework provides the flexibility of choosing different dimensions for final representations, we evaluate the impact of final image representation on the retrieval performance.

Considering our default framework –  $\mathcal{M}_{*} + \phi_{\Delta} + \psi_{\text{d}}$ , we empirically set the number of retained PCA components (of local conv. features) and the codebook size for different dimensionalities in Table 4. For compact final representations of 512-D, we choose  $d = 32$  to avoid using too few visual words as this drastically degrades performance [35]. For longer final representations, i.e. 1024, 2048, 4096, imitating Fisher and VLAD presentations for SIFT features [34],

we reduce local conv. features to  $d = 64$ . For the largest considered representation, i.e. 8064, imitating the Temb representation for SIFT features [35], we reduce local conv. features to  $d = 128$ . Note that the settings in Table 4 are applied for all later experiments.

Table 4. Number of retained PCA components (of local conv. features) and codebook size (of T-emb) for different dimensionalities.

Dim. $D$	512	1024	2048	4096	8064
$d$	32	64	64	64	128
$ C $	20	18	34	66	64

The Figure 5 shows the retrieval performances at different final feature dimensionalities for *Oxford5k* and *Paris6k* datasets. Unsurprisingly, the proposed framework can achieve higher performance gains when the final feature dimensionality increases. At 4096-D or higher, the improvements become small (or even decreased for  $\mathcal{M}_{SIFT} + \phi_{\Delta} + \psi_d$  scheme on *Paris6k* dataset). More important, the masking schemes consistently boost retrieval performances across different dimensionalities.

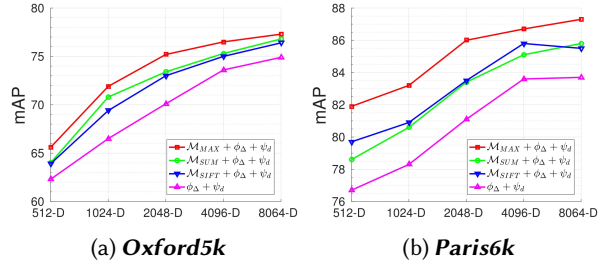


Fig. 5. Impact of the final representation dimensionality on retrieval performance.

**5.2.3 Image size.** Since our framework highly depends on the number of local conv. features, it is necessary to evaluate the performance of our framework with a smaller image size. We present the retrieval performance on *Oxford5k* and *Paris6k* datasets with the image sizes of  $\max(W_I, H_I) = 1024$  and  $\max(W_I, H_I) = 724$  in Table 5. Similar to the reported results of [62] on *Oxford5k* dataset, we observe around 6-7% drop in mAP when using smaller input images of  $\max(W_I, H_I) = 724$  rather than the original images. While on *Paris6k* dataset, interestingly, the performances are more stable to changes of the image size. We observe a small drop of 2.2% mAP on *Paris6k* dataset for R-MAC [62] with our experiments. The experimental results suggest that R-MAC [62] and our methods are equivalently affected by the change in the image size.

Table 5. Impact of different input image sizes on retrieval performance. The framework of  $\mathcal{M}_{MAX/SUM} + \phi_{\Delta} + \psi_d$  is used to produce image representations.

Dim. $D$	$\max(W_I, H_I)$	Oxford5k		Paris6k	
		$\mathcal{M}_{SUM}$	$\mathcal{M}_{MAX}$	$\mathcal{M}_{SUM}$	$\mathcal{M}_{MAX}$
512	724	56.4	60.9	79.3	81.2
	1024	64.0	65.7	78.6	81.6

The large performance drops on *Oxford5k* can be explained that with higher resolution images, the CNN can take a closer “look” on smaller details in the images. Hence, the local conv. features can better distinguish details in different images. While the stable performance on *Paris6k* dataset can be perceived that the differences among scenes are at global structures, i.e., a higher abstract level, instead of small details as on *Oxford5k* dataset. This explanation is actually consistent with human understanding on these datasets.

**5.2.4 Layer selection.** In [4], the authors mentioned that deeper conv. layers produce features that are more reliable in differentiating images. Here, we re-evaluate this statement using our proposed framework by comparing the retrieval performance (mAP) of features extracted from different conv. layers, including conv5-3, conv5-2, conv5-1, conv4-3, conv4-2, and conv4-1, at the same dimensionality. The experimental results on *Oxford5k* and *Paris6k* datasets are shown in Figure 6. We observe that the performances are slightly decreased when using lower conv. layers until conv4-3. It means that conv. features of these layers, e.g., conv5-3, conv5-2, conv5-1, conv4-3, are still very discriminative. Hence, combining information of these layers may be beneficial. However, when going down further to conv4-2 and conv4-1, the performances are significantly lower. In summary, regarding the pre-trained VGG network [53], the last conv. layer (i.e., conv5-3) produces the most reliable representation for image retrieval.

Table 6. Impact of combining multiple conv. layers as hyper-column feature maps on **Oxford5k**, **Paris6k**, and **Holidays** datasets. The framework of  $\mathcal{M}_{\text{MAX}} + \phi_{\Delta(64,18)} + \psi_d$  is used to produce image representations, where  $\phi_{\Delta(64,18)}$  denotes Temb with  $d = 64$  and  $|C| = 18$ .

conv5-3	conv5-2	conv5-1	Oxford5k	Paris6k	Holidays
✓			72.2	83.2	88.4
✓	✓		73.3	83.5	90.4
✓		✓	74.2	83.8	90.9
✓	✓	✓	74.8	84.5	90.8

Table 7. Comparison of different frameworks when the final representations are binary values. The values in brackets in **Embedding** column indicate the dimension of local conv. features after PCA and the codebook size, respectively. **Dim.** column indicates the dimension of real-valued representations before subjecting into a hash function. We evaluate the binary representations at code lengths 64, 128, 256, 512 with three state-of-the-art unsupervised hashing methods ITQ [15], RBA [12], and KMH [23]. Results are reported on **Oxford5k**, **Paris6k** and **Holidays** dataset.

	Embed	Dim.	Oxford5k				Paris6k				Holidays			
			64	128	256	512	64	128	256	512	64	128	256	512
ITQ [15]	$\phi_{\Delta(32,20)}$	512	18.3	<b>31.6</b>	45.0	<b>57.3</b>	<b>32.9</b>	49.7	63.0	74.7	57.5	70.7	<b>79.5</b>	<b>83.5</b>
	$\phi_{\Delta(64,18)}$	1024	<b>18.7</b>	29.5	42.9	55.8	33.5	49.0	61.0	72.4	<b>58.7</b>	71.5	79.4	82.9
	$\phi_{\Delta(64,34)}$	2048	18.3	27.7	38.4	50.3	28.4	44.3	57.9	68.6	57.9	71.1	79.1	82.3
	$\phi_{\Delta(64,66)}$	4096	16.0	23.0	33.6	45.8	26.1	40.1	52.9	65.4	56.1	70.2	78.8	80.5
RBA [12]	$\phi_{\Delta(32,20)}$	512	17.8	31.3	<b>45.3</b>	57.1	31.5	<b>50.8</b>	<b>63.7</b>	<b>74.9</b>	56.7	<b>71.6</b>	78.5	83.2
	$\phi_{\Delta(64,18)}$	1024	18.4	30.1	42.7	55.7	32.8	49.4	61.3	72.8	57.6	71.1	79.3	82.7
	$\phi_{\Delta(64,34)}$	2048	17.3	30.9	39.1	53.5	29.0	45.0	59.1	68.6	57.1	70.8	78.8	81.8
	$\phi_{\Delta(64,66)}$	4096	15.7	25.1	39.0	48.3	25.8	39.3	55.6	64.6	55.6	67.5	77.5	80.1
KMH [23]	$\phi_{\Delta(32,20)}$	512	18.5	26.5	39.1	54.4	32.0	45.7	61.6	75.3	53.4	65.0	75.0	80.8
	$\phi_{\Delta(64,18)}$	1024	18.1	28.1	41.4	53.5	30.0	48.3	61.4	73.7	54.6	68.0	78.0	82.4
	$\phi_{\Delta(64,34)}$	2048	15.7	26.7	38.5	51.4	26.0	40.8	56.7	69.2	51.2	68.4	76.4	81.2
	$\phi_{\Delta(64,66)}$	4096	13.4	20.7	31.2	47.0	21.1	33.3	49.8	63.7	50.0	63.2	72.8	80.1

Furthermore, as assembling multiple conv. layers of CNN would be beneficial [21] in localizing the saliency objects, we conduct additional experiments to evaluate whether combining different levels of abstraction from different conv. layers of CNN be useful for the retrieval task. Specifically, we concatenate feature maps from different layers as hyper-column feature maps which allow to normally use the proposed masking schemes. The experimental results

are reported in 6, from which we observe that combining multiple conv. layers as hyper-column features helps to improve performances across many datasets, e.g., *Oxford5k*, *Paris6k*, and *Holidays*.

**5.2.5 Binary representation framework.** In this section, we conduct experiments at a wide range of settings to find the setting that produces the best binary representation. As discussed in section 5.2.3 and 5.2.4, when using images of  $\max(W_I, H_I) = 1024$ , the last conv. layer of the VGG network

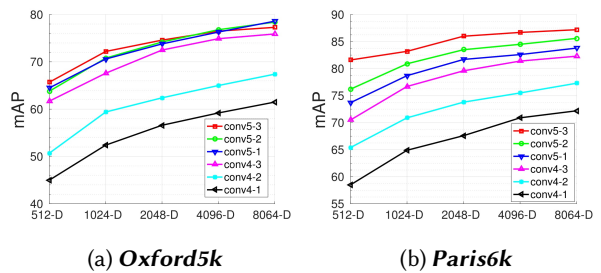


Fig. 6. Impact of local deep conv. features from different layers on retrieval performance. The framework of  $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_d$  is used to produce image representations.

[53], i.e., conv5-3, produces the most reliable representations. Hence, the default framework of  $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\text{d}}$ , in combination with these two settings (i.e.,  $\max(W_I, H_I) = 1024$  and conv5-3 features), is used to produce real-valued representations before passing forward to a hashing module. Furthermore, in the literature, unsupervised hashing methods are usually proposed to work with hand-crafted global image features, e.g., GIST [44], or deep learning features of a fully-connected layer, e.g., fc7 of AlexNet or VGG, it is unclear which method works the best with our proposed aggregated representations. Hence, we conduct experiments with various state-of-the-art unsupervised hashing methods including iterative quantization (ITQ) [15], relaxed binary autoencoder (RBA) [12], and K-means hashing (KMH) [23] to find the best hashing module for our framework.

The experimental results on *Oxford5k*, *Paris6k*, and *Holidays* datasets are presented in Table 7. There are some main observations from the results. Firstly, at the same code length, ITQ and RBA achieve comparable results, while both these methods significantly outperforms KHM, on all datasets. Secondly, as discussed in Section 5.2.2, embedding local conv. features to a higher dimensional space helps to enhance the discriminative power of the real-valued aggregated representation; however, as shown in Table 7, embedding to too high-dimensional space also causes information loss when producing compact binary codes, i.e., the best mAPs are achieved when the aggregated representation are at 512 or 1024 dimensions. The higher dimensional representations (i.e., 2048-D or 4906-D) cause the more mAPs loss. As embedding to 512-D, i.e.,  $\phi_{\Delta(32,20)}$ , gives most stable results, we use this configuration in our final framework when producing binary representations.

### 5.3 Comparison to the state of the art

We comprehensively evaluate and compare our proposed framework with the state of the art in the image retrieval task. We separate two experimental settings. The first experiment is when images are represented by mid-dimensional real-valued presentations. The second experiment is when images are represented by very compact representations, i.e., very short real-valued vectors or binary vectors.

*5.3.1 Comparison with the state of the art when images are represented by mid-dimensional real-valued vectors.* We report comparative results when images are represented by real-valued vectors in Table 8. We separate two different settings, i.e., when deep features are extracted from an off-the-shelf pretrained VGG network and are extracted from a VGG network which is fine-tuned for the image retrieval task.

**Using off-the-shelf VGG network [53].** The first observation is that at the dimensionality of 1024, our framework using MAX-mask ( $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\text{d}}$ ) achieves the best mAP in comparison to recent deep learning-based methods [1, 4, 36, 49, 62] acrossing different datasets. The second observation is that by combining multiple conv. layers, conv5-3, conv5-2, and conv5-1, denoted as  $\mathcal{M}_{\text{MAX}(\text{conv5-3,2,1})}$ , the proposed MAX scheme consistently boosts the retrieval accuracy. Our framework ( $\mathcal{M}_{\text{MAX}(\text{conv5-3,2,1})} + \phi_{\Delta} + \psi_{\text{d}}$ ) with dimensionality of 512 is competitive with other state-of-the-art methods [36, 62]. In particular, in comparison with CroW [36], while having slightly lower performances in *Oxford5k*, our method outperforms CroW on *Paris6k* and *Holidays*. In comparison with R-MAC [62], the proposed framework outperforms RMAC on *Oxford5k* and *Holidays* datasets, while it is comparable to RMAC on *Paris6k* dataset. Note that in some comparison methods, e.g., siaMAC [49], R-MAC [62], SPoC [4], CroW [36], the dimensionality is 256 or 512. This is due to the final representation dimensionality of these methods is upper bounded by the number of feature channels  $K$  of the selected network architecture and layers, e.g.,  $K = 512$  for conv5 of VGG16. Our proposed method, on the other hand, provides more flexibility in the representation dimensionality, thanks to the embedding process.

Table 8. Comparison with the state of the art when the final representations are real values. The results of compared methods are cited from the corresponding papers when available. For results of R-MAC[62] on Holidays and Holidays+Flickr1M, we use the released code of R-MAC[62] to evaluate on these datasets.

	Methods	Dim.	Size (Byte)	Datasets					
				<i>Oxf5k</i>	<i>Oxf105k</i>	<i>Par6k</i>	<i>Par106k</i>	<i>Hol</i>	<i>Hol+F11M</i>
SIFT	$\phi_{\Delta} + \psi_{\mathbf{d}}$ [35]	512	2k	52.8	46.1	-	-	61.7	46.9
	$\phi_{\Delta} + \psi_{\mathbf{d}}$ [35]	1024	4k	56.0	50.2	-	-	72.0	49.4
	$\phi_{\text{F-FAemb}} + \psi_{\mathbf{d}}$ [10]	512	2k	53.9	50.9	-	-	69.0	65.3
	$\phi_{\text{F-FAemb}} + \psi_{\mathbf{d}}$ [10]	1024	4k	58.2	53.2	-	-	70.8	68.5
Off-the-shelf network	SPoC [4]	256	1k	53.1	-	50.1	-	80.2	-
	MOP-CNN [16]	512	2k	-	-	-	-	78.4	-
	CroW [36]	512	2k	70.8	65.3	79.7	72.2	85.1	-
	MAC [49]	512	2k	56.4	47.8	72.3	58.0	76.7	-
	R-MAC [62]	512	2k	66.9	61.6	83.0	75.7	86.6	71.5
	NetVLAD [1]	1024	4k	62.6	-	73.3	-	87.3	-
	PWA [68]	1024	4k	75.3	69.3	84.2	78.2	-	-
	NetVLAD [1]	4096	16k	66.6	-	77.4	-	88.3	-
	$\mathcal{M}_{\text{SIFT}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	512	2k	64.4	59.4	79.5	70.6	86.5	-
	$\mathcal{M}_{\text{SUM}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	512	2k	64.0	58.8	78.6	70.4	86.4	-
	$\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	512	2k	65.7	60.5	81.6	72.4	85.0	71.9
	$\mathcal{M}_{\text{MAX}(\text{conv5-3,2,1})} + \phi_{\Delta} + \psi_{\mathbf{d}}$	512	2k	69.2	65.3	82.5	74.0	88.7	73.0
	$\mathcal{M}_{\text{SIFT}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	1024	4k	69.9	64.3	81.7	73.8	87.1	-
	$\mathcal{M}_{\text{SUM}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	1024	4k	70.8	64.4	80.6	73.8	86.9	-
$\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	1024	4k	72.2	67.9	83.2	76.1	88.4	79.1	
$\mathcal{M}_{\text{MAX}(\text{conv5-3,2,1})} + \phi_{\Delta} + \psi_{\mathbf{d}}$	1024	4k	74.8	70.4	84.5	78.6	90.8	81.7	
Finetuned network	siaMAC + MAC [49]	512	2k	79.7	73.9	82.4	74.6	79.5	-
	siaMAC + R-MAC [49]	512	2k	77.0	69.2	83.8	76.4	82.5	-
	NetVLAD★ [1]	1024	4k	69.2	-	76.5	-	86.5	-
	NetVLAD★ [1]	4096	16k	71.6	-	79.7	-	87.5	-
	siaMAC + $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	512	2k	77.7	72.7	83.2	76.5	86.3	-
	siaMAC + $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	1024	4k	81.4	77.4	84.8	78.9	88.9	82.1
	NetVLAD★ + $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	1024	4k	75.2	71.7	84.4	76.9	91.5	-
	NetVLAD★ + $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	4096	16k	78.2	75.7	87.8	81.8	92.2	-
siaMAC + $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$	4096	16k	83.8	80.6	88.3	83.1	90.1	-	

It is worth noting that NetVLAD [1], MOP-CNN [16], and PWA [68] methods can also produce higher dimensional representation by increasing the codebook size. However, as shown in Table 8 at comparable dimensions, the proposed framework clearly outperforms NetVLAD and MOP-CNN. In addition, at the dimensionality of 1024, our framework  $\mathcal{M}_{\text{MAX}(\text{conv5-3,2,1})} + \phi_{\Delta} + \psi_{\mathbf{d}}$  is slightly more favorable than PWA. Our framework achieves better performances on *Oxford105k*, *Paris6k*, and *Paris106k* datasets and is only slightly lower in mAP in *Oxford5k* dataset.

**Taking advantages of fine-tuned VGG networks.** Since our proposed framework takes the 3D activation tensor of a conv. layer as the input, it is totally compatible with deep networks which are fine-tuned for the image retrieval task such as siaMAC [49] and NetVLAD [1] (noted as NetVLAD★). In the “**Fine-tuned network**” section of Table 8, we evaluate our best framework –  $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$  in which the local conv. features of the fine-tuned VGG networks NetVLAD★ [1], siaMAC [49] are used as inputs.

The experimental results from Table 8 show that, for our  $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\mathbf{d}}$  framework, using conv. features from the siaMAC network usually gives better performance than using those from the fine-tuned NetVLAD★ network. When using local conv. features extracted from the fine-tuned siaMAC network [49], our method is competitive to **siaMAC + R-MAC** and **siaMAC + MAC** [49] at dimensionality of 512. At 1024 dimensions, our method consistently outperforms



Table 10. Comparison with the state of the art on very compact representations. (re.) and (bin.) mean that the representations are real values and binary values, respectively. For real values, the distance measure is cosine and for binary values, distance measure is Hamming.

Method	Dim.	Size (Byte)	Datasets				
			<i>Oxf5k</i>	<i>Oxf105k</i>	<i>Par6k</i>	<i>Par106k</i>	<i>Hol</i>
siaMAC + MAC [49]	16 (re.)	64	56.2	45.5	57.3	43.4	51.3
siaMAC + R-MAC [49]	16 (re.)	64	46.9	37.9	58.8	45.6	54.4
GeM [50]	16 (re.)	64	56.2	44.4	63.5	45.5	60.9
$\mathcal{M}_{\text{MAX}} + \phi_{\Delta(32,20)} + \psi_{\text{d}} + \text{ITQ}$	256 (bin.)	32	45.0	38.3	63.0	50.5	79.5
siaMAC + $\mathcal{M}_{\text{MAX}} + \phi_{\Delta(32,20)} + \psi_{\text{d}} + \text{ITQ}$	256 (bin.)	32	58.5	49.1	74.1	63.6	79.9
$\mathcal{M}_{\text{MAX}} + \phi_{\Delta(32,20)} + \psi_{\text{d}} + \text{ITQ}$	512 (bin.)	64	57.3	49.8	74.7	56.1	83.5
siaMAC + $\mathcal{M}_{\text{MAX}} + \phi_{\Delta(32,20)} + \psi_{\text{d}} + \text{ITQ}$	512 (bin.)	64	<b>68.9</b>	<b>60.9</b>	<b>79.1</b>	<b>70.3</b>	<b>83.6</b>

both siaMAC and NetVLAD $\star$  on all datasets. These considerable improvements indicate that our proposed framework can fully utilize the discrimination gain of local conv. features achieved by those fine-tuning networks.

**Very large scale image retrieval.** In order to verify the capabilities of our framework in real scenarios, we now evaluate it with a very large scale dataset, *Holidays+Flickr1M*. The experimental results show that the proposed framework “siaMAC +  $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\text{d}}$ ” is quite robust to the database size, i.e., when adding 1M distractor images to the Holidays dataset, the performance drop is only about 7%. We achieve a mAP of 82.1 which is significantly higher than 71.5 of R-MAC [62].

**Comparison to Selective Convolutional Descriptor Aggregation (SCDA) [66].** Recently, Wei et al. [66] proposed a method which selects deep features on *relu5\_2* and *pool\_5* layers from the pretrained VGG networks. Their method shares

Table 9. Comparison with SCDA [66]

Method	Dim.	mAP	
		<i>Oxford5k</i>	<i>Holidays</i>
SCDA[66]	4096 $\rightarrow$ 512	67.7	92.1
$\mathcal{M}_{\text{SUM}} + \phi_{\Delta(64,64)} + \psi_{\text{d}}$	4096 $\rightarrow$ 512	77.2	92.0
$\mathcal{M}_{\text{MAX}} + \phi_{\Delta(64,64)} + \psi_{\text{d}}$	4096 $\rightarrow$ 512	78.6	93.2

some similarities with our SUM-mask. Our work, however, is different from [66] in several important aspects, i.e., we propose and evaluate various masking schemes, i.e., SUM-mask, SIFT-mask, and MAX-mask. Our experiments show that the MAX-mask scheme consistently outperforms other schemes. In addition, we utilize state-of-the-art embedding and aggregating methods to enhance the discriminative power of the final representation. In order to have a complete evaluation to SCDA [66], we conduct comparison with SCDA [66] on *Oxford5k* and *Holidays* datasets. We exactly follow the setting of SCDA, i.e., the 5063 and 1491 gallery images of *Oxford5k* and *Holidays* datasets are used as the training set when learning codebooks for the embedding. In this experiment, for our methods, we do not truncate the first low frequency components when embedding. This makes the original dimension of the aggregated representations of SCDA and our methods are comparable, i.e., 4096. The low dimensionality, i.e. 512, is achieved by applying PCA. The comparative results in Table 9 clearly show the superior performances, especially on *Oxford5k* dataset, of our proposed framework over SCDA.

**5.3.2 Comparison with the state of the art when images are represented by very compact representations.** We now compare the quality of binary image representations producing by our framework with compact real-valued representations from state-of-the-art methods at comparable sizes (in Bytes) [49, 50]. Furthermore, ITQ [15] is used as the hashing function in our final framework since it gives competitive results (Section 5.2.5) and it is also computationally efficient in both training and producing new binary codes. We report the comparative results in Table 10.

Firstly, we can observe that at the same image descriptor size, e.g., 64 bytes, even when using off-the-shelf VGG [53], our framework significantly outperforms [49, 50] which use fine-tuned VGG networks. For examples, the proposed framework outperforms the second best GeM [50] large margins, i.e., +11.2% and +22.6% on *Paris6k* and *Holidays* datasets, respectively. Secondly, when using local conv. features of a fine-tuned VGG, e.g., siaMAC [49], our framework achieves significant extra improvements in retrieval performances over all datasets.

#### 5.4 Online processing time

We conduct experiments to empirically measure the on-line processing time of our proposed framework. We additionally compare our online processing time with one of the most competitive methods: R-MAC [62]. Both implementations of our framework and R-MAC are in Matlab. The experiments are executed on a workstation with a processor core (i7-6700 CPU @ 3.40GHz). Fig. 7 reports the averaged online processing time of *Oxford5k* dataset (5063 images). Note that the processing time includes the time to compute and apply masks and excludes the time for extracting 3D convolutional feature maps. The figure clearly shows that the MAX/SUM-mask can help to considerably reduce the computational cost of our proposed framework. As the proposed masking schemes can eliminate about 70% (for MAX-mask) and 50% (for SUM-mask) of local conv. features (Section 3.3). Furthermore, at 512-D, our framework  $\mathcal{M}_{\text{MAX/SUM}} + \phi_{\Delta} + \psi_d$  is faster than R-MAC [62].

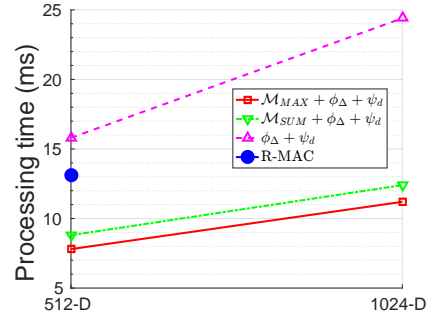


Fig. 7. The averaged online processing time of images in *Oxford5k* dataset.

## 6 CONCLUSION

In this paper, we present a novel, optimized and computationally-efficient framework for image retrieval task. The framework takes activations of a convolutional layer as the input and outputs a highly-discriminative image representation. In the framework, we propose to enhance discriminative power of the image representation in two main steps: (i) applying our proposed masking schemes, e.g., SIFT/SUM/MAX-mask, to select a subset of selective local conv. features, then (ii) employing the state-of-art embedding and aggregating methods [10, 35]. In order to make the final representations suitable for large scale search, we further compress the real-valued representation by cascading a hashing function into framework. We comprehensively evaluate and analyze each component in the framework to figure out the best configuration. Solid experimental results show that our proposed framework favorably compares with the state of the art for real-valued representations. Moreover, our binary representations are significantly outperforms the state-of-the-art methods at comparable sizes.

## REFERENCES

- [1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*.
- [2] Relja Arandjelović and Andrew Zisserman. 2012. Three things everyone should know to improve object retrieval. In *CVPR*.
- [3] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. 2015. From generic to specific deep representations for visual recognition. In *CVPR Workshops*.
- [4] Artem Babenko and Victor Lempitsky. 2015. Aggregating Local Deep Features for Image Retrieval. In *ICCV*.
- [5] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. 2014. Neural codes for image retrieval. In *ECCV*.

- [6] Jiewei Cao, Zi Huang, Peng Wang, Chao Li, Xiaoshuai Sun, and Heng Tao Shen. 2016. Quartet-net Learning for Visual Instance Retrieval. In *ACM MM*.
- [7] Miguel A. Carreira-Perpinan and Ramin Raziperchikolaei. 2015. Hashing With Binary Autoencoders. In *CVPR*.
- [8] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez. 2013. Revisiting the VLAD image representation. In *ACM MM*.
- [9] Thanh-Toan Do, Tuan Hoang, Dang-Khoa Le-Tan, Trung Pham, Huu Le, Ngai-Man Cheung, and Ian Reid. 2019. Binary Constrained Deep Hashing Network for Image Retrieval without Manual Annotation. In *WACV*.
- [10] Thanh-Toan Do and Ngai-Man Cheung. 2018. Embedding based on function approximation for large scale image search. *TPAMI* (2018).
- [11] Thanh-Toan Do, Anh-Dzung Doan, and Ngai-Man Cheung. 2016. Learning to hash with binary deep neural network. In *ECCV*.
- [12] Thanh-Toan Do, Dang-Khoa Le Tan, Trung T. Pham, and Ngai-Man Cheung. 2017. Simultaneous Feature Aggregating and Hashing for Large-Scale Image Search. In *CVPR*.
- [13] Matthijs Douze, Hervé Jégou, and Florent Perronnin. 2016. Polysemous Codes. In *ECCV*.
- [14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*.
- [15] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 2013. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval. *TPAMI* (2013), 2916–2929.
- [16] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. 2014. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*.
- [17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*.
- [18] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. 2016. Deep Image Retrieval: Learning Global Representations for Image Search. In *ECCV*.
- [19] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. 2017. End-to-End Learning of Deep Visual Representations for Image Retrieval. *IJCV* (2017).
- [20] Kristen Grauman and Rob Fergus. 2013. Learning Binary Hash Codes for Large-Scale Image Search. *Machine Learning for Computer Vision* (2013).
- [21] Bharath Hariharan, Pablo Arbeláñez, Ross Girshick, and Jitendra Malik. 2014. Hypercolumns for Object Segmentation and Fine-grained Localization. In *CVPR*.
- [22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. In *ICCV*.
- [23] Kaiming He, Fang Wen, and Jian Sun. 2013. K-Means Hashing: An Affinity-Preserving Quantization Method for Learning Binary Compact Codes. In *CVPR*.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015).
- [25] Tuan Hoang, Thanh-Toan Do, Huu Le, Dang-Khoa Le Tan, and Ngai-Man Cheung. 2018. Simultaneous Compression and Quantization: A Joint Approach for Efficient Unsupervised Hashing. (2018). <http://arxiv.org/abs/1802.06645>
- [26] Tuan Hoang, Thanh-Toan Do, Dang-Khoa Le Tan, and Ngai-Man Cheung. 2017. Selective Deep Convolutional Features for Image Retrieval. In *ACM-MM*.
- [27] Noh Hyeonwoo, Araujo Andre, Sim Jack, Weyand Tobias, and Han Bohyung. 2017. Large-Scale Image Retrieval with Attentive Deep Local Features. In *ICCV*.
- [28] Hervé Jégou and Ondřej Chum. 2012. Negative Evidences and Co-occurrences in Image Retrieval: The Benefit of PCA and Whitening. In *ECCV*.
- [29] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2008. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In *ECCV*.
- [30] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2009. On the burstiness of visual elements. In *CVPR*.
- [31] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2010. Improving Bag-of-Features for Large Scale Image Search. *IJCV* 87, 3 (May 2010), 316–336.
- [32] H. Jégou, M. Douze, and C. Schmid. 2011. Product Quantization for Nearest Neighbor Search. *TPAMI* (2011), 117–128.
- [33] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating local descriptors into a compact image representation. In *CVPR*.
- [34] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. 2012. Aggregating Local Image Descriptors into Compact Codes. *TPAMI* (2012), 1704–1716.
- [35] Hervé Jégou and Andrew Zisserman. 2014. Triangulation embedding and democratic aggregation for image search. In *CVPR*.
- [36] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-dimensional Weighting for Aggregated Deep Convolutional Features. In *ECCV Workshops*.

- [37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.
- [38] Ying Li, Xiangwei Kong, Liang Zheng, and Qi Tian. 2016. Exploiting Hierarchical Activations of Neural Network for Image Retrieval. In *ACM MM*.
- [39] Guosheng Lin, Chunhua Shen, Ian D. Reid, and Anton van den Hengel. 2015. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*.
- [40] Z. Liu, S. Wang, L. Zheng, and Q. Tian. 2017. Robust ImageGraph: Rank-Level Feature Fusion for Image Search. *TIP* 26, 7 (July 2017), 3128–3141.
- [41] David G. Lowe. 1999. Object Recognition from Local Scale-Invariant Features. In *ICCV*.
- [42] Naila Murray and Florent Perronnin. 2014. Generalized Max Pooling. In *CVPR*.
- [43] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised Learning of Visual Representations by solving Jigsaw Puzzles. In *ECCV*.
- [44] Aude Oliva and Antonio Torralba. 2001. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *IJCV* (2001), 145–175.
- [45] Florent Perronnin and Christopher Dance. 2007. Fisher Kernels on Visual Vocabularies for Image Categorization. In *CVPR*.
- [46] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. 2010. Improving the fisher kernel for large-scale image classification. In *ECCV*.
- [47] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*.
- [48] James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*.
- [49] Filip Radenović, Giorgos Tolias, and Ondřej Chum. 2016. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. In *ECCV*.
- [50] Filip Radenović, Giorgos Tolias, and Ondřej Chum. 2017. Fine-tuning CNN Image Retrieval with No Human Annotation. In *arXiv:1711.02512*.
- [51] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *CVPRW*.
- [52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*.
- [53] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- [54] Josef Sivic, Andrew Zisserman, et al. 2003. Video Google: a text retrieval approach to object matching in videos. In *ICCV*.
- [55] Jingkuan Song, Lianli Gao, Li Liu, Xiaofeng Zhu, and Nicu Sebe. 2018. Quantization-based hashing: a general framework for scalable image and video retrieval. *Pattern Recognition* 75 (2018), 175 – 187.
- [56] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, Alan Hanjalic, and Heng Tao Shen. 2018. Binary Generative Adversarial Networks for Image Retrieval. In *AAAI*.
- [57] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, and Heng Tao Shen. 2018. Deep Region Hashing for Efficient Large-scale Instance Search from Images. (2018).
- [58] J. Song, H. Zhang, X. Li, L. Gao, M. Wang, and R. Hong. 2018. Self-Supervised Video Hashing With Hierarchical Binary Auto-Encoder. *IEEE TIP* 27, 7 (2018), 3210–3221.
- [59] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*.
- [60] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.
- [61] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. 2013. To Aggregate or Not to aggregate: Selective Match Kernels for Image Search. In *ICCV*.
- [62] Giorgos Tolias, Ronan Sifre, and Hervé Jégou. 2016. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*.
- [63] Andrea Vedaldi and Brian Fulkerson. 2010. VLFeat: An Open and Portable Library of Computer Vision Algorithms. In *ACM-MM*.
- [64] Andrea Vedaldi and Karel Lenc. 2014. MatConvNet - Convolutional Neural Networks for MATLAB. *CoRR* abs/1412.4564 (2014). <http://arxiv.org/abs/1412.4564>
- [65] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen. 2017. A Survey on Learning to Hash. *TPAMI* (2017).
- [66] X. S. Wei, J. H. Luo, J. Wu, and Z. H. Zhou. 2017. Selective Convolutional Descriptor Aggregation for Fine-Grained Image Retrieval. *TIP* 26 (June 2017), 2868–2881.

- [67] Yair Weiss, Antonio Torralba, and Robert Fergus. 2008. Spectral Hashing. In *NIPS*.
- [68] Jian Xu, Cunzhaoh Shi, Chengzuo Qi, Chunheng Wang, and Baihua Xiao. 2018. Unsupervised Part-Based Weighting Aggregation of Deep Convolutional Features for Image Retrieval. In *AAAL*.
- [69] Ke Yan, Yaowei Wang, Dawei Liang, Tiejun Huang, and Yonghong Tian. 2016. CNN vs. SIFT for Image Retrieval: Alternative or Complementary?. In *ACM MM*.
- [70] Matthew D. Zeiler and Rob Fergus. 2013. Visualizing and Understanding Convolutional Networks. *CoRR* abs/1311.2901 (2013). <http://arxiv.org/abs/1311.2901>
- [71] Y. Zhang, J. Wu, and J. Cai. 2016. Compact Representation of High-Dimensional Feature Vectors for Large-Scale Image Recognition and Retrieval. *TIP* 25, 5 (May 2016).
- [72] Liang Zheng, Yi Yang, and Qi Tian. 2016. SIFT Meets CNN: A Decade Survey of Instance Retrieval. *TPAMI* (08 2016).