Selective Deep Convolutional Features for Image Retrieval

Tuan Hoang
Singapore University of Technology and Design
nguyenanhtuan_hoang@mymail.sutd.edu.sg

Thanh-Toan Do
The University of Adelaide
thanh-toan.do@adelaide.edu.au

Dang-Khoa Le Tan
Singapore University of Technology and Design
letandang_khoa@sutd.edu.sg

Ngai-Man Cheung
Singapore University of Technology and Design
ngaiman_cheung@sutd.edu.sg

ABSTRACT

Convolutional Neural Network (CNN) is a very powerful approach to extract discriminative local descriptors for effective image search. Recent work adopts fine-tuned strategies to further improve the discriminative power of the descriptors. Taking a different approach, in this paper, we propose a novel framework to achieve competitive retrieval performance. Firstly, we propose various masking schemes, namely ***SIFT-mask***, ***SUM-mask***, and ***MAX-mask***, to select a representative subset of local convolutional features and remove a large number of redundant features. We demonstrate that this can effectively address the burstiness issue and improve retrieval accuracy. Secondly, we propose to employ recent embedding and aggregating methods to further enhance feature discriminability. Extensive experiments demonstrate that our proposed framework achieves state-of-the-art retrieval accuracy.

CCS CONCEPTS

• **Computing methodologies → Image representations**;

KEYWORDS

Content Based Image Retrieval, Embedding, Aggregating, Deep Convolutional Features, Unsupervised

ACM Reference format:
Tuan Hoang, Thanh-Toan Do, Dang-Khoa Le Tan, and Ngai-Man Cheung. 2017. Selective Deep Convolutional Features for Image Retrieval. In *Proceedings of ACM Multimedia conference, Mountain View, CA USA, October 23-27, 2017 (MM'17),* 9 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Content-based image retrieval (CBIR) has attracted a sustained attention from the multimedia/computer vision community 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 [25] and its variant [2]. Even though there are great improvements of the SIFT-based image search systems over time, the performance of these systems still has room for improvement. There are two main issues: the first and the most important one is that SIFT features lack discriminability [4] to emphasize the differences in images. Even though this drawback is relieved to some extent when embedding local features to much higher dimensional space [9, 12, 20, 21, 27, 35], there is still a large semantic gap between SIFT-based image representation and human perception on instances (objects/scenes) [4]. Secondly, the strong effect of *burstiness* [18], i.e. numerous descriptors are almost similar within the same image, considerably degrade the quality of SIFT-based image representation for the image retrieval task [8, 18, 20].

Recently, deep Convolutional Neural Networks (CNN) have achieved a lot of success in various problems including image classification [16, 23, 34, 36], object detection [13, 32], etc. After training a CNN on a huge annotated dataset, e.g. ImageNet [33], outputs of middle/last layers can capture rich information at higher semantic levels. On one hand, the output of the deeper layer possesses abstract understanding of images for computer vision tasks that require high-invariance to the intra-class variability, e.g., classification, detection [13, 16, 23, 32, 34, 36]. On the other hand, the middle layers contain more visual information on edges, corners, patterns, and structures. Therefore, they are more suitable for image retrieval [1, 4, 22, 24, 39]. Utilizing the outputs of the convolutional layers to produce the image representation, recent image retrieval methods [1, 4, 22, 24, 39] achieve a considerable performance boost.

Although the local convolutional (conv.) features are more discriminative than SIFT features [4], to the best of our knowledge, none of the previous works has considered the burstiness problem which appears in the local features. In this paper, focusing on CNN based image retrieval, we delve deeper into the issue: *"How to eliminate redundant local features in a robust way?"* Since elimination of redundant local features leads to better representation and faster computation, we emphasize both aspects in our experiments. Specifically, inspired by the concept of finding a set of interest regions before deriving their corresponding local features - the concept which has been used in design of hand-crafted features, we propose three different masking schemes for selecting *representative* local conv. features, including ***SIFT-mask***, ***SUM-mask***, and ***MAX-mask***. The principal ideas of **our main contribution** are that we take advantages of SIFT detector [25] to produce SIFT-mask; moreover, we utilize sum-pooling and max-pooling over all conv. feature channels to derive SUM-mask and MAX-mask, respectively.

Additionally, most of the recent works which take local conv. features as input [22, 31, 39] do not leverage local feature embedding and aggregating [12, 20, 21, 27], which are effective processes to enhance the discriminability for hand-crafted features. In [4], the authors mentioned that the deep convolutional features are already

MM'17, Mountain View, CA USA

DOI: 10.1145/nnnnnnn.nnnnnnn

discriminative enough for image retrieval task, hence, the embedding is not necessary to enhance their discriminability. However, in this work, we find that by utilizing state-of-art embedding methods on our selected deep convolutional features [12, 20, 21, 27], we can significantly enhance the discriminability. Our experiments show that applying embedding and aggregating on our selected local conv. features significantly improves image retrieval accuracy.

The remaining of this paper is organized as follows. Section 2 discusses related works. Section 3 presents the details of our main contribution, the masking schemes, together with preliminary experimental results to justify their effectiveness. In section 4, we introduce the proposed framework for computing the final image representation which takes selected local deep conv. features as input and output a global fixed-length image representation. Section 5 presents a wide range of experiments to comprehensively evaluate our proposed framework. Section 6 concludes the paper.

2 RELATED WORK

In the task of image retrieval, the early CNN-based work [5] takes the activation of fully connected layers as global descriptors followed by dimensionality reduction. This work shows that supervised retraining the network on the dataset which is relevant to the test set is very beneficial in the retrieval task. However, as shown in [5], the creation of labeled training data is expensive and non-trivial. Gong et al. [14] proposed Multi-Scale Orderless Pooling (MOP) to embed and pool the CNN fully-connected activations of image patches of an image at different scale levels. This enhances the scale invariant of the extracted features. However, the method is computationally expensive because multiple patches (resized to the same size) of an image are fed forward into the CNN. The recent work [42] suggested that CNN fully-connected activations and SIFT features are highly complementary. They proposed to integrate SIFT features with fully-connected CNN features at different levels.

Later works shift the focus from fully-connected layers to conv. layers for extracting image features because lower layers are more general and certain level of spatial information is still preserved [3]. When conv. layers are used, the conv. features are usually considered as local features, hence, a pooling method (sum or max) is applied on the conv. features to produce the single image representation. Babenko and Lempitsky [4] showed that sum-pooling outperforms max-pooling when the final image representation is whitened. Kalantidis et al. [22] further improved sum-pooling on conv. features by proposing a non-parametric method to learn weights for both spatial locations and feature channels. Tolias et al. [39] revisited max-pooling by proposing the strategy to aggregate the maximum activation over multiple spatial regions sampled on the final conv. layer using a *fixed layout*. This work together with [22] are currently the state-of-art methods in image retrieval task using off-the-shelf CNN.

Although fine-tuning an off-the-shelf network (e.g. AlexNet or VGG) can enhance the discriminability of the deep features [5] for image retrieval, the collecting of training data is non-trivial. Recent works tried to overcome this challenge by proposing unsupervised/weak supervised fine-tuning approaches which are specific for image retrieval. Arandjelović et al. [1] proposed a new generalized VLAD layer and this layer can be stacked with any CNN architecture. The whole architecture, named NetVLAD, is trained in an end-to-end manner in which the data is collected in a weakly supervised manner from Google Street View Time Machine. Also taking the approach of fine-tuning the network in a weakly-supervised manner, Cao et al. [7] proposed an automatic method to harvest data from GeoPair dataset [37] to train a special architecture called Quartet-net with the novel double margin contrastive loss function. Concurrently, Radenović et al. [31] proposed a different approach to re-train state-of-the-art CNNs of classification task for image retrieval. They take advantages of 3D reconstruction to obtain matching/non-matching pairs of images in an unsupervised manner for re-training process.

3 SELECTIVE LOCAL DEEP CONV. FEATURES

In this section, we first define the set of local deep conv. features which we work on throughout the paper (Section 3.1). We then present proposed strategies for selecting a subset of discriminative local conv. features, including ***SIFT mask***, ***SUM mask***, and **MAX mask** (Section 3.2). Finally, we discuss experiments to illustrate the effectiveness of our methods (Section 3.3).

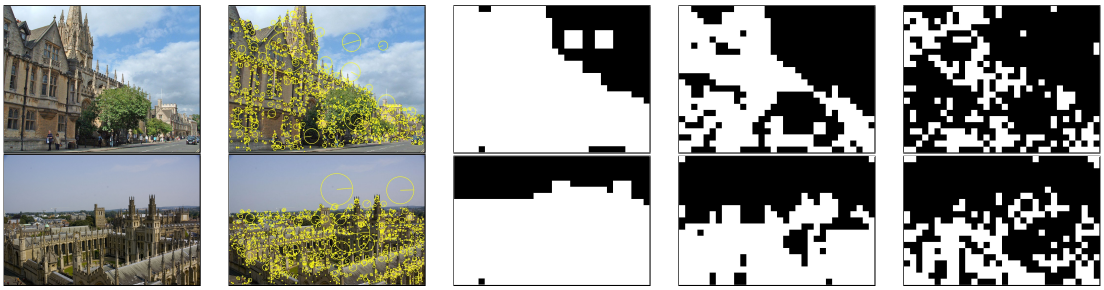
3.1 Local deep convolutional features

We consider a pre-trained CNN with all the fully connected layers discarded. Given an input image I of size $W_I \times H_I$ that is fed through a CNN, the 3D activations (responses) tensor of a conv. layer has the size of $W \times H \times K$ dimensions, where K is the number feature maps and $W \times H$ is the spatial resolution of a feature map. We consider this 3D tensor of responses as a set of $(W \times H)$ local features; each of them have K dimensions. In other words, each position on the $W \times H$ spatial grid is the location of a local feature. Each local conv. feature is a vector of K values of the K feature maps at a particular location. We denote $\mathcal{F}^{(k)}$ as k^{th} feature map (and its size is $W \times H$). Note that the choice of the conv. layer to be used is not fixed in our method. We investigate the impact of choosing different conv. layers in Section 5.

3.2 Selective features

We now formally propose different methods to compute a selection mask, i.e. a set of unique coordinates $\{(x, y)\}$ in the feature maps where local conv. features are retained ($1 \leq x \leq W; 1 \leq y \leq H$). Our proposed methods for selecting discriminative local deep conv. features are inspired by the concept of finding the interest regions in the input images which is traditionally used in the design of hand-crafted features.

3.2.1 SIFT Mask. Prior the emergence of CNN features in the image retrieval task, most previous works [8, 12, 17, 18, 20, 21, 27, 38] are based on SIFT [25] features and its variant RootSIFT [2]. Even though it has been showed that there is still a gap between SIFT-based representation and the semantic meaning in the image, these works have clearly demonstrated the capability of SIFT feature, especially in the aspect of key-point detection. Figure (1b) shows local image regions which are covered by SIFT. We can obverse that regions covered by SIFT mainly focus on the salient regions, i.e., buildings. This means that SIFT keypoint detector is capable to locate important regions of images. Hence, we propose a method



(a) Original image (b) Image+SIFT (c) SIFT-mask (d) SUM-mask (e) MAX-mask

Figure 1: Examples of masks to select local features. The original images are showed on the first column (1a). The second column shows regions which are covered by SIFT features. The SIFT/SUM/MAX-masks of corresponding images are showed in the last three columns (1c,1d,1e).

which takes advantage of SIFT detector in combination with highly-semantic local deep conv. features.

Specifically, let set $\mathcal{S} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ be SIFT feature locations extracted from an image with the size of $W_I \times H_I$; each location on the spatial grid $W \times H$ is location of a local deep conv. feature. Based on the fact that convolutional layers still preserve the spatial information of the input image [39], we select a subset of locations on the spatial grid which correspond to locations of SIFT key-points, i.e.,

$$\mathcal{M}_{\text{SIFT}} = \left\{\left(x^{(i)}_{\text{SIFT}}, y^{(i)}_{\text{SIFT}}\right)\right\} \qquad i = 1, \cdots, n \tag{1}$$

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

Note that SIFT detector has the issue of burstiness [18]. However, regarding local conv. features, this burstiness effect is expected to be less severe since local conv. features have much larger receptive fields than those of SIFT features. For examples, a local conv. feature from pool5 layers of AlexNet [23] and VGG16 [34] covers a region of 195×195 and 212×212 in the input image, respectively.

3.2.2 MAX Mask. It is known that each feature map contains the activations of a specific visual structure [13, 44]. Hence, we propose to select a subset of local conv. features which contain high activations for all visual contents, i.e. we select the local features that capture the most prominent structures in the input images. This property, actually, is desirable to distinguish scenes.

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

$$\begin{aligned} \mathcal{M}_{\text{MAX}} &= \left\{\left(x^{(k)}_{\text{MAX}}, y^{(k)}_{\text{MAX}}\right)\right\} \qquad k = 1, \cdots, K \\ \left(x^{(k)}_{\text{MAX}}, y^{(k)}_{\text{MAX}}\right) &= \arg\max_{(x,y)} \mathcal{F}^{(k)}_{(x,y)} \end{aligned} \tag{2}$$

3.2.3 SUM Mask. Departing from the MAX-mask idea, we propose a different masking method based on the idea 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 that have large values of sum, we can expect that those local conv. features contain a lot of information from different local image structures [44]. Formally, we define the selected locations $\mathcal{M}_{\text{SUM}}$ as follows:

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

3.3 Effectiveness of masking schemes

In this section, we evaluate the effectiveness of our proposed masking schemes in eliminating redundant local conv. features. Firstly, Figure 2a shows the averaged percentage of the remaining local conv. features after applying our proposed masks on three datasets: *Oxford5k* [29], *Paris6k* [30], and *Holidays* [19]. Clearly, there are a large number of local conv. features removed, about 25%, 50%, and 70% for SIFT/SUM/MAX-mask respectively[1]. Additionally, we present the normalized histograms of covariances of selected local conv. features after applying different masks in Figure 2b, 2c, and 2d. To compute the covariances, we first $l2$-normalize local conv. features, which are extracted from pool5 layer of the pre-trained VGG [34] (the input image is of size $\max(W_I, H_I) = 1024$). We then compute the dot products for all pairs of features. For comparison, we include the normalized histograms of covariances of all available local conv. features (i.e., before masking). These figures clearly show 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 some reduction of correlation between the features with the use of mask. Furthermore, Figure 2e shows the averaged percentage of $l2$-normalized feature pairs that have dot products in the range of $[-0.15, 0.15]$. The chart shows that the selected features are more uncorrelated. In summary, Figure 2 suggests that our proposed masking schemes can help to remove a large proportion of redundant local conv. features, hence achieving a better representative set of local conv. features. Note that with the reduced number of features, we can reduce computational cost, e.g. embedding of features in the subsequent step.

[1] With the input image sizes of $\max(W_I, H_I) = 1024$.

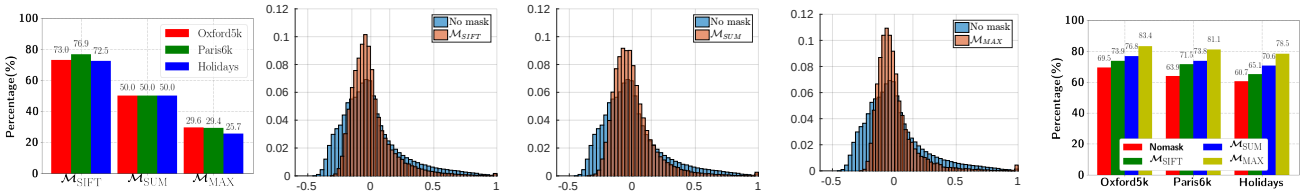


Figure 2: Fig. 2a: The averaged percentage of remaining local conv. features after applying masks. Fig. 2b, 2c, 2d: Examples of normalized histograms of covariances of sets of local conv. features (from the input image of the first row in Fig. 1) with/without applying masks. Fig. 2e: The averaged percentage of the covariance values in the range of $[-0.15, 0.15]$.

4 FRAMEWORK: EMBEDDING AND AGGREGATING ON SELECTIVE CONV. FEATURES

4.1 Pre-processing

Given a set $\mathcal{X} = \{\mathbf{x}_{(x,y)} \mid (x,y) \in \mathcal{M}_*\}$, 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 smaller dimension d: $\mathbf{x}^{(d)} = M_{\text{PCA}}\mathbf{x}$, where M_{PCA} is the PCA-matrix. There are two reasons for this dimensional reduction operation. Firstly, the lower dimensional local features helps to produce compact final image representation (even applying embedding) as the current trend in image retrieval [4, 31, 39]. Secondly, applying PCA could help to remove noise and redundancy; hence, enhancing the discrimination. We subsequently $l2$-normalize the compressed local conv. features.

4.2 Embedding

In this section, we aim to enhance the discriminability of selected local conv. features. We propose to accomplish this by embedding the conv. features to higher-dimensional space: $\mathbf{x} \mapsto \phi(\mathbf{x})$, using state-of-the-art embedding methods [9, 20, 21, 27]. It is worth noting that while in [4], the authors avoid applying embedding on the *original* set of local deep conv. features. However, we find that applying the embedding on the set of *selected* features significantly improves their discriminability.

We brief describle embedding methods used in our work, i.e. Fisher Vector (FV) [27], VLAD [20], Temb [21], F-FAemb [9]. Note that in the original design of FV and VLAD, the embedding and the aggregation (i.e., sum aggregation) are integrated. This prevents the using of recent state-of-the-art aggregation (i.e., democratic pooling [21]) on the embedded vectors produced by FV, VLAD. Hence, in order to make the embedding and the aggregating flexible, we decompose the formulation of VLAD and FV. Specifically, we apply the embedding on each local feature separately. This allows different aggregating methods to be applied on the embedded features.

For clarity, we pre-define the codebook of visual words learning by Gaussian Mixture Model used in Fisher Vector method as $C_{\text{G}} = \{\mu_i; \Sigma_i; w_i\}_{i=1}^{k}$, where w_i, μ_i and Σ_i denote respectively the weight, mean vector and covariance matrix of the i-th Gaussian. Similarly, the codebook learning by K-means used in VLAD, T-emb, and F-FAemb methods are defined as $C_{\mathbf{K}} = \{c_j\}_{j=1}^{k}$, where c_j is a centroid.

Fisher Vector (FV) produces a high-dimensional vector representation of $(2 \times k \times d)$-dimension when considering both 1-st and 2-nd order statistic of the local features.

$$\phi_{\text{FV}}(\mathbf{x}) = \left[\cdots, u_i^T, \cdots, v_i^T, \cdots\right]^T \qquad i = 1, \cdots, k$$
$$u_i = \frac{p_i(\mathbf{x})}{\sqrt{w_i}}\left(\frac{\mathbf{x} - \mu_i}{\sigma_i}\right) \qquad v_i = \frac{p_i(\mathbf{x})}{\sqrt{2w_i}}\left[\left(\frac{\mathbf{x} - \mu_i}{\sigma_i}\right)^2 - 1\right] \tag{4}$$

Where $p_i(\mathbf{x})$ is the posterior probability capturing the strength of relationship between a sample $\mathbf{x}$ and the i-th Gaussian model and $\sigma_i = \sqrt{\text{diag}(\Sigma_i)}$.

VLAD [20] is considered as a simplification of the FV. It embeds $\mathbf{x}$ to the feature space of $(d \times k)$-dimension.

$$\phi_{\text{VLAD}}(\mathbf{x}) = [\cdots, q_i(\mathbf{x} - c_i)^T, \cdots]^T \qquad i = 1, \cdots, k \tag{5}$$

Where c_i is the i-th visual word of the codebook $C_{\mathbf{K}}$, $q_i = 1$ if c_i is the nearest visual word of $\mathbf{x}$ and $q_i = 0$ otherwise.

T-emb [21]. Different from FV and VLAD, T-emb avoids the dependency on absolute distances by only preserve direction information between a feature $\mathbf{x}$ and visual words $c_i \in C_{\mathbf{K}}$.

$$\phi_{\Delta}(\mathbf{x}) = \left[\cdots, \left(\frac{\mathbf{x} - c_i}{\|\mathbf{x} - c_i\|}\right)^T, \cdots\right]^T \qquad i = 1, \cdots, k \tag{6}$$

F-FAemb [9]. Departing from the idea of linearly approximation of non-linear function in high dimensional space, the authors showed that the resulted embedded vector of the approximation process is the generalization of several well-known embedding methods such as VLAD [20], TLCC [43], VLAT [26].

$$s_i = \gamma_i(\mathbf{x}) V\left((\mathbf{x} - c_i)(\mathbf{x} - c_i)^T\right) \qquad i = 1, \cdots, k \tag{7}$$

where $\gamma_i(\mathbf{x})$ is coefficient corresponding to visual word c_i achieved by the function approximation process and $V(H)$ is a function that flattens the matrix to a vector. The vectors s_i are concatenated to form the single embedded feature $\phi_{\text{F-FAemb}}$.

4.3 Aggregating

Let $\mathcal{X}_\phi = \{\phi(\mathbf{x}_i)\}$ be a set of embedded local descriptors. Sum/average-pooling and max-pooling are two common methods for aggregating this set to a single global feature of length D.

When using the features generating from the activation function, e.g. ReLU [23], of a CNN, ***sum/average-pooling*** ($\psi_{\text{s}}/\psi_{\text{a}}$) lack discriminability because they average the high activated outputs by non-active outputs. Consequently, they weaken the effect of highly activated features. ***Max-pooling*** (ψ_{m}), on the other hand, is more preferable since it only retains the high activation for each visual content. However, it is worth noting that in practical, the

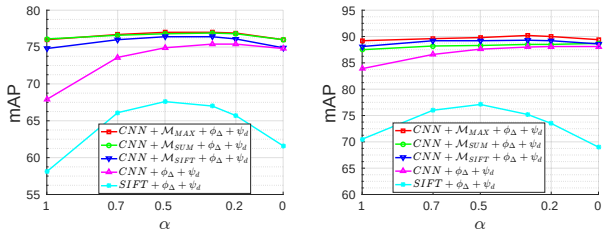
max-pooling is only successfully applied when features are sparse [6]. For examples, in [31, 39], the max-pooling is applied on each feature map because there are few of high activation values in a feature map. When the embedding is applied to embed local features to high dimensional space, the max-pooling may be failed since the local features are no longer sparse [6].

Recently, H. Jegou et. al. [21] introduced ***democratic aggregation*** (ψ_{d}) method applied to image retrieval problem. Democratic aggregation can work out-of-the-box with various embedded features such as VLAD [20], Fisher vector [27], T-emb [21], FAemb [12], F-FAemb [9], and it has been shown to outperform sum-pooling in term of retrieval performance with embedded hand-crafted SIFT features [21]. We also conduct experiments for this method on our framework.

4.4 Post-processing

Power-law normalization (PN). The *burstiness* of visual elements [18] is known as a major drawback of hand-crafted local descriptors, e.g. SIFT [25], such that numerous descriptors are almost similar within the same image. As a result, this phenomenon strongly affects the measure of similarity between two images. By applying power-law normalization [28] to the final image representation ψ and subsequently $l2$-normalization, it has been shown to be an efficient way to reduce the effect of burstiness [21]. The power-law normalization formula is given as $PN(x) = \mathrm{sign}(x)|x^{\alpha}|$, where $0 \leq \alpha \leq 1$ is a constant [28].

However, to the best of our knowledge, no previous work has re-evaluated the *burstiness* phenomena on the local conv. features. Figure 3 shows the analysis of PN effect on local conv. features using various masking schemes. This figure shows that the local conv. features ($CNN + \phi_{\Delta} + \psi_{\mathrm{d}}$) are still affected by the burstiness: the retrieval performance changes when applying PN. The figure also shows that the burstiness has much stronger effect on SIFT features ($SIFT + \phi_{\Delta} + \psi_{\mathrm{d}}$) than conv. features. The proposed SIFT, SUM and MAX masks help reduce the burstiness effect significantly: the PN has less effect on $CNN + \mathcal{M}_{\mathrm{MAX/SUM/SIFT}} + \phi_{\Delta} + \psi_{\mathrm{d}}$ than on $CNN + \phi_{\Delta} + \psi_{\mathrm{d}}$. This illustrates the capability of removing redundant local features of our proposed masking schemes. Similar to previous work, we set $\alpha = 0.5$ in our later experiments.



(a) *Oxford5k* **(b) *Holidays***

Figure 3: Impact of power-law normalization factor α on *Oxford5k* and *Holidays* datasets. Following the setting in [21], 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 [34].

Rotation normalization and dimension reduction. The power-law normalization suppresses visual burstiness but not frequent co-occurrences which also corrupt the similarity measure [17]. In order to reduce the effect of co-occurrences, we follow [17, 21] to rotate data with a whitening matrix learned from the aggregated vectors of the training set. The rotated vectors are used as the final image representations in our image retrieval system.

5 EXPERIMENTS

In this section, we will conduct comprehensive experiments to evaluate our proposed framework on three standard image retrieval benchmark datasets, including INRIA Holidays [19], Oxford Buildings dataset [29], and Paris dataset [30].

5.1 Datasets, Evaluation protocols, and Implementation notes

The **INRIA Holidays dataset** (*Holidays*) [19] contains 1491 vacation snapshots corresponding to 500 groups of the same instances. The query image set consists of one image from each group. We also manually rotate images (by ±90 degrees) to fix the wrong image orientation as in [4, 5, 22].

The **Oxford Buildings dataset** (*Oxford5k*) [29] contains 5063 photographs from Flickr associated with Oxford landmarks. 55 queries corresponding to 11 buildings/landmarks are fixed, and the ground truth relevance of the remaining dataset w.r.t. these 11 classes is provided. Following the standard protocol [15, 39], we use the cropped query images based on provided bounding boxes.

The **Paris dataset** (*Paris6k*) [30] are composed of 6412 images of famous landmarks in Paris. Similar to *Oxford5k*, this dataset has 55 queries corresponding to 11 buildings/landmarks. We also use provided bounding boxes to crop the query images accordingly.

Larger datasets. We additionally use 100k Flickr images [29] in combination with *Oxford5k* and *Paris6k* to compose *Oxford105k* and *Paris106k*, respectively. This 100k distractors are to allow evaluating retrieval methods at larger scale.

Evaluation protocols. The retrieval performance is reported as mean average precision (***mAP***) over query sets for all datasets. In addition, the *junk* images, which are defined as unclear to be relevant or not, are removed from the ranking.

Implementation notes. In the image retrieval task, it is important to use held-out datasets to learn all necessary parameters as to avoid overfitting [4, 15, 31]. In particular, the set of 5000 Flickr images[2] is used as the held-out dataset to learn parameters for *Holidays*. Similarly, *Oxford5k* is used for *Paris6k* and *Paris106k*, and *Paris6k* for *Oxford5k* and *Oxford105k*.

All images are resized so that the maximum dimension is 1024 while preserving aspect ratios before fed into the CNN. Additionally, as the common practice in recent works [4, 15, 31, 39], the pretrained VGG16 [34] (with Matconvnet [41] toolbox) is used to extract deep convolutional features. We utilize the VLFeat toolbox [40] for SIFT detector [3]. Additionally, in the rare case of an image with no SIFT feature, the SIFT-mask is ignored and we apply embedding and aggregating for all local features. We summarize the notations in Table 1.

[2] We randomly select 5000 images from the 100.071 Flickr image set [29].

[3] Note that VLFeat toolbox combines both SIFT detector and extractor in a single built-in function.

Table 1: Notations and their corresponding meanings. $\mathcal{M}, \phi, \psi$ denote masking, pooling and embedding respectively.

Notation	Meaning	Notation	Meaning
$\mathcal{M}_{\text{SIFT}}$	SIFT-mask	ψ_{a}	Average-pooling
$\mathcal{M}_{\text{SUM}}$	SUM-mask	ψ_{s}	Sum-pooling
$\mathcal{M}_{\text{MAX}}$	MAX-mask	ψ_{d}	Democratic-pooling [21]
ϕ_{FV}	FV [27]	ϕ_{VLAD}	VLAD [20]
ϕ_{Δ}	T-emb [21]	$\phi_{\text{F-FAemb}}$	F-FAemb [9]

5.2 Effects of parameters

5.2.1 Framework. In this section, we will conduct experiment to comprehensively compare various embedding and aggregating frameworks in combination with different proposed masking schemes. To make a fair comparison, we empirically set the retained PCA components-d and size of the visual codebooks-$|C|$ so as to produce the same final feature dimensionality-D as mentioned in Table 2. Note that, as proposed in original papers, F-FAemb [9] method requires to remove first $d(d+1)/2$ components of the features after aggregating step (Section 4.3). However, we empirically found that truncating the first $d(d+1)$ components generally achieves better performances.

Table 2: Configuration of different embedding methods.

Method	**PCA-d**	$\lvert C\rvert$	D
FV [27]	48	44	$2 \times d \times \lvert C\rvert = 4224$
VLAD [20]	64	66	$d \times \lvert C\rvert = 4224$
T-emb [21]	64	68	$d \times \lvert C\rvert - 128 = 4224$
F-FAemb [9]	32	10	$\frac{(\lvert C\rvert - 2) \times d \times (d+1)}{2} = 4224$

The comparison results on *Oxford5k*, *Paris6k*, and *Holidays* datasets are reported in Table 3. Firstly, we can observe that the democratic pooling [21] clearly outperforms sum/average-pooling on both FV [27] and VLAD [20] embedding methods. Secondly, our proposed masking schemes help to achieve considerable gains in performance across the variety of embedding and aggregating frameworks. Additionally, the MAX-mask generally provides the higher performance boosts than the SUM/SIFT-mask, while SUM-mask and SIFT-mask give comparable results. At the comparison dimensionality-$D = 4224$, the framework of $\phi_{\Delta} + \psi_{\text{d}}$ and $\phi_{\text{F-FAemb}} + \psi_{\text{d}}$ achieves comparable performances across various masking schemes and datasets. In this paper, since $\mathcal{M}_{\text{MAX}} + \phi_{\Delta} + \psi_{\text{d}}$ provides the best performance, slightly better than $\mathcal{M}_{\text{MAX}} + \phi_{\text{F-FAemb}} + \psi_{\text{d}}$, we choose $\mathcal{M}_{*} + \phi_{\Delta} + \psi_{\text{d}}$ as our default framework.

5.2.2 Final feature dimensionality. Different from recent works using convolutional features [4, 22, 31, 39], which have the final feature dimensionality upper bounded by the number of output feature channel K of network architecture and selected layer, e.g. $K = 512$ for `Conv5` of VGG [34]. Taking the advantages of embedding methods, similar to NetVLAD [1], our proposed framework provides more flexibility on choosing the length of final representation.

Considering our default framework - $\mathcal{M}_{*} + \phi_{\Delta} + \psi_{\text{d}}$, we empirically set the number of retained PCA components and the codebook size when varying the dimensionality in Table 4. For compact final representations, we choose $d = 32$ to avoid using too few visual words since this drastically degrades performance [21]. For longer final representations, imitating the setting for SIFT in [21], we reduce local conv. features to $d = 128$ and set $|C| = 64$. Note that the settings in Table 4 are applied for all later experiments.

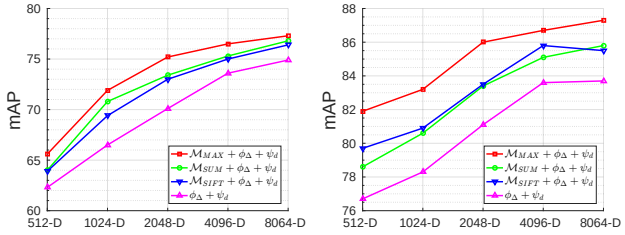
Table 3: Comparison of different frameworks. The "Bold" values indicates the best performance in each masking method and the "Underline" values indicates best performance across all settings.

	Method	$\mathcal{M}_{\text{MAX}}$	$\mathcal{M}_{\text{SUM}}$	$\mathcal{M}_{\text{SIFT}}$	None
Oxford5k	$\phi_{\text{FV}} + \psi_{\text{a}}$	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
	$\phi_{\Delta} + \psi_{\text{d}}$	**75.8**	**75.7**	**75.3**	73.4
	$\phi_{\text{F-FAemb}} + \psi_{\text{d}}$	75.2	74.7	74.4	**73.8**
Paris6k	$\phi_{\text{FV}} + \psi_{\text{a}}$	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
	$\phi_{\Delta} + \psi_{\text{d}}$	**86.9**	84.8	85.3	**83.9**
	$\phi_{\text{F-FAemb}} + \psi_{\text{d}}$	86.6	**85.9**	**85.6**	82.9
Holidays	$\phi_{\text{FV}} + \psi_{\text{a}}$	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
	$\phi_{\Delta} + \psi_{\text{d}}$	**89.1**	88.1	**88.6**	**87.3**
	$\phi_{\text{F-FAemb}} + \psi_{\text{d}}$	88.6	**88.4**	88.5	86.4

Table 4: Number of retained PCA components and codebook size when varying the dimensionality.

Dim. D	512-D	1024-D	2048-D	4096-D	8064-D
PCA d	32	64	64	64	128
$\lvert C\rvert$	20	18	34	66	64

The Figure 4 shows the retrieval performance of two datasets, *Oxford5k* and *Paris6k*, when varying the final feature dimensionality. Obviously, our proposed method can significantly boost the performance when increasing the final feature dimensionality. In addition, we also observe that the masking schemes consistently help to gain extra performance across different dimensionalities.



(a) *Oxford5k* **(b) *Paris6k***

Figure 4: Impact of the final representation dimensionality on *Oxford5k* and *Paris6k* datasets.

5.2.3 Image size. Even though the authors in [22, 39] found that the original size of images ($\max(W_I, H_I) = 1024$) provides higher performance, it is important to evaluate our method with a smaller image size on the performance since our method depends on the number of local conv. features. Table 5 shows the retrieval performance of *Oxford5k* and *Paris6k* datasets with the image size of $\max(W_I, H_I) = 1024$ and $\max(W_I, H_I) = 724$. Similar to the reported results of [39] on *Oxford5k* dataset, we observe around 6-7% drop in ***mAP*** when scaling down images to $\max(W_I, H_I) = 724$ rather than the original images. While on *Paris6k* dataset, interestingly, the performances are more stable to the image size. We also observe a small drop of 2.2% on *Paris6k* dataset for R-MAC [39] with our implementation. These suggest that our method and R-MAC method [39] equivalently affected by the change in the image size.

The performance drops on *Oxford5k* can be explained that with bigger 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 on *Paris6k* dataset can be perceived that the differences on these scenes are at global structures rather than small details as on *Oxford5k* dataset.

Table 5: Impact of input image size on *Oxford5k* and *Paris6k* datasets. The framework of $\mathcal{M}_{\mathbf{MAX/SUM}} + \phi_\Delta + \psi_\mathbf{d}$ is used to produce image representations.

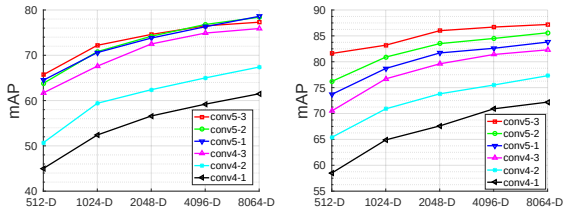
Dim. D	$\max(W_I, H_I)$	**Oxford5k**		**Paris6k**	
		$\mathcal{M}_{\text{SUM}}$	$\mathcal{M}_{\text{MAX}}$	$\mathcal{M}_{\text{SUM}}$	$\mathcal{M}_{\text{MAX}}$
512	724	56.4	60.9	79.3	81.2
	1024	64.0	65.7	78.6	81.6

5.2.4 Layer selection. In [4], while evaluating at different feature lengths, the authors claimed that deeper conv. layer produces features with more reliable similarities. Hence, we want to re-evaluate this statement by comparing the retrieval performance (***mAP***) of features extracted from different conv. layers at the same dimensionality. In this experiment, we extract features from different conv. layers, including `conv5-3`, `conv5-2`, `conv5-1`, `conv4-3`, `conv4-2`, and `conv4-1`, following by a 2×2 `max-pool` layer with stride of 2. The results of our comprehensive experiments on *Oxford5k* and *Paris6k* datasets are presented in Figure 5. We can observe that there are small drops in performance when using lower conv. layers until `conv4-3`. When going down further to `conv4-2` and `conv4-1`, there are significant drops in performance. Regarding the pre-trained VGG network [34], this fact indicates that the last conv. layer produces the most reliable representation for image retrieval.

5.3 Comparison to the state of the art

We thoroughly compare our proposed framework with state-of-art methods in image retrieval task. We report experimental results in Table 6.

Using off-the-shelf VGG network [34]. At dimensionality of 1024, our method using MAX-mask ($\mathcal{M}_{\text{MAX}} + \phi_\Delta + \psi_\text{d}$) achieves the highest ***mAP*** of all compared methods [1, 4, 22, 31, 39] with pre-trained VGG16 network [34] across different datasets. Note that some compared methods, e.g. [4, 22, 31, 39], have the dimensionality of 512 or 256. This is because the final feature dimensionality of these methods is upper bounded by the number of output feature channel K of network architecture and selected layer, e.g. K = 512 for `Conv5` of VGG16. While our proposed method provides more flexibility in the feature dimensional length. Furthermore, as discussed in Section 5.2.2, when increasing the final representation length, our methods can gain extra performance. In particular, at the dimensionality of 4096, our method is very competitive with methods that require complicated data collection process and days of re-training on powerful GPU [1, 31]. Our results at 4096-D are lower than [31] in *Oxford5k* while higher by a fair margin in *Paris6k* and *Holidays*.



(a) *Oxford5k* **(b) *Paris6k***

Figure 5: Evaluation of retrieval performance of local deep conv. features from different layers on *Oxford5k* and *Paris6k* datasets. The framework of $\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$ is used to produce image representations.

Note that it is unclear in the performance gain when increasing the length of the final representation in R-MAC [39] or CRoW [22], even at the cost of a significant increase in the number of CNN parameters and the additional efforts of re-training. In fact, in [3], the authors design an experiment to investigate whether increasing the number of conv. layers, before the fully connected one from which the representation is extracted, would help increase the performance of various visual tasks, including image classification, attribute detection, fine-grained recognition, compositional, and instance retrieval. Interestingly, the experimental results show that while the performance increases on other tasks, it degrades on the retrieval one. The authors explain that the more powerful the network is, the more generality it can provide. As a result, the representation becomes more invariant to instance level differences. Even though, in this experiment, the image representation is constructed from a fully-connected layer, which is different from our current context using conv. layer, the explanation in [3] could still be applicable. This raises the question about the efficiency of increasing number of channels in a conv. layer as a way to increase final representation dimensionality in SPoC [4], R-MAC [39], or CRoW [22].

Regarding NetVLAD [1] and MOP-CNN [14], these methods also can produce higher-dimensional representation. However, at a certain length, our method clearly achieves higher retrieval performance.

Taking advantages of fine-tuned VGG network. Since our proposed methods take the 3D activation tensor of a conv. layer as the input, our framework is totally compatible with fine-tuned networks [1, 31]. In the **Fine-tuned network** section of Table 6, we evaluate our best framework - $\mathcal{M}_{\text{MAX}} + \phi_\Delta + \psi_\text{d}$ - with the local conv. features of fine-tuned VGG for image retrieval task from [1, 31] as input. "NetVLAD ⋆" and "siaMAC †" mean that the fine-tuned VGG from NetVLAD [1] and siaMAC [31] respectively are

Table 6: Comparison with the state of the art.

	Method	Dim.	Datasets				
			Oxford5k	*Oxford105k*	*Paris6k*	*Paris106k*	*Holidays*
Off-the-shell network	SPoC [4]	256	53.1	-	50.1	-	80.2
	MOP-CNN [14]	512	-	-	-	-	78.4
	CroW [22]	512	70.8	65.3	79.7	72.2	85.1
	MAC [31]	512	56.4	47.8	72.3	58.0	76.7
	R-MAC [39]	512	66.9	61.6	83.0	75.7	-
	NetVLAD [1]	1024	62.6	-	73.3	-	87.3
	$\mathcal{M}_{\mathbf{SIFT}} + \phi_\Delta + \psi_\mathbf{d}$	512	64.4	59.4	79.5	70.6	86.5
	$\mathcal{M}_{\mathbf{SUM}} + \phi_\Delta + \psi_\mathbf{d}$	512	64.0	58.8	78.6	70.4	86.4
	$\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	512	65.7	60.5	81.6	72.4	85.0
	$\mathcal{M}_{\mathbf{SIFT}} + \phi_\Delta + \psi_\mathbf{d}$	1024	69.9	64.3	81.7	73.8	87.1
	$\mathcal{M}_{\mathbf{SUM}} + \phi_\Delta + \psi_\mathbf{d}$	1024	70.8	64.4	80.6	73.8	86.9
	$\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	1024	72.2	67.9	83.2	76.1	88.4
	NetVLAD [1]	4096	66.6	-	77.4	-	88.3
	$\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	4096	75.3	71.4	86.7	80.6	89.0
Finetuned network	siaMAC + R-MAC [31]	512	77.0	69.2	83.8	76.4	82.5
	NetVLAD fine-tuned [1]	1024	69.2	-	76.5	-	86.5
	siaMAC † [31] + $\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	512	77.7	72.7	83.2	76.5	86.3
	siaMAC † [31] + $\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	1024	81.4	77.4	84.8	78.9	88.9
	NetVLAD ⋆ [1] + $\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	1024	75.2	71.7	84.4	76.9	91.5
	NetVLAD fine-tuned [1]	4096	71.6	-	79.7	-	87.5
	NetVLAD ⋆ [1] + $\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	4096	78.2	75.7	87.8	81.8	92.2
	siaMAC † [31] + $\mathcal{M}_{\mathbf{MAX}} + \phi_\Delta + \psi_\mathbf{d}$	4096	83.8	80.6	88.3	83.1	90.1

used to extracted local conv. features. Additionally, "NetVLAD fine-tuned" represents the results reported in [1] after fine-tuning for differentiating the results using the off-the-shelf VGG network.

When using local conv. features extracted from fine-tuned network from [31], our method can achieve very competitive results with those from [31] at dimensionality of 512. Our method outperforms [31] in majority of benchmark datasets, including *Oxford5k*, *Oxford105k*, *Holidays*, and *Paris106k*. Furthermore, at 1024 dimensionality, our method outperforms the most competitive method [1, 31] by more than +2.5%, except *Paris6k* dataset with +1.0% performance gain, to the next best ***mAP*** values. It is important to note that the end-to-end training architecture proposed in [31] still inherits the drawback of upper-bounded final representation dimensionality from R-MAC [39].

5.4 Processing time

We empirically evaluate the online processing time of our proposed framework. We also compare the online processing time between our proposed framework and one of the most competitive methods[4]: R-MAC [39]. The experiments are carried out on a processor core (i7-6700 CPU @ 3.40GHz). The reported processing time in Figure 6 is the averaged online processing times of 5063 images of *Oxford5k* dataset using our default framework, excluding the time for feature extraction. This figure shows that by applying MAX/SUM-mask, our proposed framework can significantly reduce the computational cost, since they help remove about 70% and 50% of local conv. features respectively (Section 3.3). Additionally, at the dimensionality 512-D, our framework $\mathcal{M}_{\text{MAX/SUM}} + \phi_\Delta + \psi_\text{d}$ is computationally faster than R-MAC [39].

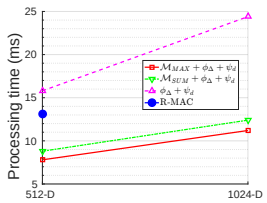


Figure 6: The averaged online processing time of 5063 images of *Oxford5k* dataset.

6 CONCLUSION

In this paper, we present an effective framework which takes activation of convolutional layer as input and produces highly-discriminative image representation for image retrieval. In our proposed framework, we propose to enhance discriminative power of the image representation in two main steps: (i) selecting a representative set of local conv. features using our proposed masking schemes, including SIFT/SUM/MAX mask, then (ii) embedding and aggregating using the state-of-art methods [12, 21]. Solid experimental results show that the proposed methods compare favorably with the state of the art. A further push the proposed system to achieve very compact binary codes (e.g., by jointly aggregating and hashing [11] or deep learning-based hashing [10]) seems interesting future works.

[4]We do not evaluate online processing time for CRoW [22] as its published codes are in Python, and it is not appropriate to directly compare with the Matlab implementation of our method.

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] Y-Lan Boureau, Jean Ponce, and Yann Lecun. 2010. A Theoretical Analysis of Feature Pooling in Visual Recognition. In *ICML*.

[7] 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*.

[8] Jonathan Delhumeau, Philippe-Henri Gosselin, Hervé Jégou, and Patrick Pérez. 2013. Revisiting the VLAD image representation. In *ACM MM*.

[9] Thanh-Toan Do and Ngai-Man Cheung. 2017. Embedding based on function approximation for large scale image search. *TPAMI* (2017).

[10] Thanh-Toan Do, Anh-Dzung Doan, and Ngai-Man Cheung. 2016. Learning to hash with binary deep neural network. In *ECCV*.

[11] 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*.

[12] Thanh-Toan Do, Quang Tran, and Ngai-Man Cheung. 2015. FAemb: A function approximation-based embedding method for image retrieval. In *CVPR*.

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*.

[14] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. 2014. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*.

[15] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. 2016. Deep Image Retrieval: Learning Global Representations for Image Search. In *ECCV*.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385* (2015).

[17] Hervé Jégou and Ondřej Chum. 2012. Negative Evidences and Co-occurences in Image Retrieval: The Benefit of PCA and Whitening. In *ECCV*.

[18] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2009. On the burstiness of visual elements. In *CVPR*.

[19] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2010. Improving Bag-of-Features for Large Scale Image Search. *IJCV* 87, 3 (May 2010), 316–336.

[20] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating local descriptors into a compact image representation. In *CVPR*.

[21] Hervé Jégou and Andrew Zisserman. 2014. Triangulation embedding and democratic aggregation for image search. In *CVPR*.

[22] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-dimensional Weighting for Aggregated Deep Convolutional Features. In *ECCV Workshops*.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

[24] Ying Li, Xiangwei Kong, Liang Zheng, and Qi Tian. 2016. Exploiting Hierarchical Activations of Neural Network for Image Retrieval. In *ACM MM*.

[25] David G. Lowe. 1999. Object Recognition from Local Scale-Invariant Features. In *ICCV*.

[26] Romain Negrel, David Picard, and P Gosselin. 2013. Web scale image retrieval using compact tensor aggregation of visual descriptors. In *MultiMedia*, Vol. 20. IEEE, 24–33.

[27] Florent Perronnin and Christopher Dance. 2007. Fisher Kernels on Visual Vocabularies for Image Categorization. In *CVPR*.

[28] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. 2010. Improving the fisher kernel for large-scale image classification. In *ECCV*.

[29] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*.

[30] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2008. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*.

[31] Filip Radenović, Giorgos Tolias, and Ondřej Chum. 2016. CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples. In *ECCV*.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 115, 3 (2015), 211–252.

[34] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[35] Josef Sivic, Andrew Zisserman, and others. 2003. Video Google: a text retrieval approach to object matching in videos. In *ICCV*.

[36] 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*.

[37] 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.

[38] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. 2013. To Aggregate or Not to aggregate: Selective Match Kernels for Image Search. In *ICCV*.

[39] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. 2016. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR*.

[40] Andrea Vedaldi and Brian Fulkerson. 2008. VLFeat: An Open and Portable Library of Computer Vision Algorithms. http://www.vlfeat.org/,. (2008).

[41] Andrea Vedaldi and Karel Lenc. 2014. MatConvNet - Convolutional Neural Networks for MATLAB. *CoRR* abs/1412.4564 (2014). http://arxiv.org/abs/1412.4564

[42] Ke Yan, Yaowei Wang, Dawei Liang, Tiejun Huang, and Yonghong Tian. 2016. CNN vs. SIFT for Image Retrieval: Alternative or Complementary?. In *ACM MM*.

[43] Kai Yu and Tong Zhang. 2010. Improved Local Coordinate Coding using Local Tangents. In *ICML*.

[44] Matthew D. Zeiler and Rob Fergus. 2013. Visualizing and Understanding Convolutional Networks. *CoRR* abs/1311.2901 (2013). http://arxiv.org/abs/1311.2901