REGION BASED IMAGE RETRIEVAL WITH QUERY-ADAPTIVE FEATURE FUSION

Guixuan Zhang[1,3], Shuwu Zhang[1,2], Zhi Zeng[1], Hu Guan[1], Fangxin Wang[1,3]

[1] Beijing Engineering Research Center of Digital Content Technology, Institute of Automation, CAS
[2] Advanced Innovation Center for Future Visual Entertainment
[3] University of Chinese Academy of Sciences
{guixuan.zhang, shuwu.zhang, zhi.zeng, hu.guan, wangfangxin2014}@ia.ac.cn

ABSTRACT

Recently, image representation based on convolutional neural network (CNN) becomes more popular than SIFT based feature, such as Fisher vector (FV). However, which of the two works better for image retrieval is not entirely clear yet. In this paper, we propose to fuse CNN and FV to incorporate the advantages of both features for image retrieval. We extract CNN feature and FV from multi-scale regions, which makes the representation more robust to image noise. Then a query-adaptive feature fusion method is proposed, which is used jointly with 2-D inverted index under the framework of bag-of-words. Moreover, we make an evaluation of different CNN feature extraction methods for the region based method. Extensive experiments on four benchmark datasets demonstrate the effectiveness of our method with efficiency in both time cost and memory usage.

Index Terms— Image retrieval, convolutional neural networks, fisher vector, query-adaptive feature fusion

1. INTRODUCTION

This paper considers the task of image and object retrieval. Image retrieval aims at finding images that represent the same object or scene viewed under different imaging conditions in a large set of images.

Many image retrieval systems adopt the bag-of-words (BoW) model [1,2] and rely on matching of local descriptors, such as SIFT [3]. In BoW, a visual vocabulary is trained with k-means algorithm. Two SIFTs are assumed to match if they are quantized to the same visual word. An inverted index is often leveraged to enable fast retrieval. To improve the matching accuracy, Hamming embedding (HE) is widely used [4]. It refines a binary signature for each SIFT when quantizing it. Two coarsely matched SIFTs will be filtered out if their Hamming distance between binary codes is larger than a threshold. In the SIFT based matching methods, each feature should be indexed individually. Since several thousand SIFTs are often extracted from an image, the search time and memory usage may be important constraints when the database is very large.

Another strategy for image retrieval is to extract the global representation from the whole image. Every image is described by a single vector. Aggregated SIFT features such as Fisher vector (FV) [5] and VLAD [6] used to be widely adopted. These features preserve some properties inherited from SIFT, such as rotation and scale invariance. Recently, convolutional neural network (CNN) achieves state-of-the-art performance in many computer vision tasks [7-10]. The research focus of image retrieval also turns to the CNN model. Image representation built on pre-trained CNN shows promising results for image search [11,12]. However, CNN representation lacks invariance to geometrical transformation, such as rotation [13].

Fig. 1. Two examples of image retrieval from UKBench [2]. Multi-scale regional features are used. For each query, results obtained by CNN feature (the first row), FV (the second row), conventional 2-D inverted index with CNN and FV (the third row), and our proposed method (the last row) are demonstrated. Relevant images are marked in red, and irrelevant ones green. FV works better for the left query while CNN works better for the right query. For both queries, feature fusion with conventional 2-D inverted index results in a lower accuracy.

This paper proposes to fuse CNN and FV for image retrieval. Instead of extracting features from the whole image, we decompose the image to multi-scale regions and extract the feature from each region, which makes the representation more robust to occlusion and clutter interference. Regional features have been used by some recent works [13-17]. However, all these methods suffer from either low efficiency or limited accuracy. In this work, we adopt 2-D inverted index to fuse the region based CNN and FV in an efficient way. The multi-dimensional inverted index proposed by [18] has been employed by [19,20] for image search. The problem of conventional multi-index is that it ignores the variety of the effectiveness of a feature. Each feature may demonstrate different effectiveness for different queries and a bad feature may lower the accuracy

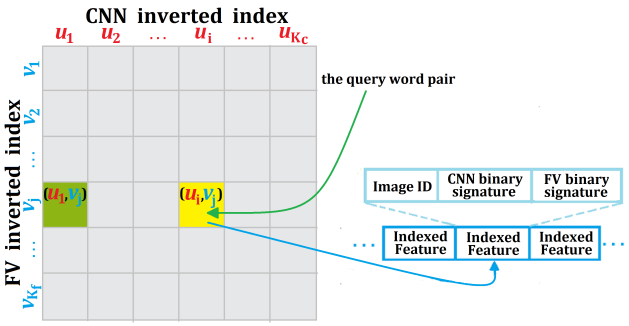

Fig. 2. Structure of 2-D inverted index. It contains $K_c \times K_f$ entries. Each entry corresponds to an inverted list, which stores the regional features quantized to this entry. During online retrieval, the entry of word tuple (u_i, v_j) is checked. Assume that one true-matching region is in the entry of (u_1, v_j). If FV works only, all entries in Row v_j are explored and the matching region can be retrieved. When fusing FV and CNN, it will be missed.

after fusion (Fig. 1). To address this problem, we propose a query-adaptive method to identify feature effectiveness. According to the feature effectiveness, we assign different search strategies and weights to different features.

The contribution of this paper is summarized in three aspects. First, we fuse CNN and FV to incorporate the advantages of both features for image retrieval task. We extract features from multi-scale regions to weaken the interference of image noise. Second, we introduce a query-adaptive feature fusion method, which helps to improve the performance. Third, we evaluate different CNN feature extraction methods for region based image search. Extensive experiments on benchmark datasets demonstrate that our method compares favorably to state-of-the-art methods with efficiency in terms of both time cost and memory usage.

The remainder of this paper is organized as follows. We review 2-D inverted index in Section 2. The proposed algorithm is shown in Section 3. We provide experimental results in Section 4. Final conclusions are in Section 5.

2. 2-D INVERTED INDEX

Let $\vec{x} = [x^c, x^f]$ be a coupled feature for a region in image Q. In this paper, two kinds of regional feature, CNN and FV, are considered. For 2-D inverted index, two vocabulary are trained for each feature, noted by $\mathcal{U} = \{u_1, u_2, ..., u_{K_c}\}$ and $\mathcal{V} = \{v_1, v_2, ..., v_{K_f}\}$, where K_c and K_f are vocabulary sizes. As a result, this 2-D inverted index contains $K_c \times K_f$ entries. After building the index, all feature tuples $\vec{x} = [x^c, x^f]$ are quantized to visual word pairs (u_i, v_j) using vocabulary $\mathcal{U}$ and $\mathcal{V}$, where u_i and v_j are the nearest centroids to feature x^c and x^f, respectively. Then useful cues (e.g. image ID and regional binary signatures) related to the current feature tuple $\vec{x}$ are saved in the corresponding entry (see Fig. 2).

Given two feature tuples $\vec{x} = [x^c, x^f]$ and $\vec{y} = [y^c, y^f]$, the matching function is written as

$$f^0(\vec{x}, \vec{y}) = \delta_{q_c(x^c), q_c(y^c)} \cdot \delta_{q_f(x^f), q_f(y^f)}, \quad (1)$$

where $q_c(\cdot)$ and $q_f(\cdot)$ are quantization functions for two different features, and δ is the Kronecker delta response. To further improve the matching precision, methods such as HE can be used. Each regional feature is associated with a binary signature. Then the matching function is updated as

$$f(\vec{x}, \vec{y}) = \begin{cases} f^0(\vec{x}, \vec{y}) \left(\exp(\frac{-d_c^2}{\sigma_c^2}) + \exp(\frac{-d_f^2}{\sigma_f^2}) \right) & \textbf{if } d_c \le \kappa, d_f \le \kappa \\ 0 & \textbf{otherwise} \end{cases}, \quad (2)$$

where d_c and d_f are the Hamming distance between their corresponding binary signatures, and κ is the Hamming distance threshold. σ_c and σ_f are the parameters of exponential score function $s = \exp(-d_c^2/\sigma_c^2) + \exp(-d_f^2/\sigma_f^2)$.

With 2-D inverted index, fewer entries need to be explored, which makes the search process efficient. However, it may result in a low recall. In Fig. 2, the query and one matching region are not quantized to the same word in CNN feature space. Though FV works well for this query, the true-matching region is still missed after feature fusion.

3. PROPOSED APPROACH

3.1. Multi-Scale Regions

From each image, we extract L-layer regions of different sizes at different locations. Denote the width and height of an image as W and H. At level l, $1 \le l \le L$, we extract l^2 overlapping regions whose union covers the entire image. All regions at the l-th level have a fixed size $(w_l = 2W/(l+1), h_l = 2H/(l+1))$. In our experiments, we set $L = 4$ and therefore generate 30 regions for each image.

3.2. Regional Features

We extract CNN features based on two classic architectures, AlexNet [7] and VGGNet [8]. These two models have been trained on ImageNet, which is a large dataset consisting of diverse images. We introduce different feature extraction methods and will evaluate them in the experimental section.

The outputs of fully connected (FC) layers describe high-level semantics of the input regions. Features from the first and second FC layer are often adopted. For AlexNet, FC6 and FC7 are used. For VGGNet (we use the 16-layer model), FC14 and FC15 are used.

The outputs of deep convolutional layers correspond to mid-level visual patterns. To increase the invariance to image translations, global pooling should be used. We test two pooling types, sum-pooling [21] and max-pooling [22]. The features of the last convolutional layer are usually employed [21-23]. These two CNN features are denoted as ConvSumP and ConvMaxP, respectively. Instead of decomposing the original image, we generate multi-scale regions and extract features directly on the convolutional feature maps. In this way, we can feed the entire image to the network only once to obtain multiple convolutional features.

For each region in an image, FV and one kind of the CNN features are extracted. All the regional features are indexed according to the 2-D inverted index in Section 2. To generate regional binary signature, we choose locality-sensitive hashing (LSH) algorithm [24]. Each regional feature is associated with a 128-bit signature.

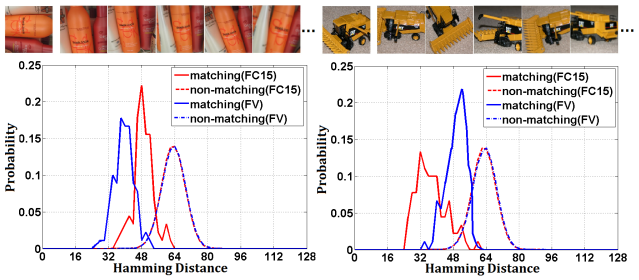

Fig. 3. Distribution of the Hamming distance for two query regions. We generate a set of matching regions according to the ground truth. The non-matching regions are randomly selected from unrelated images. FC15 and FV are used to describe the regions. Each regional feature is transformed to 128-bit signature with LSH.

3.3. Query-Adaptive Feature Fusion

To address the problem of conventional 2-D inverted index (described in Section 2 and illustrated in Fig. 2), we introduce a query-adaptive feature fusion method. This approach identifies the effectiveness of each feature, and then assigns different search strategies and weights to different features according to the feature effectiveness.

To identify which feature works better, we choose the Hamming distance of regional binary signatures as the evidence. As an example, Fig. 3 depicts the distributions of the Hamming distance of the matching and non-matching features for two different query regions. It can be seen that, if one feature works better for a given query, the Hamming distance between the query and its matching regions is more likely to be smaller. Based on this observation, our query-adaptive feature fusion is two-stage as follows.

First Stage. This stage does an initial search based on Eq. 2. Since some truly matching features may be assigned to different visual words, we apply multiple assignment (MA) [4] to decrease the interference. Each query feature is assigned to several nearest visual words. The MA value is denoted as MA_1 in this stage. For a query region with feature tuple $\vec{x}=[x^{\mathrm{c}},x^{\mathrm{f}}]$, we retrieve a set of regions $\mathcal{R}=\{r_t,t=1,...,T\}$. Each region in the set is associated with two Hamming distances to the query.

Second Stage. We sort the Hamming distances in the set $\mathcal{R}$ for CNN and FV, respectively. For each kind of feature, we find p smallest Hamming distances. The corresponding p regions are more likely to be similar to the query region since they have the smallest Hamming distance to the query. Then we calculate the average of the p Hamming distances for each feature, denoted as h_{c} and h_{f}, respectively. By comparing h_{c} and h_{f}, we identify the feature effectiveness. The feature with smaller average Hamming distance is considered as the more effective feature, and the other is less effective feature. Since some similar regions may be far away from each other in the less effective feature space, we increase the MA value (denoted as MA_2) for this feature to explore more visual words, so that the recall can be improved. We set $\mathrm{MA}_2=\max(h_{\mathrm{c}},h_{\mathrm{f}})/128\times m$, where m will be discussed in the experimental section.

We also allocate different weights according to the feature effectiveness. The exponential score function in Eq. 2 is updated as

$$s=\begin{cases}\exp(-d_{\mathrm{c}}^2/\sigma_{\mathrm{c}}^2)+h_{\mathrm{c}}/h_{\mathrm{f}}\cdot\exp(-d_{\mathrm{f}}^2/\sigma_{\mathrm{f}}^2) & \textbf{if } h_{\mathrm{c}}\le h_{\mathrm{f}}\\ h_{\mathrm{f}}/h_{\mathrm{c}}\cdot\exp(-d_{\mathrm{c}}^2/\sigma_{\mathrm{c}}^2)+\exp(-d_{\mathrm{f}}^2/\sigma_{\mathrm{f}}^2) & \textbf{otherwise}\end{cases}. \quad (3)$$

Finally, the similarity score between a database image I and query image Q is defined as

$$sim(Q,I)=\sum_{\vec{x}\in Q,\vec{y}\in I} f(\vec{x},\vec{y}). \quad (4)$$

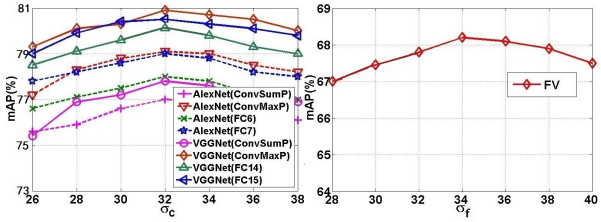

Fig.4. Influence of σ_{c} and σ_{f} on the Holidays dataset.

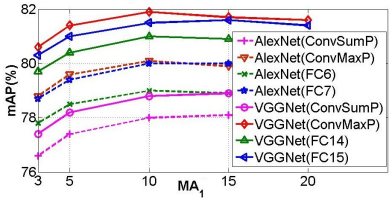

Fig. 5. The influence of MA_1 in our method.

4. EXPERIMENTS

We evaluate the proposed method on four benchmark datasets: Holidays [4], UKBench [2], Oxford Buildings [25] and Paris Buildings [26]. Following the standard evaluation protocols, we report mean Average Precision (mAP) for Holidays, Oxford and Paris. For UKBench, the performance is reported by the average recall of the top 4 ranked images, referred to as N-S score.

Implementation details. We adopt AlexNet and VGG-Net implemented by Caffe [27] to extract CNN features. For FV, 64 Gaussian mixture models are used. The vocabulary size is set to 1000 for both K_{c} and K_{f}, resulting in a total of 1M entries. The Hamming threshold κ is set to 64.

Parameter analysis. We test the parameters σ_{c} and σ_{f} with 1-D inverted index and apply the optimized value when 2-D inverted index is used. According to the results in Fig. 4, we set $\sigma_{\mathrm{c}}=32$ and $\sigma_{\mathrm{f}}=34$. The influence of MA_1 which is used in the first stage of our fusion method is shown in Fig.5. Based on the results, we set $\mathrm{MA}_1=10$.

We test different p and show the results in Fig. 6. Small value (e.g. 1) is sensitive to noise. A large value may involve many irrelevant regions to compute the average Hamming distance, which leads to the failure in identifying the feature effectiveness. Based on the results, we choose to set $p=10$, which achieves promising performance. The impact of parameter m in MA_2 is shown in Fig. 7. Increasing m means more entries will be explored and the recall can be improved. However, large m increases the search time. We set $m=400$ for the rest of our experiments.

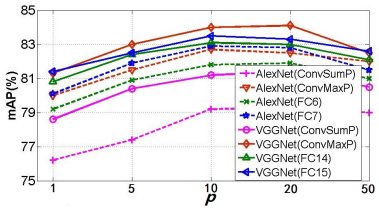

Fig. 6. Impact of p on the Holidays dataset. m is set to 1000.

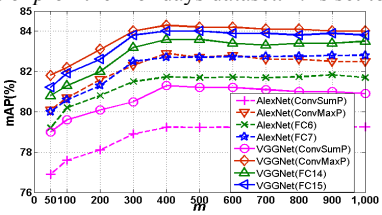

Fig. 7. Influence of the parameter m on the Holidays dataset.

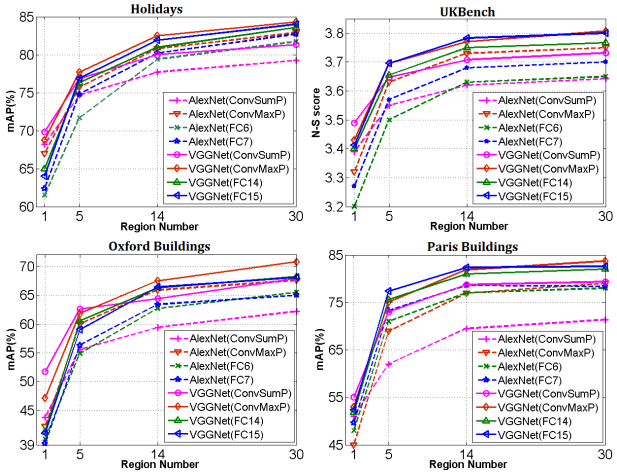

Fig. 8. Results with different CNN features. We test the performance for different number of regions (from $L = 1$ to $L = 4$).

Evaluation for different CNN features. We evaluate the performance of different CNN features and show the results in Fig. 8. Features extracted from VGGNet have better performance than features of AlexNet. When only one region is used, the feature means the global representation. In this case, ConvSumP is superior to other features. Such behavior is consistent with other works [21,28]. However, with the increase of region number, ConvSumP is outperformed by other features. Comparing to FC features, ConvMaxP seems to have better results in most cases. Another advantage of convolutional features is that we need to pass the image through the deep network only once to extract multiple regional features. Based on these results, we suggest that ConvMaxP feature is a good choice for the region based image retrieval approach.

Evaluation of the query-adaptive feature fusion. In Table 1, comparing to conventional 2-D inverted index fusion method, the proposed feature fusion approach improves the performance significantly for all the four datasets, which shows the effectiveness of our method.

Complexity. In our method, 4 bytes are allocated to store image ID for each indexed feature. Each regional feature requires 16 bytes to store 128-bit signature. So every image

Table 1. Evaluation of our query-adaptive feature fusion method. CMP is short for ConvMaxP. * indicates that conventional 2-D inverted index method is used.

Method	Feature	Holidays	UKB	Oxford	Paris
2-D Index*	FV+Alex(CMP)	79.9	3.71	63.2	77.3
	FV+VGG(CMP)	82.1	3.74	66.5	81.0
Proposed	FV+Alex(CMP)	82.9	3.75	67.6	79.2
	FV+VGG(CMP)	**84.3**	**3.80**	**70.7**	**83.6**

Table 2. Comparison with state-of-the-art. * we assume there are 1000 SIFTs per image. All CNN based methods extract features from pre-trained models. They are without any fine-tuning stage.

Methods &Types		Holidays	UKB	Oxford	Paris	Mem./Img (bytes)
SIFT based	[4]	81.3	3.38	60.5	-	$12\times1000^*$
	[19]	84.0	3.71	-	-	$15\times1000^*$
	[29]	81.0	-	**80.4**	77.0	$20\times1000^*$
CNN based	[14]	84.3	3.64	68.0	79.5	16k
	[20]	85.6	3.76	-	81.2	4.06k
	[21]	78.4	3.66	65.7	-	1k
	[23]	-	-	66.8	83.0	2k
Fusion	[30]	**86.2**	3.78	79.8	-	22.9k*
	Ours	84.3	**3.80**	70.7	**83.6**	1080

costs $(4+16\times2)\times30=1080$ bytes. Our method consumes only 1GB memory for 1M dataset. The extraction of FV and ConvMaxP takes an average of 0.15s and 0.08s, respectively. The average query time on 1M dataset is 0.72s.

Comparison with state-of-the-art. The comparison is shown in Table 2. We achieve the best performance for UKBench and Paris datasets. For Holidays, our result falls slightly behind [20]. Note that [20] trains the vocabulary from the Holidays dataset itself, while we train it from an independent dataset. For Oxford, SIFT based matching method [29] has the best performance. Since thousands of SIFTs are extracted from an image, this method has low search efficiency. The fusion method of [30] performs the best on Holidays, but it also suffers large memory cost and low efficiency. Some recent works [31,32] achieve better performance than ours on Holidays, Oxford and Paris. However, their objectives are different from ours. They focus on learning good representation by fine-tuning CNN models for particular task, while our work focuses on the search strategy for fusing features. Their CNN features can also be used in our approach to improve the performance.

5. CONCLUSIONS

We propose to extract different features from multi-scale regions and adopt 2-D inverted index to implement efficient image retrieval. The query-adaptive feature fusion method has been proven to improve the performance. We evaluate different CNN feature extraction methods and suggest that ConvMaxP works best for region based retrieval approach.

6. ACKNOWLEDGEMENT

This work is part of the research achievements of the Key Laboratory of Digital Rights Services, which is one of the National Science and Standardization Key Labs for Press and Publication Industry.

7. REFERENCES

[1] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *ICCV*, pp. 1470–1477, 2003.

[2] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," in *CVPR*, pp. 2161–2168, 2006.

[3] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis*., vol. 60, no. 2, pp. 91–110, 2004.

[4] H. Jégou, M. Douze, and C. Schmid, "Improving Bag-of-Features for Large Scale Image Search," *Int. J. Comput. Vis*., vol. 87, pp. 316–336, 2010.

[5] F. Perronnin and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," in *CVPR*, pp. 1-8, 2007.

[6] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating Local Image Descriptors into Compact Codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704-1716, 2012.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks." in *NIPS,* pp. 1106-1114, 2012.

[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," CoRR abs/1409.1556, 2014.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *NIPS,* pp. 91-99, 2015.

[10] J. Long, E. Shelhamer, T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *CVPR*, pp. 3431-3440, 2015.

[11] A. Babenko, A. Slesarev, A. Chigorin, and V. S. Lempitsky, "Neural Codes for Image Retrieval," in *ECCV*, pp. 584-599, 2014.

[12] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep Learning for Content-Based Image Retrieval: A Comprehensive Study," in *ACM Multimedia*, pp157-166, 2014.

[13] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-Scale Orderless Pooling of Deep Convolutional Activation Features," in *ECCV*, pp. 392-407, 2014.

[14] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features Off-The-Shelf: An Astounding Baseline for Recognition," in *CVPR Workshops*, pp.512-519, 2014.

[15] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image Classification and Retrieval are ONE," in *ICMR*, pp. 3-10, 2015.

[16] J. Y. H. Ng, F. Yang, and L. S. Davis, "Exploiting Local Features from Deep Networks for Image Retrieval," *in CVPR Workshops*, pp. 53-61, 2015.

[17] K. R. Mopuri and R. V. Babu, "Object Level Deep Feature Pooling for Compact Image Representation," in *CVPR Workshops*, pp. 62-70, 2015.

[18] A. Babenko and V. Lempitsky, "The Inverted Multi-Index", in *CVPR*, pp. 3069-3076, 2012.

[19] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Packing and Padding: Coupled Multi-Index for Accurate Image Retrieval," in *CVPR*, pp. 1947-1954, 2014.

[20] Y. Liu, Y. Guo, S. Wu, and M. S. Lew, "DeepIndex for Accurate and Efficient Image Retrieval," in *ICMR*, pp. 43-50, 2015.

[21] A. Babenko and V. Lempitsky, "Aggregating Deep Convolutional Features for Image Retrieval," in *ICCV*, pp.1269-1277, 2015.

[22] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "A Baseline for Visual Instance Retrieval with Deep Convolutional Networks," in *ICLR Workshops*, 2015.

[23] G. Tolias, R. Sicre, and H. Jégou, "Particular Object Retrieval with Integral Max-Pooling of CNN Activations," in *ICLR*, 2016.

[24] M. S. Charikar, "Similarity Estimation Techniques from Rounding Algorithms," in *ACM Symposium on Theory of Computing*, pp.380-388, 2002.

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *CVPR*, pp. 1-8, 2007.

[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases," in *CVPR*, pp. 1-8, 2008.

[27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," CoRR abs/1408.5093, 2014.

[28] A. Salvador, X. Giro-i-Nieto, F. Marques, and S. Satoh, "Faster R-CNN Features for Instance Search," in *CVPR Workshops*, pp. 9-16, 2016.

[29] G. Tolias, Y. Avrithis, and H. Jégou, "To Aggregate or not to Aggregate: Selective Match Kernels for Image Search," in *ICCV*, pp. 1401-1408, 2013.

[30] L. Zheng, S. Wang, J. Wang, and Q. Tian, "Accurate Image Search with Multi-Scale Contextual Evidences", *Int. J. Comput. Vis*., vol. 120, no. 1, pp. 1–13, 2016.

[31] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "Deep Image Retrieval: Learning Global Representations for Image Search," in *ECCV*, pp 241-257, 2016.

[32] F. Radenovic, G. Tolias, and O. Chum, "CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples," in *ECCV*, pp 3-20, 2016.