

# Efficient Nearest Neighbors Search for Large-Scale Landmark Recognition

Federico Magliani, Tomaso Fontanini, and Andrea Prati

IMP lab - University of Parma, 43124 Parma, Italy,  
federico.magliani@studenti.unipr.it,  
<http://implab.ce.unipr.it>

**Abstract.** The problem of landmark recognition has achieved excellent results in small-scale datasets. When dealing with large-scale retrieval, issues that were irrelevant with small amount of data, quickly become fundamental for an efficient retrieval phase. In particular, computational time needs to be kept as low as possible, whilst the retrieval accuracy has to be preserved as much as possible. In this paper we propose a novel multi-index hashing method called Bag of Indexes (BoI) for Approximate Nearest Neighbors (ANN) search. It allows to drastically reduce the query time and outperforms the accuracy results compared to the state-of-the-art methods for large-scale landmark recognition. It has been demonstrated that this family of algorithms can be applied on different embedding techniques like VLAD and R-MAC obtaining excellent results in very short times on different public datasets: Holidays+Flickr1M, Oxford105k and Paris106k.

**Keywords:** Landmark recognition, Nearest neighbors search, Large-scale image retrieval, Approximate search

## 1 Introduction

Landmark recognition is an emerging field of research in computer vision. In a nutshell, starting from an image dataset divided into classes, with each image represented by a feature vector, the objective is to correctly identify to which class a query image belongs. This task presents several challenges: reaching high accuracy in the recognition phase, fast research time during the retrieval phase and reduced memory occupancy when working with a large amount of data. The large-scale retrieval has recently become interesting because the results obtained in the majority of small-scale datasets are over the 90% of the accuracy retrieval (*e.g.* to Gordo *et al.*[6]). Searching the correct  $k$  nearest neighbors of each query is the crucial problem of large-scale retrieval because, due to the great dimension of data, a lot of distractors are present and should not be considered as possible query neighbors. In order to deal with large-scale datasets, an efficient search algorithm, that retrieves query results faster than naïve brute force approach, while keeping a high accuracy, is crucial. With an approximate search not all the returned neighbors are correct, then some are approximate, but they are typically still close to the exact neighbors. Usually, obtaining good results in the image retrieval task is strictly correlated with the high dimensionality of the global image descriptors,

but on a large-scale version of the same problem is not advisable to use the same approach, due to the large amount of memory that would be needed. A possible solution is to first reduce the dimensionality of the descriptors, for example through PCA, and, then, apply techniques based on hashing functions for an efficient retrieval.

Following this strategy, this paper introduces a new multi-index hashing method called *Bag of Indexes* (BoI) for large-scale landmark recognition based on Locality-Sensitive Hashing (LSH) and its variants, which allows to minimize the accuracy reduction with the growth of the data. The proposed method is tested on different public benchmarks using different embeddings in order to prove that is not an ad-hoc solution.

This paper is organized as follows. Section 2 introduces the general techniques used in the state of the art. Next, Section 3 describes the proposed *Bag of Indexes* (BoI) algorithm. Finally, Section 4 reports the experimental results on three public datasets: Holidays+Flickr1M, Oxford105k and Paris106k. Finally, concluding remarks are reported.

## 2 Related work

In the last years, the problem of landmark recognition was addressed in many different ways [12] [19] [22]. Recently, with the development of new powerful GPUs, the deep learning approach has shown its superior performance in many tasks of image retrieval [24] [5] [1] [26].

Whenever the number of images in the dataset becomes too large, a Nearest Neighbor (NN) search approach to the landmark recognition task becomes infeasible, due to the well-known problem of the curse of dimensionality. Therefore, Approximate Nearest Neighbors (ANN) becomes useful, since it consists in returning a point that has a distance from the query equals to at most  $c$  times the distance from the query to its nearest points, where  $c > 1$ .

One of the proposed techniques that allows to efficiently treat the ANN search problem is the Locality-Sensitive Hashing (LSH [9]), where the index of the descriptor is created through hash functions. LSH projects points that are close to each other into the same bucket with high probability. There are many different variants of LSH, such as E2LSH [3], multi-probe LSH [15], and many others.

While LSH is a data-independent hashing method, there exist also data-dependent methods like Spectral Hashing [25], which, however, is slower than LSH and therefore not appropriate for large-scale retrieval. In Permutation-Pivots index [2], data objects and queries are represented as appropriate permutations of a set of randomly selected reference objects, and their similarity is approximated by comparing their representation in terms of permutations. Product Quantization (PQ) [10] is used for searching local descriptors. It divides the feature space in disjoint subspaces and then quantizes each subspace separately. It pre-computes the distances and saves them in look-up tables for speeding up the search. Locally Optimized Product Quantization (LOPQ) [13] is an optimization of PQ that tries to locally optimize an individual product quantizer per cell and uses it to encode residuals. Instead, FLANN [17] is an open source library for ANN and one of the most popular for nearest neighbor matching. It includes dif-

ferent algorithms and has an automated configuration procedure for finding the best algorithm to search in a particular data set.

### 3 Bag of Indexes

The proposed Bag of Indexes (BoI) borrows concepts from the well-known Bag of Words (BoW) approach. It is a form of multi-index hashing method [7] [18] for the resolution of ANN search problem.

Firstly, following the LSH approach,  $L$  hash tables composed by  $2^\delta$  buckets, that will contain the indexes of the database descriptors, are created. The parameter  $\delta$  represents the hash dimension in bits. The list of parameters of BoI and chosen values are reported in Table 1 in Section 3.2. Secondly, the descriptors are projected  $L$  times using hashing functions. It is worth to note that this approach can be used in combination with different projection functions, not only hashing and LSH functions. Finally, each index of the descriptors is saved in the corresponding bucket that is the one matching the projection result.

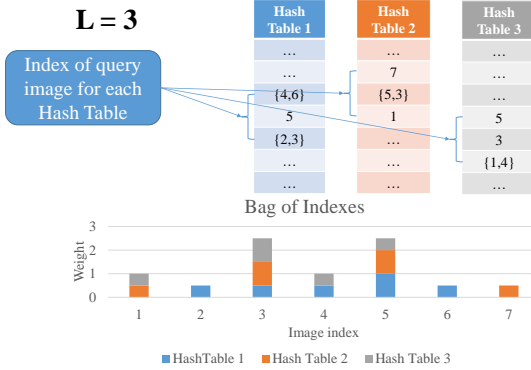
At query time, for each query, a BoI structure is created, that is a vector of  $n$  weights (each corresponding to one image of the database) initialized to zero. Every element of the vector will be filled based on the weighing method explained in Section 3.1. So, at the end of the projection phase, it is possible to make a coarse-grain evaluation of the similarity between the query image and the other images without calculating the Euclidean distance between them, but considering only their frequencies in the query buckets. Subsequently, at the end of the retrieval phase, the  $\varepsilon$  elements of the vector with the highest weights are re-ranked according to their Euclidean distance from the query. The nearest neighbor is then searched only in this short re-ranked list. By computing the Euclidean distances only at the end of the retrieval phase and only on this short list (instead of computing them on each hash table like in standard LSH), the computational time is greatly reduced. Furthermore, this approach, unlike LSH, does not require to maintain a ranking list without duplicates for all the  $L$  hash tables. The detailed analysis of the memory occupation of BoI is reported in Section 4.

#### 3.1 Weighing metric

As previously reported, BoI can be used in combination with different hashing functions. When used with baseline LSH, the corresponding bucket of the query image will be checked. In this case, even though it is faster than LSH, the accuracy suffers a significant loss. Conversely, when BoI is combined with multi-probe LSH, also the  $l$ -neighboring buckets are considered.

The  $l$ -neighbors are the buckets that have a Hamming distance less than or equal to  $l$  from the hashed value of the query, which corresponds to the query bucket. The weights for the any value of  $l$  are chosen as follows:

$$w(i, q, l) = \begin{cases} \frac{1}{2^{H(i, q)}} & \text{if } H(i, q) \leq l \\ 0 & \text{otherwise} \end{cases} \quad (1)$$



**Fig. 1.** Overview figure of the retrieval through BoI multi-probe LSH.

where  $i$  is a generic bucket,  $q$  is the query bucket and  $H(i, q)$  is the Hamming distance between  $i$  and  $q$ . The BoI multi-probe LSH approach increases the number of buckets considered during the retrieval and, thus, the probability of retrieving the correct result, by exploiting the main principle of LSH that similar objects should fall in the same bucket or in the ones that are close to it. However, even if we want to account for some uncertainty in the selection of the correct bucket, we also want to weight less as soon as we move farther from the “central” bucket.

Fig. 1 shows an exemplar overview of the BoI computation. With  $L = 3$  hash tables and 1-neighbours (i.e.,  $l = 1$ ), a query can be projected in different buckets. The corresponding weights (see eq. 1) are accumulated in the BoI (see the graph on the bottom of the image). Only the  $\epsilon$  images with the highest weights are considered for the last step (re-ranking) for improving the recall.

### 3.2 BoI adaptive multi-probe LSH

This BoI multi-probe LSH approach has the drawback of increasing the computational time since it also needs to search in neighboring buckets (which are  $\sum_{i=0}^l \binom{\log_2 \delta}{i}$ , being  $\delta$  the hash dimension). To mitigate this drawback, we introduce a further variant, called *BoI adaptive multi-probe LSH*. The main idea of this approach is to iteratively refine the search bucket space, by starting with a large number of neighboring buckets  $\gamma_0$  (e.g., 10) and slowly reduce  $\gamma$  when the number of hash tables increases. This adaptive increase of focus can, on the one hand, reduce the computational time and, on the other hand, reduce the noise. In fact, at each iteration, the retrieval results are supposed to be more likely correct and the last iterations are meant to just confirm them, so there is no need to search on a large number of buckets. In order to avoid checking the same neighbors during different experiments, the list of neighbors to check is shuffled randomly at each experiment.

Two different techniques for the reduction of the number of hash tables are evaluated:

- *linear*: the number of neighboring buckets  $\gamma$  is reduced by 2 every 40 hash tables, i.e.:

$$\gamma_i = \begin{cases} \gamma_{i-1} - 2 & \text{if } i = \{\Delta_1, \dots, k_1 \Delta_1\} \\ \gamma_{i-1} & \text{otherwise} \end{cases} \quad (2)$$

with  $i = \{1, \dots, L\}$ ,  $\Delta_1 = 40$  and  $k_1 : k_1 \Delta_1 \leq L$

- *sublinear*: the number of neighboring buckets  $\gamma$  is reduced by 2 every 25 hash tables, but only after the first half of hash tables, i.e.:

$$\gamma_i = \begin{cases} \gamma_{i-1} & \text{if } i \leq L/2 \\ \gamma_{i-1} - 2 & \text{if } i = \{L/2 + \Delta_2, \dots, L/2 + k_2 \Delta_2\} \\ \gamma_{i-1} & \text{otherwise} \end{cases} \quad (3)$$

with  $i = \{1, \dots, L\}$ ,  $\Delta_2 = 25$  and  $k_2 : L/2 + k_2 \Delta_2 \leq L$

Symbol	Definition	Chosen value
$n$	number of images	-
$\delta$	hash dimension	$2^8 = 256$
$L$	number of hash tables	100
$\gamma_0$	initial gap	10
$l$	neighbors bucket	1-neighbors
$\varepsilon$	elements in the re-ranking list	250
-	reduction	sublinear

**Table 1.** Summary of notation.

The proposed approach contains several parameters. Their values were chosen after an extensive parameter analysis (out of the scope of this paper) and summary of notation is reported in Table 1.  $L$ ,  $\delta$  and  $l$  should be as low as possible since they directly affect the number of buckets  $\mathcal{N}_q^l$  to be checked and therefore the computational time at each query  $q$ , as follows:

$$\mathcal{N}_q^l = L \sum_{i=0}^l \binom{\gamma_i}{i} = L \sum_{i=0}^l \frac{(\gamma_i)!}{i! (\gamma_i - i)!} \quad (4)$$

where  $\gamma_i = \gamma_0 = \log_2 \delta, \forall i$  for standard BoI multi-probe LSH, whereas, in the case of BoI adaptive multi-probe LSH,  $\gamma_i$  can be computed using the eqs. 2 or 3.

## 4 Experimental results

The proposed approach has been extensively tested on public datasets in order to evaluate the accuracy against the state of the art.

Dataset	Size	Query images
Holidays [11] + Flickr1M	1 001 491	500
Oxford105k [20]	105 063	55
Paris106k [21]	106 392	55

**Table 2.** Datasets used in the experiments

#### 4.1 Datasets and evaluation metrics

The performance is measured on three public image datasets: Holidays+Flickr1M, Oxford105k and Paris106k as shown in Table 2.

**Holidays** [11] is composed by 1491 images representing the holidays photos of different locations, subdivided in 500 classes. The database images are 991 and the query images are 500, one for every class.

**Oxford5k** [20] is composed by 5062 images of Oxford landmarks. The classes are 11 and the queries are 55 (5 for each class).

**Paris** [21] is composed by 6412 images of landmarks of Paris, France. The classes are 11 and the queries are 55 (5 for each class).

**Flickr1M** [8] contains 1 million Flickr images used as distractors for Holidays, Oxford5k and Paris6k generating Holidays +Flickr1M, Oxford105k and Paris106k datasets.

**Evaluation.** Mean Average Precision (mAP) was used as metrics for accuracy.

**Distance.**  $L_2$  distance was employed to compare query images with the database.

**Implementation.** All experiments have been run on 4 separate threads. The CNN features used for the creation of locVLAD [16] descriptors are calculated on a NVIDIA GeForce GTX 1070 GPU mounted on a computer with 8-core and 3.40GHz CPU.

#### 4.2 Results on Holidays+Flickr1M datasets

This section reports the results of our approach, by adding to the Holidays dataset a different number of distractors, obtained from the Flickr1M dataset. All the experiments have been conducted several times and a mean has been computed in order to eliminate the randomness of the Gaussian distribution used in the hashing function. The embeddings used are locVLAD descriptors [16], while the features are extracted from the layer mixed8 of Inception V3 network [23] that is a CNN pre-trained on the ImageNet [4] dataset. The vocabulary used for the creation of locVLAD descriptors is calculated on Paris6k.

Table 3 summarizes the results on Holidays+Flickr1M dataset in terms of mAP and average retrieval time (msec). The first experiments evaluated only the top  $\epsilon = 250$  nearest neighbors.

LSH and multi-probe LSH achieve excellent results, but with an huge retrieval time. Also PP-index [2] needs more than 3 seconds for a query to retrieve the results. LOPQ[13] and FLANN [17] reach poor results on large-scale retrieval. LOPQ reached 36.37%, while FLANN achieved 83.97%. However, while query time for LOPQ is pretty low, FLANN is not able to keep the query time low. It is worth saying that both LOPQ and FLANN has been tested using the available codes from authors and reported results correspond to the best found configuration of parameters. Given the significantly

Method	$\epsilon$	Holidays+Flickr1M	
		mAP	avg retrieval time
LSH*	250	86.03%	3 103
Multi-probe LSH* ( $L = 50$ )	250	86.10%	16 706
PP-index* [2]	250	82.70%	2 844
LOPQ [13]	250	36.37%	4
FLANN [17]	250	83.97%	995
BoI LSH	250	78.10%	5
BoI multi-probe LSH	250	85.16%	12
BoI adaptive multi-probe LSH	250	85.35%	8
PP-index* [2]	10k	85.51%	15 640
LOPQ [13]	10k	67.22%	72
FLANN [17]	10k	85.66%	1 004
BoI adaptive multi-probe LSH	10k	<b>86.09%</b>	<b>16</b>

**Table 3.** Results in terms of mAP and average retrieval time in msec on Holidays+Flickr1M. \* indicates our re-implementation.

low (especially for LOPQ) performance in accuracy, further experiments have been conducted for LOPQ, FLANN, as well as PP-index and our method by increasing  $\epsilon$  from 250 to 10k. As foreseeable, all the accuracy results improved with respect to  $\epsilon = 250$  (LOPQ increases from 36.37% to 67.22%), but the proposed BoI adaptive multi-probe LSH method still outperforms all the others. Moreover, our method still results to be faster than the others (LOPQ is fast like ours, but with lower accuracy, while PP-index and FLANN are slightly lower in accuracy, but much slower).

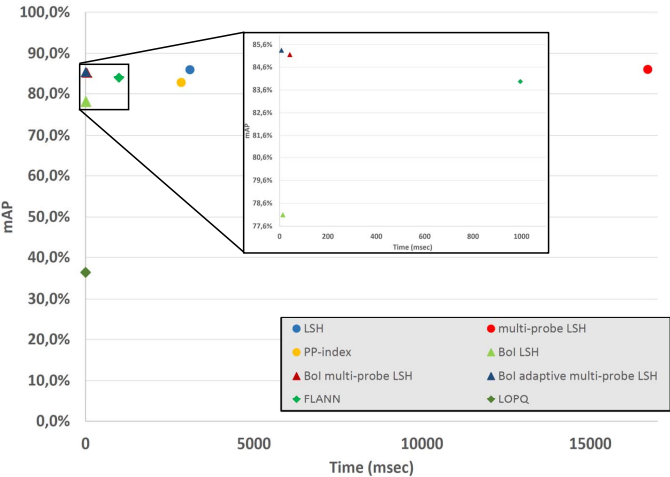
Overall speaking, our proposal outperforms all the compared methods in the trade-off between accuracy and efficiency. To better highlight this, Fig. 2 shows jointly the mAP (on y-axis) and the average query time (on x-axis). The best trade-off has to be found in the upper left corner of this graph, i.e. corresponding to high accuracy and low query time. All the BoI-based methods clearly outperform the other methods.

Regarding the memory footprint of the algorithm for 1M images with 1M descriptors of 128D (float = 4 bytes), brute-force approach requires 0.5Gb ( $1M \times 128 \times 4$ ). LSH needs only 100 Mb: 1M indexes for each of the  $L=100$  hash tables, because each indexes is represented by a byte (8 bit) and so  $1M \text{ indexes} \times 100 \text{ hash tables} \times 1 \text{ byte} = 100\text{Mb}$ . The proposed BoI only requires additional 4 Mb to store 1M weights.

### 4.3 Results on Oxford105k and Paris106k datasets

Since our goal is to execute large-scale retrieval for landmark recognition, we have also used the Oxford105k and Paris106k datasets. In this case, all the methods are tested using R-MAC descriptors, fine-tuned by Gordo *et al.* [6], since VLAD descriptors are demonstrated to be not suited for these datasets [14].

Table 4 show the mAP and the average retrieval time. Using  $\epsilon = 2500$ , the proposed approach obtained slightly worse results than PP-index, but resulted one order of magnitude faster in both datasets. When more top-ranked images are used ( $\epsilon = 10k$ ), BoI adaptive multi-probe LSH obtained the best results and with lower query time.



**Fig. 2.** Relationship between time and accuracy on Holidays+Flickr1M with different approaches.

Furthermore, LOPQ [13] works better on Paris106k than Oxford105k, while FLANN [17] performs poorly on both datasets.

5 Conclusions

In this paper, a novel multi-index hashing methods called Bag of Indexes (BoI) for approximate nearest neighbor search problem is proposed. This method demonstrated an overall better trade-off between accuracy and speed w.r.t. state-of-the-art methods on several large-scale landmark recognition datasets. Also, it works well with different embedding types (VLAD and R-MAC). The main future directions of our work will be

Method	ϵ	Oxford105k		Paris106k	
		mAP	avg ret. time	mAP	avg ret. time
LSH*	2500	80.83%	610	86.50%	607
PP-index* [2]	2500	81.89%	240	88.14%	140
LOPQ [13]	2500	71.90%	346	87.47%	295
FLANN [17]	2500	70.33%	2118	68.93%	2132
BoI adaptive multi-probe LSH	2500	81.44%	12	87.90%	13
PP-index* [2]	10k	82.82%	250	89.04%	164
LOPQ [13]	10k	69.94%	1153	88.00%	841
FLANN [17]	10k	69.37%	2135	70.73%	2156
BoI adaptive multi-probe LSH	10k	<b>84.38%</b>	<b>25</b>	<b>92.31%</b>	<b>27</b>

**Table 4.** Results in terms of mAP and average retrieval time (msec) on Oxford105k and Paris106k. \* indicates our re-implementation of the method.



related to reduce the dimension of the descriptor in order to speed the creation of bucket structure and to adapt the proposed method for dataset with billions of elements.

**Acknowledgments.** This work is partially funded by Regione Emilia Romagna under the “Piano triennale alte competenze per la ricerca, il trasferimento tecnologico e l’imprenditorialità”.

## References

1. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: European conference on computer vision, Springer (2014) 584–599
2. Chavez, E., Figueroa, K., Navarro, G.: Effective proximity retrieval by ordering permutations. *IEEE transactions on Pattern Analysis and Machine Intelligence* **30** (2008) 1647–1658
3. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: *Proceedings of the twentieth annual symposium on Computational geometry*, ACM (2004) 253–262
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE (2009) 248–255
5. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: Learning global representations for image search. In: *European Conference on Computer Vision*, Springer (2016) 241–257
6. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision* **124**(2) (2017) 237–254
7. Greene, D., Parnas, M., Yao, F.: Multi-index hashing for information retrieval. In: *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, IEEE (1994) 722–731
8. Huiskes, M.J., Lew, M.S.: The mir flickr retrieval evaluation. In: *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, ACM (2008) 39–43
9. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, ACM (1998) 604–613
10. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* **33**(1) (2011) 117–128
11. Jégou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometry consistency for large scale image search-extended version. (2008)
12. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. *CVPR* (2010) 3304–3311
13. Kalantidis, Y., Avrithis, Y.: Locally optimized product quantization for approximate nearest neighbor search. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2014) 2321–2328
14. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high-level semantics. *Pattern recognition* **40**(1) (2007) 262–282
15. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-probe LSH: efficient indexing for high-dimensional similarity search. In: *Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment* (2007) 950–961
16. Magliani, F., Bidgoli, N.M., Prati, A.: A location-aware embedding technique for accurate landmark recognition. *ICDSC* (2017)
17. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. *IEEE transactions on pattern analysis and machine intelligence* **36**(11) (2014) 2227–2240

18. Norouzi, M., Punjani, A., Fleet, D.J.: Fast search in hamming space with multi-index hashing. In: *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on, IEEE (2012) 3108–3115
19. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, IEEE (2010) 3384–3391
20. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2007)
21. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in alrge scale image databases. *CVPR* (2008)
22. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. *ICCV* **2** (2003) 1470–1477
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2016) 2818–2826
24. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879* (2015)
25. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: *Advances in neural information processing systems*. (2009) 1753–1760
26. Yue-Hei Ng, J., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. (2015) 53–61