

Bags of Local Convolutional Features for Scalable Instance Search

Eva Mohedano*, Kevin McGuinness and
Noel E. O'Connor
Insight Center for Data Analytics
Dublin City University
Dublin, Ireland
eva.mohedano@insight-centre.org

Amaia Salvador*, Ferran Marqués,
and Xavier Giró-i-Nieto
Image Processing Group
Universitat Politècnica de Catalunya
Barcelona, Catalonia/Spain
{amaia.salvador, xavier.giro}@upc.edu

ABSTRACT

This work proposes a simple instance retrieval pipeline based on encoding the convolutional features of CNN using the bag of words aggregation scheme (BoW). Assigning each local array of activations in a convolutional layer to a visual word produces an *assignment map*, a compact representation that relates regions of an image with a visual word. We use the assignment map for fast spatial reranking, obtaining object localizations that are used for query expansion. We demonstrate the suitability of the BoW representation based on local CNN features for instance retrieval, achieving competitive performance on the Oxford and Paris buildings benchmarks. We show that our proposed system for CNN feature aggregation with BoW outperforms state-of-the-art techniques using sum pooling at a subset of the challenging TRECVID INS benchmark.

General Terms

Algorithms

Keywords

Instance Retrieval, Convolutional Neural Networks, Bag of Words

1. INTRODUCTION

Convolutional neural networks (CNNs) have been demonstrated to produce global representations that effectively capture the semantics in images. Such CNN-based representations [2, 3, 15, 16, 20, 21] have improved upon the state-of-the-art for instance retrieval.

Despite CNN-based descriptors performing remarkably well in retrieval benchmarks like Oxford and Paris Buildings, state-of-the-art solutions for more challenging datasets such as TRECVID Instance Search (INS) have not yet adopted

*denotes equal contribution.



Figure 1: Examples of the top generated rankings for queries of three different datasets: Oxford (top), Paris (middle), TRECVID INS 2013 (bottom).

pipelines that depend solely on CNN features. Current INS systems [11, 22, 23, 24] are still based on aggregating local hand-crafted features using bag of words (BoW) encoding [18] to produce very high-dimensional sparse image representations. Such high-dimensional sparse representations are more likely to be linearly separable, while having relatively few non-zero elements makes them efficient both in terms of storage and computation.

Many successful image retrieval engines combine an initial highly-scalable ranking mechanism on the full image database with a more computational demanding yet higher-precision reranking mechanism (e.g. geometric verification) applied to the top retrieved items.

Inspired by advances in CNN-based descriptors for image retrieval, yet still focusing on instance search, we revisit the BoW encoding scheme using local features from convolutional layers of a CNN. This work presents three contributions:

- We propose a sparse visual descriptor based on a Bag of Local Convolutional Features (BLCF), which allows fast image retrieval by means of an inverted index.
- We introduce the assignment map as a new compact representation of the image, which maps pixels in the image to their corresponding visual words, allowing the fast composition of a BoW descriptor for any region of the image.
- We take advantage of the scalability properties of the assignment map to perform a local analysis of multiple image regions for reranking, followed by a query expansion stage using the obtained object localizations.

Using this approach, we present a retrieval system that achieves state-of-the-art performance in several instance retrieval benchmarks. Figure 1 illustrates some of the rankings produced by our system on several datasets.

2. RELATED WORK

Many works in the literature have proposed CNN-based representations for image retrieval. Early works [3, 15] focused on replacing traditional hand-crafted descriptors with features from fully connected layers of a CNN pre-trained for image classification. A second generation of works reported significant gains in performance when switching from fully connected to either sum [2, 7] or max [16, 20] pooled convolutional features. Our work shares similarities with all the former in that we use convolutional features extracted from a pre-trained CNN. Unlike these approaches, however, we propose using a sparse, high-dimensional encoding that better represents local image features.

Several works have tried to exploit local information in images by passing multiple image sub patches through a CNN to obtain local features from either fully connected [15, 9] or convolutional [4] layers, which are in turn aggregated using techniques like average pooling [15], BoW [9], or VLAD [4]. Although many of these methods perform well in retrieval benchmarks, they require CNN feature extraction from many image patches, which slows down indexing and feature extraction at retrieval time. Instead, other works [10, 1] treat the activations of the different neuron arrays across all feature maps in a convolutional layer as local features. This way, a single forward pass of the entire image through the CNN is enough to obtain the activations of its local patches, which are then encoded using VLAD. Our approach is similar to the ones in [10, 1] in that we also treat the features in a convolutional layer as local features. We, however, use BoW encoding instead of VLAD to take advantage of sparse representations for fast retrieval in large-scale databases.

Some of the approaches cited above propose systems that are based or partially based on spatial search over multiple regions of the image. Razavian et al. [16] achieve a remarkable increase in performance by applying a spatial search strategy over an arbitrary grid of windows at different scales. Although they report high performance in several retrieval benchmarks, their proposed approach is very computationally costly and does not scale well to larger datasets and real-time search scenarios. Tolia et al. [20] introduce a local analysis of multiple image patches, which is only applied to the top elements of an initial ranking. They propose an efficient workaround for sub patch feature pooling based on integral images, which allows them to quickly evaluate many image windows. Their approach improves their baseline ranking and also provides approximate object localizations. In this direction, our work proposes using the assignment map to quickly build the BoW representation of any image patch, which allows us to apply a spatial search for reranking. Unlike [20], we use the object localizations obtained with spatial search to mask out the activations of the background and perform query expansion using the detected object location.

3. BAG OF WORDS FRAMEWORK

The proposed pipeline for feature extraction uses the activations at different locations of a convolutional layer in a pre-trained CNN as local features. Similar to [20, 7], we

discard the softmax and fully connected layers of the original network while keeping the original image aspect ratio. Each convolutional layer in the network has D different $N \times M$ feature maps, which can be viewed as $N \times M$ local descriptors of dimension D .

We propose to use bag of words encode the local convolutional features of an image into a single vector. Although more elaborate aggregation strategies have been shown to outperform BoW-based approaches for some tasks in the literature [5, 12], Bag of words encodings produce sparse high-dimensional codes that can be stored in inverted indices, which are beneficial for fast retrieval. Moreover, BoW-based representations are more compact, faster to compute and easier to interpret.

Bag of words models require constructing a visual codebook to map vectors to their nearest centroid. We use k -means on local CNN features to fit this codebook. Each local CNN feature in the convolutional layer is then assigned its closest visual word in the learned codebook. This procedure generates the *assignment map*, i.e. a 2D array of size $N \times M$ that relates each local CNN feature with a visual word. The assignment map is therefore a compact representation of the image which relates each pixel of the original image with its visual word with a precision of $(\frac{W}{N}, \frac{H}{M})$ pixels, where W and H are the width and height of the original image. This property allows us to quickly generate the BoW vectors of not only the full image, but also its parts. We describe the use of this property in our work in Section 4.

Figure 2 shows the pipeline of the proposed approach, which encodes the image into a sparse high dimensional descriptor to be used for instance retrieval.

4. IMAGE RETRIEVAL

This section describes the image retrieval pipeline, which consists of an initial ranking stage, followed by a spatial reranking, and query expansion.

(a) **Initial search:** The initial ranking is computed using the cosine similarity between the BoW vector of the query image and the BoW vectors of the full images in the database. We use a sparse matrix based inverted index and GPU-based sparse matrix multiplications to allow fast retrieval. The image list is then sorted based on the cosine similarity of its elements to the query. We use two types of image search based on the query information that is used:

- Global search (GS): The BoW vector of the query is built with the visual words of all the local CNN features in the convolutional layer extracted for the query image.
- Local search (LS): The BoW vector of the query contains only the visual words of the local CNN features that fall inside the query bounding box.

(b) **Local reranking (R):** After the initial search, the top T images in the ranking are reranked based on a localization score. We choose windows of all possible combinations of width $w \in \{W, \frac{W}{2}, \frac{W}{4}\}$ and height $h \in \{H, \frac{H}{2}, \frac{H}{4}\}$, where W and H are the width and height of the assignment map. We use a sliding window strategy directly on the assignment map with 50% of overlap in both directions.

We additionally perform a simple filtering strategy to discard those windows whose aspect ratio is too different to the aspect ratio of the query. Let the aspect ratio of the query bounding box be $AR_q = \frac{W_q}{H_q}$ and $AR_w = \frac{W_w}{H_w}$ be

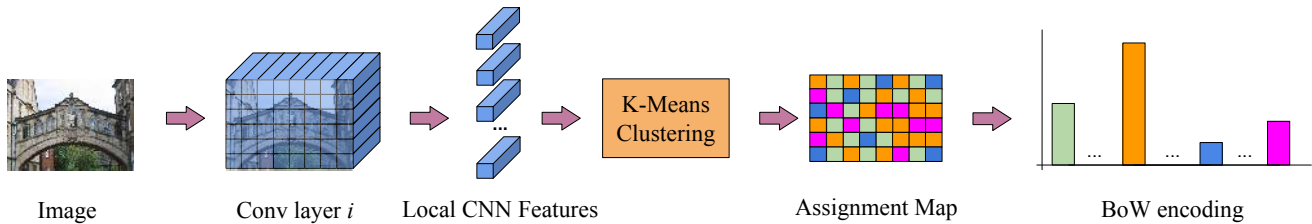


Figure 2: The Bag of Local Convolutional Features pipeline (BLCF).

the aspect ratio of the window. The score for window w is defined as $score_w = \frac{\min(AR_w, AR_q)}{\max(AR_w, AR_q)}$. All windows with a score lower than a threshold th are discarded.

For each of the remaining windows, we construct the BoW vector representation and compare it with the query representation using cosine similarity. The window with the highest cosine similarity is picked as the object localization and its score is kept for the image. We also enhance the BoW window representation with spatial pyramid matching [8] with $L = 2$ resolution levels (i.e. the full window and its 4 sub regions). We construct the BoW representation of all sub regions at the 2 levels, and weight their contribution to the similarity score with inverse proportion to the resolution level of the region. The cosine similarity of a sub region r to the corresponding query sub region is therefore weighted by $w_r = \frac{1}{2^{(L-l_r)}}$, where l_r is the resolution level of the region r . With this procedure, the top T elements of the ranking are sorted based on the cosine similarity of their regions to the query’s, and also provides the region with the highest score as a rough localization of the object.

(c) **Query expansion** We investigate two query expansion strategies based on global and local BoW descriptors:

- Global query expansion (GQE): The BoW vectors of the N images at the top of the ranking are averaged together with the BoW of the query to form the new representation for the query.
- Local query expansion (LQE): Locations obtained in the local reranking step are used to mask out the background and build the BoW descriptor of only the region of interest of the N -top images in the ranking. These BoW vectors are averaged together with the BoW of the query bounding box to perform a second search.

5. EXPERIMENTS

5.1 Preliminary experiments

We experiment with three different instance retrieval benchmarks to evaluate the performance of our approach: Oxford Buildings [13], Paris Buildings [14] and a subset of TRECVID Instance Search 2013 (INS 23k) [19] containing only those keyframes that are relevant to at least one of the queries. For the Paris and Oxford datasets, we also test the performance when adding 100,000 distractor images collected from Flickr to the original datasets (resulting in the Oxford 105k and Paris 106k datasets, respectively).

Feature extraction was performed using *Caffe* [6] and the VGG16 pre-trained network [17]. We extracted features from the last three convolutional layers (conv5_1, conv5_2 and conv5_3) and compared their performance on the Oxford 5k dataset. We experimented on different image input sizes:

Table 1: mAP on Oxford 5k and Paris 6k for the different stages in the pipeline introduced in Section 4.

		Baseline	+R	+GQE	+R +GQE	+R +LQE
Oxford	GS	0.653	0.701	0.702	0.771	0.782
	LS	0.738	0.734	0.773	0.769	0.788
Paris	GS	0.699	0.719	0.774	0.801	0.835
	LS	0.820	0.815	0.814	0.807	0.848

1/3 and 2/3 of the original image. Following several other authors [2, 7], we l_2 -normalize all local features, followed by PCA, whitening, and a second round of l_2 -normalization. The PCA models were fit on the same dataset as the test data in all cases. Unless stated otherwise, all experiments used a visual codebook of 25,000 centroids fit using the (L^2 -PCA- L^2 transformed) local CNN features of all images in the same dataset.

We apply bilinear interpolation on the convolutional layers to obtain higher resolution maps as a workaround of using bigger input size image. We find this strategy to be beneficial for all layers tested, achieving the best results when using conv5_1. Inspired by the boost in performance of the Gaussian center prior in SPoC features [2], we apply a weighting scheme on the visual words to provide more importance to those belonging to the center of the image.

5.2 Reranking and query expansion

We apply the local reranking (R) stage on the top-100 images in the initial ranking, using the sliding window approach described in Section 4. The presented aspect ratio filtering is applied with a threshold $th = 0.4$, which was chosen based on a visual inspection of results on a subset of Oxford 5k. Query expansion is later applied considering the top-10 images of the resulting ranking. Table 1 contains the results for the different stages in the pipeline.

The results indicate that the local reranking is significantly beneficial only when applied to a ranking obtained from a search using the global BoW descriptor of the query image (GS). This is consistent with the work by Tolias et al. [20], who also apply a spatial reranking followed by query expansion to a ranking obtained with a search using descriptors of full images. They achieve a mAP of 0.66 in Oxford 5k, which is increased to 0.77 after spatial reranking and query expansion, while we reach similar results (e.g. from 0.652 to 0.769). However, our results indicate that a ranking originating from a local search (LS) does not benefit from local reranking. Since the BoW representation allows us to effectively perform a local search (LS) in a database of full indexed images, we find the local reranking stage applied to LS to be redundant in terms of the achieved quality of the ranking. However, the local reranking stage does provide

Table 2: Comparison to state-of-the-art CNN representations (mAP). Results in the lower section consider reranking and/or query expansion.

	Oxford		Paris	
	5k	105k	6k	106k
Ng <i>et al.</i> [10]	0.649	-	0.694	-
Razavian <i>et al.</i> [16]	0.844	-	0.853	-
SPoC [2]	0.657	0.642	-	-
R-MAC [20]	0.668	0.616	0.830	0.757
CroW [7]	0.682	0.632	0.796	0.710
uCroW [7]	0.666	0.629	0.767	0.695
GS	0.652	0.510	0.698	0.421
LS	0.739	0.593	0.820	0.648
CroW + GQE [7]	0.722	0.678	0.855	0.797
R-MAC + R + GQE [20]	0.770	0.726	0.877	0.817
LS + GQE	0.773	0.602	0.814	0.632
LS + R + LQE	0.788	0.651	0.848	0.641

with a rough localization of the object in the images of the ranking, as depicted in Figure 1. We use this information to perform query expansion based on local features (LQE).

Results indicate that query expansion stages greatly improve performance in Oxford 5k. We do not observe significant gains after reranking and QE in the Paris 6k dataset, although we achieve our best result with LS + R + LQE.

5.3 Comparison with the state-of-the-art

We compare our approach with other CNN-based representations that make use of features from convolutional layers on the Oxford and Paris datasets. Table 2 includes the best result for each approach in the literature. Our performance using global search is comparable to that of Ng *et al.* [10], which is the one that most resembles our approach. However, they achieve this result using VLAD features, which are more expensive to compute and, being a dense high-dimensional representation, do not scale as well to larger datasets. Similarly, Razavian *et al.* [16] achieve the highest performance of all approaches in both the Oxford and Paris benchmarks by applying a spatial search at different scales for all images in the database. Such approach is prohibitively costly when dealing with larger datasets, especially for real-time search scenarios. Our BoW-based representation is highly sparse, allowing for fast retrieval using inverted indices, and achieves consistently high mAP in all tested datasets.

We also compare our local reranking and query expansion results with similar approaches in the state-of-the-art. The authors of R-MAC [20] apply a spatial search for reranking, followed by a query expansion stage, while the authors of CroW [7] only apply query expansion after the initial search. Our proposed approach also achieves competitive results in this section, achieving the best result for Oxford 5k.

5.4 Experiments on TRECVID INS

In this section, we compare the Bag of Local Convolutional Features (BLCF) with the sum pooled convolutional features proposed in several works in the literature. We use our own implementation of the *uCroW* descriptor from [7] and compare it with BLCF for the TRECVID INS subset. For the sake of comparison, we test our implementation of sum pooling using both our chosen CNN layer and input size (conv5_1 and 1/3 image size), and the ones reported in [7] (pool5 and full image resolution). For the BoW representa-

tion, we train the codebook using 3M local CNN features chosen randomly from the INS subset. In this case, we do not apply center prior to the feature to avoid down weighting local features from image areas where the objects might appear. Table 3 compares sum pooling with BoW in Oxford, Paris, and TRECVID subset datasets. As stated in earlier sections, sum pooling and BoW have similar performance in Oxford and Paris datasets. For the TRECVID INS subset, however, Bag of Words significantly outperforms sum pooling, which demonstrates its suitability for challenging instance search datasets, in which queries are not centered and have variable size and appearance. We also observe a different behavior when using the provided query object locations (LS) to search, which was highly beneficial in Oxford and Paris datasets, but does not provide any gain in TRECVID INS. We hypothesize that the fact that the size of the instances is much smaller in TRECVID than in Paris and Oxford datasets causes this drop in performance. Global search (GS) achieves better results on TRECVID INS, which suggests that query instances are in many cases correctly retrieved due to their context.

Table 3: mAP of sum pooling and BoW aggregation techniques in Oxford, Paris and TRECVID INS subset.

		Oxford 5k	Paris 6k	INS 23k
BoW	GS	0.650	0.698	0.323
	LS	0.739	0.819	0.295
Sum pooling (as ours)	GS	0.606	0.712	0.156
	LS	0.583	0.742	0.097
Sum pooling (as in [7])	GS	0.672	0.774	0.139
	LS	0.683	0.763	0.120

6. CONCLUSION

We proposed an aggregation strategy based on Bag of Words to encode features from convolutional neural networks into a sparse representations for instance search. We achieved competitive performance with respect to other CNN-based representations in Oxford and Paris benchmarks, while being more scalable in terms of index size, cost of indexing, and search time. We also compared our BoW encoding scheme with sum pooling of CNN features in the far more challenging TRECVID instance search task, and demonstrated that our method consistently and significantly performs better. Our method does, however, appear to be more sensitive to large numbers of distractor images than methods based on sum and max pooling. We speculate that this may be because the distractor images are drawn from a different distribution to the original dataset, and may therefore require a larger codebook to better represent the diversity in the visual words.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under grant number SFI/12/RC/2289, project BigGraph TEC2013-43935-R, funded by the Spanish Ministerio de Economía y Competitividad and the European Regional Development Fund (ERDF), as well as the donation of a GeForce GTX Titan X from NVIDIA Corporation.

7. REFERENCES

- [1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. *arXiv:1511.07247*, 2015.
- [2] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *International Conference on Computer Vision (ICCV)*, December 2015.
- [3] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *Computer Vision–ECCV 2014*. 2014.
- [4] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Computer Vision–ECCV 2014*. 2014.
- [5] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, 2014.
- [7] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. *arXiv:1512.04065*, 2015.
- [8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [9] Y. Liu, Y. Guo, S. Wu, and M. S. Lew. Deepindex for accurate and efficient image retrieval. In *International Conference on Multimedia Retrieval (ICMR)*, 2015.
- [10] J. Ng, F. Yang, and L. Davis. Exploiting local features from deep networks for image retrieval. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015.
- [11] V.-T. Nguyen, D.-L. Nguyen, M.-T. Tran, D.-D. Le, D. A. Duong, and S. Satoh. Query-adaptive late fusion with neural network for instance search. In *International Workshop on Multimedia Signal Processing (MMSP)*, 2015.
- [12] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [15] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014.
- [16] A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. A baseline for visual instance retrieval with deep convolutional networks. In *International Conference on Learning Representations. ICLR*, 2015.
- [17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [18] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4), 2009.
- [19] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *International Workshop on Multimedia Information Retrieval (MIR)*, 2006.
- [20] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of CNN activations. *arXiv preprint arXiv:1511.05879*, 2015.
- [21] L. Xie, Q. Tian, R. Hong, and B. Zhang. Image classification and retrieval are one. In *International Conference on Multimedia Retrieval (ICMR)*, 2015.
- [22] W. Zhang and C.-W. Ngo. Topological spatial verification for instance search. *IEEE Transactions on Multimedia*, 17(8), 2015.
- [23] X. Zhou, C.-Z. Zhu, Q. Zhu, S. Satoh, and Y.-T. Guo. A practical spatial re-ranking method for instance search from videos. In *International Conference on Image Processing (ICIP)*, 2014.
- [24] C.-Z. Zhu and S. Satoh. Large vocabulary quantization for searching instances from videos. In *International Conference on Multimedia Retrieval (ICMR)*, 2012.