Scalar Quantization-Based Text Encoding for Large Scale Image Retrieval

(DISCUSSION PAPER)

Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Fausto Rabitti, and Lucia Vadicamo

Institute of Information Science and Technologies (ISTI), CNR, Pisa
firstname.lastname@isti.cnr.it

Abstract. The great success of visual features learned from deep neural networks has led to a significant effort to develop efficient and scalable technologies for image retrieval. This paper presents an approach to transform neural network features into text codes suitable for being indexed by a standard full-text retrieval engine such as Elasticsearch. The basic idea is providing a transformation of neural network features with the twofold aim of promoting the sparsity without the need of unsupervised pre-training. We validate our approach on a recent convolutional neural network feature, namely Regional Maximum Activations of Convolutions (R-MAC), which is a state-of-art descriptor for image retrieval. An extensive experimental evaluation conducted on standard benchmarks shows the effectiveness and efficiency of the proposed approach and how it compares to state-of-the-art main-memory indexes.

Keywords: Image retrieval · Deep Features · Inverted index

1 Introduction

Full-text search engines on the Web have achieved great results in terms of efficiency thanks to the use of inverted index technology. In the last years, we experienced an increasing interest in the retrieval of other forms of expression, such as images; nevertheless, the development in those cases was not as rapid as text-based paradigms.

In the field of image retrieval, since 2014 we have witnessed a great development of learned features obtained by neural networks, in particular Convolutional Neural Networks (CNN), which have emerged as effective image descriptors. Differently from text, in which inverted indexes perfectly marry the sparse document representation in standard vector models, learned image descriptors

tend to be dense and compact, thus making directly unfeasible the usage of mature text-tailored index technologies. While efficient index structures for this type of data exist [13,14], they usually come with caveats that prevent their usage in very large-scale scenarios, such as main-memory-only implementations and computationally expensive indexing or codebook-learning phases.

This paper summarizes the main contributions presented in [2], where we explored new approaches to make image retrieval as similar as possible to text retrieval so as to reuse the technologies and platforms exploited today for text retrieval without the need for dedicated access methods. In a nutshell, the idea is to use image representations extracted from a CNN (*deep features*) and to transform them into text so that they can be indexed with a standard text search engine. In particular, we propose a *Scalar Quantization* approach that transforms deep features, which are (dense) vectors of real numbers, into sparse vectors of integer numbers. The components of these integer vectors are then translated to "term frequencies" of synthetic textual documents. Sparseness is necessary for efficiency issues to achieve sufficient levels of efficiency exactly as it does for search engines for text documents.

We consider the problem of image retrieval in a large-scale context, with an eye to scalability. This aspect is often overlooked by the literature, most of the image retrieval systems are designed to work in main memory and many of these cannot be distributed across a cluster of nodes. Many techniques present in literature try to tackle this problem by heavily compressing the representation of visual features to adapt more and more data to the secondary memory. However, these approaches are not able to scale because sooner or later response times become unacceptable as the size of the data to be managed increases.

2 Related Work

In the last years, image features extracted using deep CNNs have been widely employed as effective image descriptors. *Deep features* achieved state-of-the-art results in several vision tasks, including image retrieval [6,5] and object detection [8]. As a consequence, there is an increasing interest in identifying techniques able to efficiently index and search large set of deep features.

To frame our work in the context of scientific literature, we focus on techniques that deal with emerging deep features using an inverted index. Liu et al. [13] proposed a framework that adapts the BoW model and inverted table to index deep features. However, it needs to learn a large visual dictionary when dealing with a large-scale dataset. Other works treat the features in a convolutional layer as local features by using aggregation schemes like BoW and VLAD [4,19] or try to quantize the deep features using a codebook [14]. Jegou et al. [11] proposed an approximate nearest neighbor algorithm based on product quantization (PQ), which exploits an inverted index. In PQ, the original vector is divided into M sub-vectors that are independently quantized. A codebook is learned via k-means for each of the M sub-division, and each sub-vector is compressed by

storing the nearest centroid index. An implementation of PQ-compressed inverted indexes, denoted IVFPQ, is available in the FAISS library.

In our work, we propose a novel approach to generate sparse representations of deep features that can be employed to efficiently index and search the deep features by using inverted files. Without loss of generality, we use the Regional Maximum Activations of Convolutions (R-MAC) feature vector defined by Gordo et al. [9] as representative of the family of deep-learned dense real-valued representations for instance-level image retrieval. This kind of features poses challenges to the current technique of sparse encoding when dealing with non-sparse and real-valued feature vectors that this work aims to tackle. On the other hand, other types of deep features typically used in this field, such as ReLU-ed features extracted from pretrained image classifiers, have already been explored and indexed with a similar approach in a preliminary work [3].

3 Scalar Quantization

In this section, we present a novel approach to generate sparse representations for *D-dimensional vectors* originally compared with the dot product. In particular, we focus on R-MAC descriptors, which are real-valued dense vectors that are particularly powerful for applications of instance-level and content-based image retrieval. However, our method can be adapted to general euclidean vectors.

In a nutshell, we aim to define a transformation $f : \mathbb{R}^D \to \mathbb{N}^n$ that generate sparse vectors and preserves the similarities of objects as much as possible. The reason why we want to transform real-valued dense vectors into sparse vectors of natural numbers is because we want to use a full-text search engine to index the vectors so transformed. In fact, a full-text search engine based on the *vector space model* [16] generates a vector representation of a text via term frequencies, i.e. the number of occurrences of the words in it. These systems transform the texts into vector representations using the well-known TF scheme and practically use the dot product as a function of similarity between vectors. So, the abstract transformation $f(\cdot)$ represents a function that exactly generates the vectors that are internally represented by the search engine in the case of the simple term-weighting scheme. In other words, given a dictionary of n codewords we transform an object o into a synthetic text encoding t_o that is obtained as a space-separated concatenation of codewords so that the i-th codeword is repeated a number of times equal to the i-th element of the vector $f(o)$. Using this representation, the search engine indexes the text by using inverted files, i.e. each object o is stored in the posting lists associated to the codewords appearing in the text representation of o. The number of posting lists equals the number of codewords of the considered vocabulary. This approach is known in the literature as Surrogate Text Representation [1,7].

The idea behind our *Scalar Quantization* approach is to map the real-valued vector components independently into a smaller set of integer values which act as the term frequencies of a predefined set of codewords. The first step is applying a transformation to the vectors that helps preventing the presence of unbalanced

posting lists in the inverted file (thus important for efficiency of inverted indices). To understand why, note that each component of the vectors is associated with a posting list storing the *id* of the vector and the value of the component itself, if nonzero. Therefore, if on average some component is nonzero for many data vectors then the corresponding posting list will be accessed many times, provided that the queries follow the same distribution of the data. The ideal case occurs when the component share exactly the same distribution (same mean and variance is sufficient). To this end, we apply a random orthogonal transformation to the entire set of data vectors, which is known to provide good balancing for high dimensional vectors without the need to search for an optimal balancing transformation [12]. An important aspect of the orthogonal transformation is that it preserves the ranking when we search using the kNN approach by ordering the vectors on the basis of their Euclidean distance to the query. Moreover, if applying the orthogonal transformation and the mean centering to all the data objects and just the orthogonal transformation to the query, we have an ordering preserving transformation with respect to the dot-product (see [2] for further details). Thus, our preprocessing step is defined as:

$$\boldsymbol{v} \rightarrow R(\boldsymbol{v} - \boldsymbol{\mu}) \tag{1}$$

$$\boldsymbol{q} \rightarrow R\,\boldsymbol{q} \tag{2}$$

where R is a *random* orthogonal matrix and $\boldsymbol{\mu} \in \mathbb{R}^D$ is set to center the data to zero mean. The next step is transforming the rotated vectors into term frequency vectors. We do it by quantizing the vectors so that posting entries will contain numeric values proportional to the float values of the deep feature entries. Specifically, we use the transformation $\boldsymbol{w} \rightarrow \lfloor s\boldsymbol{w} \rfloor$ where $\lfloor \cdot \rfloor$ denotes the floor function and s is a multiplication factor > 1.

The approach presented so far is intended to encode a vector of real numbers into a vector of integers preserving as much as possible the order with respect to the dot product. However, this approach does not solve the problem that in most cases these vectors are dense, which leads to low efficiency when using inverted files to index textual documents. To sparsify the term frequency vectors, that is to discard their less significant components, we must accept a further loss in precision. To achieve this, we propose to keep components above a certain threshold $1/\gamma$ and zeroing the others. The parameter $\gamma \in \mathbb{N}$ controls the sparseness of the thresholded feature. This approach is optimal when we have many components near or equal to zero; thus, we exploit the previously defined transformation (Eq. 1) to center the mean values of each dimension to zero.

To sum up, our proposed transformation is $f : \boldsymbol{v} \mapsto g_\gamma(\lfloor sR(\boldsymbol{v}-\boldsymbol{\mu}) \rfloor)$, where g_γ is a component-wise thresholding function, i.e. $g_\gamma(x) = x$ if $x > 1/\gamma$, 0 otherwise.

Dealing with negative values In order to index R-MAC features and represent them in the vector space model of text retrieval, we encode each dimension of the R-MAC features as a different codeword, and we use the TF field to represent a single value of our feature vector. However, TF must be positive (most search engine admits positive-only TFs even if this in principle would be possible),

nonetheless, both negative and positive elements contribute to informativeness. In the scalar quantization approach presented above, the negative values are flatted to zero. Naive techniques such as taking the absolute value result in a degraded performance due to respectively loss or aliasing of information. In order to prevent this imbalance towards positive activations at the expense of negative ones, we use the Concatenated Rectified Linear Unit (CReLU) transformation [17]. It simply makes an identical copy of vector elements, negate it, concatenate both original vector and its negation, and then apply ReLU altogether. More precisely the CReLU of the vector $\boldsymbol{v}$ is defined as $\boldsymbol{v}^+ = ReLU([\boldsymbol{v}, -\boldsymbol{v}])$, where the $ReLU(\cdot) = max(\cdot, 0)$ is applied element-wise. After applying CReLU, we apply the transformation f to $\boldsymbol{v}^+$ as described in the previous section.

4 Experimental Evaluation

To assess the performance of our Scalar Quatization technique in content-based image retrieval task, we performed an extensive experimental evaluation on two instance-level retrieval benchmarks. A complete presentation of the results is given in [2]; here we report some of the most relevant results.

The experiments were conducted on two benchmarks. *INRIA Holidays* [10] is a collaction of 1,491 images representing a large variety of scene type and 500 queries for which result lists are provided. Usually this benchmark is extended with a distractor dataset, namely MIRFlickr1M[1], that contains 1M images. *Oxford Buildings* [15] is composed of 5,062 images of 11 Oxford landmarks downloaded from Flickr. A manually labeled groundtruth is available for five queries for each landmark, for a total of 55 queries. As for INRIA Holidays, we merged the dataset with the distraction dataset Flickr100k including 100k images[2].

We used the ResNet-101 trained model as an R-MAC feature extractor, which has been shown to achieve the best performance on standard benchmarks of instance-level image retrieval. We extracted the R-MAC features using fixed grid regions at two different scales as proposed in [18]. Then, we produced the sparse representations by using different sparsification thresholds γ.

We compared the performance of our approach with FAISS[3], which includes state-of-the-art approximate nearest neighbor algorithms based on PQ [11]. Note that PQ constitutes a suitable competitor to our approach since is a one-stage inverted-file-based index as the one used in standard textual search engines. For a fair comparison, we used the configuration for FAISS that gives the best effectiveness-efficiency trade-off for each dataset (see [2] for further details). PQ-based methods need a training set and an offline training phase to initialize the inverted index. We examined two possible scenarios dubbed IVFPQ and IVFPQ*. In the former, the index is trained on the data to be indexed, while in the latter, a set of unrelated images (the T4SA dataset[4]) is used as training set.

[1] `http://press.liacs.nl/mirflickr/`
[2] `http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/`
[3] `https://github.com/facebookresearch/faiss`
[4] `http://t4sa.it`

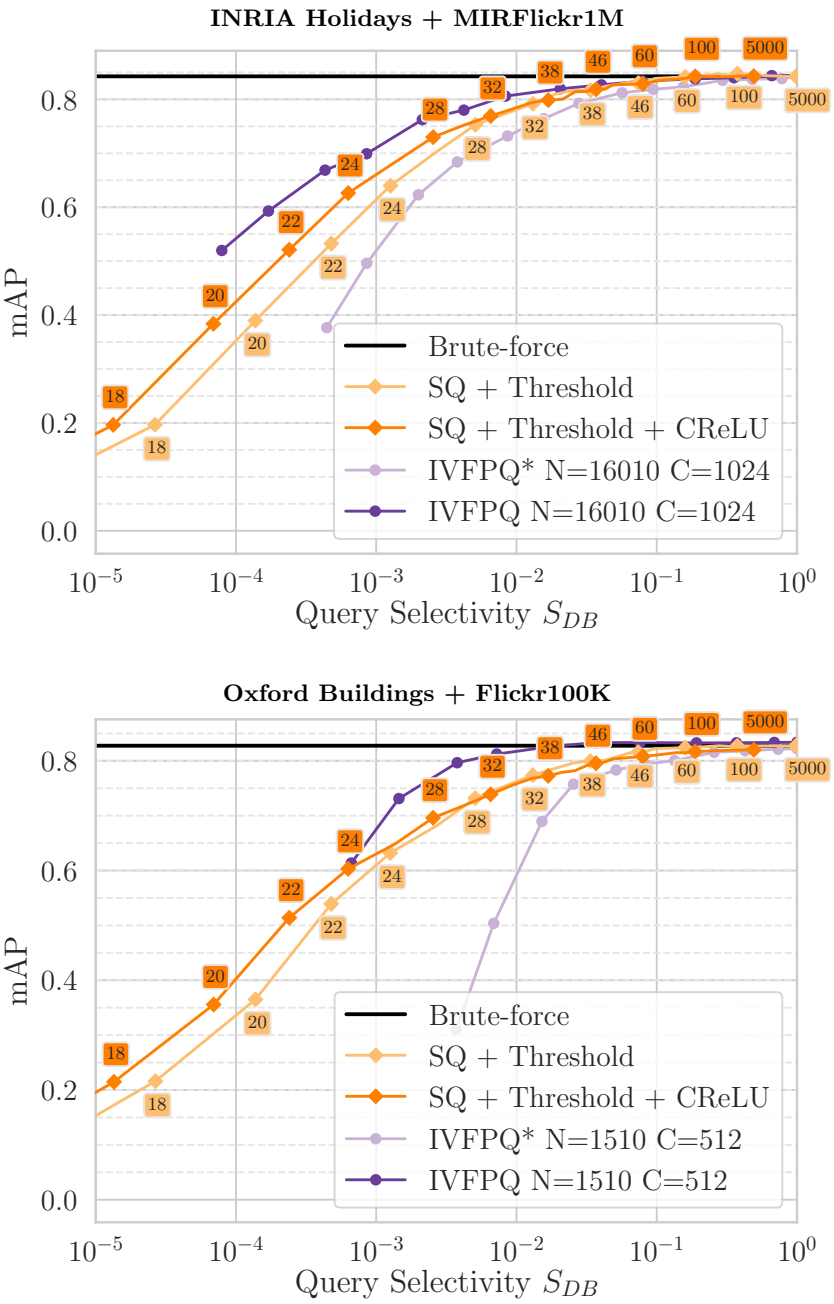

Fig. 1. Effectiveness (mAP) vs efficiency (S_{DB} — fraction of dataset accessed) trade-offs of Scalar Quantization based (SQ) and Product Quantization based (IVFPQ) methods. Curves are produced varying γ for SQ (reported near each point) and the number of accessed lists (nprobe) for IVFPQ. Brute-force represents the sequential scan baseline. IVFPQ* represents IVFPQ trained on out-of-distribution images.

To assess the quality of the results, we used the *mean Average Precision* (mAP) which is a standard evaluation measure in information retrieval. It is defined as the mean of the average precision scores for a set of queries, where the average precision equals the area under the precision-recall curve. As we put our work in the context of large scale image search, in the experiments, we report the mAP in function of the *Query Selectivity* S_{DB}, i.e. the average fraction of database accessed per query, on both the considered datasets (Figure 1). Each line is obtained varying the most effective parameter (γ for SQ, the number of accessed list *nprobe* for PQ). Well-trained PQ-based index perform best, but SQ-based methods provide a slightly degraded off-the-shelf performance without the need of any initial training set, training phase, or specialized index structure that instead highly influences IVFPQ. Moreover, the CReLU transformation consistently boost performance over plain SQ in all regimes.

5 Conclusions

This paper presented a simple and effective methodology to index and retrieve deep features without the need for a time-consuming codebook learning step. Our approach relies on transforming the deep features into text encodings, which can be subsequently indexed and searched using off-the-shelf text search engines. An important aspect is that our encoding technique is completely independent from the technology used for indexing. We can rely on the latest text search engine technologies, without having to worry about issues related to implementation problems, such as software maintenance, updates to new hardware technologies, bugs, etc. Furthermore, with our approach, it is possible to include in the image records, in addition to the visual features (which are in textual form), other information such as text metadata, geotags, etc.

Acknowledgements The work was partially supported by Smart News (CUP CIPE D58C15000270008), VISECH, ARCO-CNR (CUP B56J17001330004), ADA (CUP CIPE D55F17000290009), and the AI4EU project (funded by the EC, H2020 - Contract n. 825619). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

1. Amato, G., Bolettieri, P., Falchi, F., Gennaro, C., Rabitti, F.: Combining local and global visual feature similarity using a text search engine. In: Proceedings of the 2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI). pp. 49 –54 (june 2011)
2. Amato, G., Carrara, F., Falchi, F., Gennaro, C., Vadicamo, L.: Large-scale instance-level image retrieval. Information Processing & Management p. 102100 (2019)
3. Amato, G., Falchi, F., Gennaro, C., Vadicamo, L.: Deep Permutations: Deep convolutional neural networks and permutation-based indexing. In: Proceedings of the 9th International Conference on Similarity Search and Applications. pp. 93–106. SISAP 2016, LNCS, Springer International Publishing (2016)

4. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5297–5307. CVPR 2016, IEEE (June 2016)
5. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Proceedings of 13th European Conference on Computer Vision. pp. 584–599. ECCV 2014, Springer (2014)
6. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: A deep convolutional activation feature for generic visual recognition. CoRR **abs/1310.1531** (2013)
7. Gennaro, C., Amato, G., Bolettieri, P., Savino, P.: An approach to content-based image retrieval based on the lucene search engine library. In: Proceedings of the International Conference on Theory and Practice of Digital Libraries. pp. 55–66. TPDL 2010, Springer Berlin Heidelberg (2010)
8. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587. CVPR 2014, IEEE (June 2014)
9. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: End-to-end learning of deep visual representations for image retrieval. International Journal of Computer Vision **124**(2), 237–254 (Sep 2017)
10. Jégou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Proceedings of the European Conference on Computer Vision, ECCV 2008, LNCS, vol. 5302, pp. 304–317. Springer (2008)
11. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(1), 117–128 (Jan 2011)
12. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: IEEE Conference on Computer Vision & Pattern Recognition. pp. 3304–3311 (jun 2010)
13. Liu, R., Wei, S., Zhao, Y., Yang, Y.: Indexing of the CNN features for the large scale image search. Multimedia Tools and Applications **77**(24), 32107–32131 (Dec 2018)
14. Mohedano, E., McGuinness, K., O'Connor, N.E., Salvador, A., Marques, F., Giro-i Nieto, X.: Bags of local convolutional features for scalable instance search. In: Proceedings of the ACM International Conference on Multimedia Retrieval. pp. 327–331. ICMR 2016, ACM (2016)
15. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. CVPR 2007, IEEE (2007)
16. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA (1986)
17. Shang, W., Sohn, K., Almeida, D., Lee, H.: Understanding and improving convolutional neural networks via concatenated rectified linear units. In: Proceedings of the 33rd International Conference on Machine Learning. ICML 2016, vol. 48, pp. 2217–2225. JMLR.org (2016)
18. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. CoRR **abs/1511.05879** (2015)
19. 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. pp. 53–61. CVPRW 2015, IEEE (2015)