Surrogate Text Representation of Visual Features for Fast Image Retrieval

Fabio Carrara[0000−0001−5014−5089]
Supervised by Dr. Giuseppe Amato, Dr. Claudio Gennaro, and Prof. Francesco Marcelloni.

Institute of Information Science and Technologies (ISTI), Italian National Research Council (CNR), Via G. Moruzzi 1, 56124 Pisa, Italy
fabio.carrara@isti.cnr.it

Abstract. We propose a simple and effective methodology to index and retrieve image features without the need for a time-consuming codebook learning step. We employ a scalar quantization approach combined with Surrogate Text Representation (STR) to perform large-scale image retrieval relying on the latest text search engine technologies. Experiments on large-scale image retrieval benchmarks show that we improve the effectiveness-efficiency trade-off of current STR approaches while performing comparably to state-of-the-art main-memory methods without requiring a codebook learning procedure.

Keywords: image retrieval · deep features · surrogate text representation · inverted index

1 Intruduction

As part of the contributions of our thesis, we explore new approaches for making image retrieval as similar as possible to text 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 representation extracted from a CNN, often referred to as *deep features*, and to transform them into a surrogate text that standard text search engine can index. Our general approach is based on the transformation of deep features, which are (dense) vectors of real numbers, into sparse vectors of integer numbers that can be mapped in term frequencies. Moreover, sparseness is necessary to achieve sufficient levels of efficiency as it does for search engines for text documents. We introduce and evaluate a novel approach for surrogate text representation of deep features based on Scalar Quantization (SQ).

2 Surrogate Text Representation

We define a family of transformations that map a feature vector into a textual representation without the need for tedious training procedures. We require that such transformations preserve as much as possible the proximity relations between the data, i.e., similar feature vectors are mapped to similar textual documents. This basic idea was firstly exploited in [2], where the authors defined the Surrogate Text Representation (STR) to represent a generic metric object, i.e., an object living in a space where a distance function is defined [7]. The STR of an object is a space-separated concatenation of some alphanumeric codeword selected from a pre-defined dictionary. We observe that a surrogate text representation for an object o of a data domain $\mathcal{X}$ can be obtained more generally by defining a transformation

$$\begin{aligned} f : \mathcal{X} &\to \mathbb{N}^n \\ o &\mapsto \boldsymbol{f}_o = [f_o^{(1)}, \ldots, f_o^{(n)}] \,, \end{aligned} \tag{1}$$

where $\boldsymbol{f}_o$ will act as the vector of the term frequencies of a synthetic text document t_o with respect to a dictionary of n words. Given a dictionary $\{\tau_1, \ldots, \tau_n\}$ and the transformation $f : \mathbb{R}^m \to \mathbb{N}^n$, we define the surrogate text $t_o = \mathrm{STR}_f(o)$ as

$$\mathrm{STR}_f(o) = \bigcup_{i=1}^{n} \bigcup_{j=1}^{f_o^{(i)}} \tau_i \tag{2}$$

where, by abuse of notation, we denote the space-separated concatenation of the codewords with the union operator $\cup$. Thus, by construction, the integer values of the i-th component of the vector $\boldsymbol{f}_o$ is the frequency of the codeword τ_i in the text $\mathrm{STR}_f(o)$. For example, given $\boldsymbol{f}_o = [1, 3, 0, 2]$ and a codebook $\{\tau_1 =$ "A'', $\tau_2 =$ "B'', $\tau_3 =$ "C'', $\tau_4 =$ "$D''\}$, we have $\mathrm{STR}_f(o) =$ "$A\,B\,B\,B\,D\,D''$. 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 $\mathrm{STR}_f(o)$. We demonstrate indexing and searching *D-dimensional vectors* compared with the dot product, i.e. $\mathcal{X} = \mathbb{R}^D$.

3 Scalar Quantization-Based Approach

The first step in our approach is the application of an random orthogonal transformation to the vectors $\mathbf{v} \to R(\mathbf{v} - \boldsymbol{\mu}), \mathbf{q} \to R\mathbf{q}$, where $\mathbf{v}$ and $\mathbf{q}$ are vectors of database and query images, R is a *random* orthogonal matrix ($\|R\|_2 = 1$), and $\boldsymbol{\mu} \in \mathbb{R}^D$ can be arbitrary chosen. The benefit of this transformation is manyfold: a) it re-balances the variance of the components of the feature vectors [4], thus preventing unbalanced posting lists in the inverted file; b) it is ordering-preserving with respect to the dot-product: given a rotation matrix $R \in \mathbb{R}^{D \times D}$ and a vector $\boldsymbol{\mu} \in \mathbb{R}^D$, then

$$\forall\, \mathbf{q}, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^D \quad \mathbf{q} \cdot \mathbf{v}_1 \leq \mathbf{q} \cdot \mathbf{v}_2 \Rightarrow R\mathbf{q} \cdot R(\mathbf{v}_1 - \boldsymbol{\mu}) \leq R\mathbf{q} \cdot R(\mathbf{v}_2 - \boldsymbol{\mu}) \,. \tag{3}$$

Then, we transform the rotated vectors into integer term-frequency vectors by scaling and quantization: $\mathbf{v} \to \lfloor s\mathbf{v} \rfloor$ where $\lfloor \cdot \rfloor$ denotes the floor function, and $s > 1$ is the *quantization factor*. This process introduces a quantization error due to the representation of float components in integers that does not affect the retrieval effectiveness. In summary, the SQ-based STR is obtained using the transformation $f_{\text{SQ}} : \mathbf{v} \mapsto \lfloor sR(\mathbf{v} - \boldsymbol{\mu}) \rfloor$. For instance, suppose after the random rotation we have the feature vector $\mathbf{v} = [0.1, 0.3, 0.4, 0, 0.2]$, by adopting a multiplication factor $s = 10$, we obtain the term frequencies vector will be $f_{\text{SQ}}(\mathbf{v}) = [1, 3, 4, 0, 2]$, and thus $\text{STR}_{f_{\text{SQ}}}(\mathbf{v}) =$ "$A\,B\,B\,B\,C\,C\,C\,C\,E\,E$".

In order to sparsify the term frequency vectors, that is, to cancel the less significant components of the vectors, we adopt thresholding

$$\mathbf{v}_\gamma(i) = \begin{cases} \mathbf{v}(i) & \text{if} \quad \mathbf{v}(i) \geq \frac{1}{\gamma} \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where $\mathbf{v}(i)$ indicates the i-th dimension of $\mathbf{v}$. To optimize this step, we set $\boldsymbol{\mu}$ of Eq. (3) to the mean vector to obtain zero-centered components. Thresholding alone ignores the negative components of the original real-valued feature vectors, discarding useful information. In order to prevent this, before our proposed transformation, we also apply the Concatenated ReLU (CReLU) transformation to feature vectors, defined as $\mathbf{v}^+ = \max([\mathbf{v}, -\mathbf{v}], \mathbf{0})$, where $\max(\cdot, 0)$ is applied element-wise. Notice that in general, the CReLU operation is lossy since the dot product between vectors is not preserved, i.e., $\mathbf{v}_1 \cdot \mathbf{v}_2 \leq \mathbf{v}_1^+ \cdot \mathbf{v}_2^+$. However, this transformation allows us not to completely neglect the negative components.

4 Experimental Evaluation and Discussion

To validate our approach, we extract R-MAC features [6] — a 2048-dimensional real-valued feature vector — from the images of INRIA Holidays + MIR-Flickr1M [3] — a standard benchmark for image retrieval. We evaluate our approach for different values of the thersholding parameter γ. We compare with Deep Permutations (DP) [1], a permutation-based STR approach. We generate different sets of permutations from the original and CReLU-ed features by considering the top-k truncated sorting permutations at different values of k. We also compared with state-of-the-art main-memory approximate nearest neighbor algorithms based on Product Quantization FAISS [5]. For a fair comparison, we use a configuration for FAISS which gives the best trade-off between effectiveness and efficiency; we choose a relatively big code size ($C = 1,024$) and optimal number of Voronoi cells ($N = 16$k), resulting in 1GB in main memory against about 0.7GB of our solution in secondary memory in the larger configuration. Since FAISS methods need to learn a codebook from data, we report results with two different training datasets: the indexed dataset itself, on which the mAP evaluation is performed, and T4SA, an uncorrelated dataset of Twitter images. Results in Figure 1 show a satisfactory trade-off trend between effectiveness and query selectivity and the general benefit introduced by the CReLU preprocessing. We notice that the impact of codebook learning on PQ-based methods is

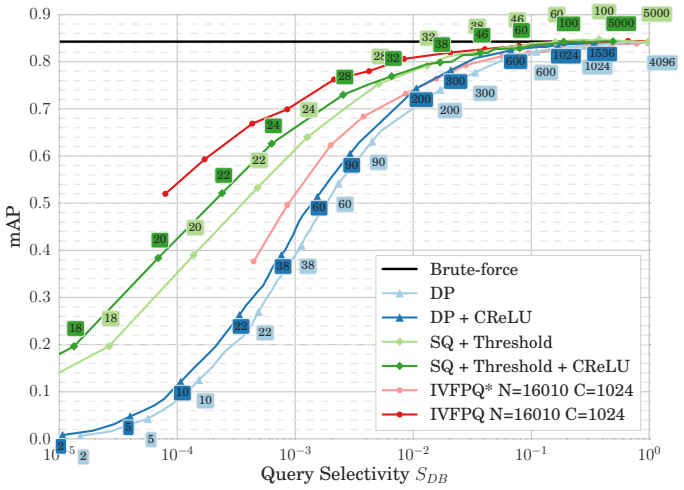

Fig. 1. Performance on Holidays + MIRFlickr1M dataset. The curves are obtained varying k (for deep permutations), γ (for thresholded SQ), and the number of Voronoi cell accessed P (for IVFPQ). Values of k and γ are reported near each point. The horizontal line represents the mAP obtained using the original R-MAC vectors and performing a sequential scan of all the dataset (brute-force approach).

really strong, and it could, in real applications, influence the scalability of the system or require continuous codebook adjustments, forcing to re-indexing the data periodically. Our solution has an intermediate performance but does not require any training procedure and therefore any re-adjustments.

References

1. Amato, G., Bolettieri, P., Carrara, F., Falchi, F., Gennaro, C.: Large-scale image retrieval with elasticsearch. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 925–928. ACM (2018)
2. Gennaro, C., Amato, G., Bolettieri, P., Savino, P.: An approach to content-based image retrieval based on the lucene search engine library. In: International Conference on Theory and Practice of Digital Libraries. pp. 55–66. Springer (2010)
3. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: European conference on computer vision. pp. 304–317. Springer (2008)
4. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3304–3311. IEEE (2010)
5. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 (2017)
6. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of cnn activations (2016)
7. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity search: the metric space approach, vol. 32. Springer Science & Business Media (2006)