Preserving local spatial information in image similarity using tensor aggregation of local features

David Picard

PRESERVING LOCAL SPATIAL INFORMATION IN IMAGE SIMILARITY USING TENSOR AGGREGATION OF LOCAL FEATURES

David Picard

picard@ensea.fr
ETIS - ENSEA/Université de Cergy-Pontoise/CNRS UMR 8051, F-95000 Cergy-Pontoise, France

ABSTRACT

In this paper, we propose an aggregation scheme of local descriptors that preserves local spatial information. Our method is based on the binary product of similarities of nearby matching pairs of descriptors. The similarities are linearized using a tensor framework. We show our approach can be used with any local descriptors, handcrafted like SIFT, or learned like the outputs of convolutional layers in deep neural networks. We perform experiments on the Holidays dataset that show the soundness of the approach.

***Index Terms*—** Image retrieval, Image databases, Image representation

1. INTRODUCTION

Content based image similarity has been the foundation of many innovative applications in the last decade, from copy detection to automatic labeling and object detection. Since these applications cover a wide variety of topics, methods have mostly been tailored to solve specific problems. We can order these problems and the methods that attempt to solve them along a precision/generalization axis. At the one end of this axis, problems like copy detection attempt to match an image with geometrical or colorimetric transforms of itself (like rotation, scale, crop, *etc*). The methods developed to solve this set of problems rely on very precise features that ought to be invariant to these transforms. On the other end of the axis are problems like object detection, where a bounding box has to be drawn around instances of a specific class, *e.g.*, *cat*. Due to the large variability inside the targeted class, these methods rely on features (often learned) that can generalize to most samples in the class. The common point in these methods is that they all define a visual similarity between images.

In this paper, we are interested in the encoding of local spatial information in such visual similarities. Local spatial information considers the layout of salient features in a small neighborhood of a specific region, contrarily to global spatial information which considers the layout of the entire image. While global spatial information is mainly taken into account in the existing methods, local spatial information is often missing. To add the spatial relationship between local features, we draw from the keypoint matching methods and use a tensor framework to allow the embedding of a spatially sensitive matching scheme into an image signature.

Our contributions are the following: We show that spatially sensitive pairwise matching of local features can be rewritten as the dot product of their tensors. Using this tensor framework, we propose a aggregating scheme of local features that embed pairwise matching. We use this new aggregation scheme with both handcrafted features (SIFT [1]) and learned features obtained from deep convolutional neural networks (CNN [2]).

The paper is organized as follows: In the next section, we recall popular image signatures and how they incorporate spatial information. In Section 3, we detail our framework and discuss its properties. In section 4 we show experiments on the Holidays dataset that validate our approach, before we conclude in Section 5.

2. RELATED WORK

In copy detection or in near duplicate search, the best performing methods are based on the matching of highly discriminative local features such has SIFT [1]. The matches are then refined using a geometric consistency check that keeps only the matches forming a consensus with respect to the geometric transform between the two images [3]. The assumption behind this check is that the spatial structure of the object of interest does not change by the transform. In that sense, matching methods using geometric consistency filtering embed local spatial information. However, these methods are computationally costly since the matching cost increases quadratically with the number of local features, and the geometric filtering is often also very dependent on the number of matches. Even with compression techniques and approximate searches [4], these methods needs to store all the compressed features and thus scale badly.

To overcome these problems, aggregation methods have been proposed, following the Bag of Word model [5]. The idea is to aggregate all local features into a single vector, called signature, such that a similarity measure on the signatures approximates the matching schemes. These methods rely on a clustering of the descriptors space called *Visual Codebook* and usually obtained by k-means or GMM. Considering that descriptors should be matched only if they fall inside the same cluster, the framework proposed in [6] allows to describe the process using Taylor expansion of the Gaussian matching kernel. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be 2 local descriptors and $\delta_{ij} = 1$ if both descriptors are assigned to the same codebook entry, and 0 otherwise. The following Gaussian matching kernel can then be approximate using its n-th order Taylor expansion:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij} e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|} \approx \delta_{ij} \sum_k^n \alpha_k \|\mathbf{x}_i - \mathbf{x}_j\|^k \quad (1)$$

When descriptors are normalized, the expansion can bee linearized:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij} \sum_k^n \beta_k \langle \mathbf{x}_i, \mathbf{x}_j \rangle^k = \delta_{ij} \sum_k^n \beta_k \langle \mathbf{x}_i^{\otimes k}, \mathbf{x}_j^{\otimes k} \rangle \quad (2)$$

With $\mathbf{x}^{\otimes k}$ being the n-th order tensor of $\mathbf{x}$. Using the linearity property, the sum of all possible matching similarities can then be simplified by performing the aggregation of encoded features as a preprocessing. Setting different values for n allows to consider different aggregation schemes, namely BoW for $n = 0$ [5], VLAD for

$n = 1$ [7] and VLAT for $n = 2$ [6]. Although they don't fit the framework, it is worth mentioning that popular Fisher Vectors [8] are closely related to VLAT since they also consider the second order moments of the descriptors distribution. Moreover, they seem to perform identically [9].

One of the drawbacks of such methods is that all spatial information is lost, since the descriptors are considered as orderless bags of vectors. To bring back some spatial information, several methods have been proposed. The most popular is the Spatial Pyramid Matching [10] (SPM) in which descriptors are aggregated in several fixed regions of the image following a recursive grid partitioning. In [11], the authors consider to take into account the location of the descriptors when computing the codebook and encoding the descriptors. Both methods encode the layout information of the image (*i.e.*, which pattern is located at which absolute place), and not the relative spatial information (*i.e.*, which pattern is near which other pattern).

In contrast to aggregation schemes, deep convolutional neural networks have been successfully used to perform image classification [2]. A convolutional layer consists in computing the activation of neurons on a sliding window, producing a activation map akin to the convolution of the image with a (non-linear) filter bank. Another interpretation is to consider that the map corresponds to the localized detection scores of the pattern encoded in the neuron weights. The activation function of the neurons induces a non-linearity that helps reducing the noise in the activation by zeroing low responses. A typical non-linear activation is the rectified linear unit (ReLU), which is basically a simple soft-thresholding strategy [12]. In deep networks, several of such convolutional layers are stacked (from 5 in [2] to over 150 in [13]). Then, fully connected layers are added to aggregate the local responses in a global description. The last layer of the fully connected stack is composed of as many neurons as there are classification classes and produces the class prediction outputs.

The weight of deep neural networks are usually learn by performing the back-propagation of the classification error, but unsupervised criteria based on the reconstruction error provide a good initialization [14]. Wavelet based weights have also been explored with success [15].

It should be noted that in CNN, the convolutional layers encode local spatial information, whereas the fully connected layers encode global layout information. Indeed, if the neurons at convolutional layer n correspond to specific patterns, then the neurons at convolutional layer $n + 1$ are the combination of said patterns with specific relative position within a small window. On the contrary, entries of the first FC layer are combination of these patterns located at specific locations in the whole image. With respect to spatial information, FC layers can thus be compared to the popular SPM of aggregation scheme.

Finally, deformable part models [16] encode the relative location of specific pattern with respect to one another. While these methods achieve high performances in object detection, they need to be trained for each object class. As such, they cannot be used to compute the similarity between images for which no prior knowledge on the contained objects is available.

3. PROPOSED METHOD

Since local spatial information is often missing in existing methods, we propose to focus on it. The main idea of our method is as follows: We consider the matching of a descriptor of the query image with its corresponding descriptor in the target image. Under fair low-deformation assumption on the content, and if a second descriptor in the vicinity of the first one has a correspondence in the target image, then it should also be in the vicinity of the match in the target image. As such, a way to encode this local spatial information is to consider a binary product (AND operator) between the first descriptor matching function and its neighbor matching function. We show that this binary product of pairwise matching can be efficiently linearized using tensors.

Given $B_i = \{\mathbf{x}_{ri}\}$ the set of local descriptors extracted from image i, let $\Omega(\mathbf{x}_{ri}) \subset B_i$ be be the set of descriptors of B_i in the vicinity of $\mathbf{x}_{ri}$ defined by a spatial support Ω. If $k(\cdot, \cdot)$ is a function that measures the similarity between any 2 descriptors, then counting the binary product of pairwise matching in Ω of 2 descriptors $\mathbf{x}_{ri} \in B_i$ and $\mathbf{x}_{sj} \in B_j$ is simply:

$$k_\Omega(\mathbf{x}_{ri}, \mathbf{x}_{sj}) = \sum_{\substack{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri}) \\ \mathbf{x}_v \in \Omega(\mathbf{x}_{sj})}} k(\mathbf{x}_{ri}, \mathbf{x}_{sj}) k(\mathbf{x}_u, \mathbf{x}_v) \tag{3}$$

In the following, we will consider that k is simply the dot product, but any non-linear similarity function that can be approximate by a Taylor expansion as in [17] will work in our framework. Considering the dot product, k can be simplified to:

$$k_\Omega(\mathbf{x}_{ri}, \mathbf{x}_{sj}) = \sum_{\substack{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri}) \\ \mathbf{x}_v \in \Omega(\mathbf{x}_{sj})}} \langle \mathbf{x}_{ri}, \mathbf{x}_{sj} \rangle \langle \mathbf{x}_u, \mathbf{x}_v \rangle \tag{4}$$

$$= \sum_{\substack{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri}) \\ \mathbf{x}_v \in \Omega(\mathbf{x}_{sj})}} \langle \mathbf{x}_{ri} \otimes \mathbf{x}_u, \mathbf{x}_{sj} \otimes \mathbf{x}_v \rangle \tag{5}$$

$$= \left\langle \sum_{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri})} \mathbf{x}_{ri} \otimes \mathbf{x}_u, \sum_{\mathbf{x}_v \in \Omega(\mathbf{x}_{sj})} \mathbf{x}_{sj} \otimes \mathbf{x}_v \right\rangle \tag{6}$$

The similarity between image i and j is then simply the sum of such matching over all descriptors of both images:

$$K(B_i, B_j) = \sum_{\substack{\mathbf{x}_{ri} \in B_i \\ \mathbf{x}_{sj} \in B_j}} k_\Omega(\mathbf{x}_{ri}, \mathbf{x}_{sj}) \tag{7}$$

$$= \left\langle \sum_{\substack{\mathbf{x}_{ri} \in Bi \\ \mathbf{x}_u \in \Omega(\mathbf{x}_{ri})}} \mathbf{x}_{ri} \otimes \mathbf{x}_u, \sum_{\substack{\mathbf{x}_{sj} \in B_j \\ \mathbf{x}_v \in \Omega(\mathbf{x}_{sj})}} \mathbf{x}_{sj} \otimes \mathbf{x}_v \right\rangle \tag{8}$$

The tensor products of descriptors in Ω can be computed ahead of time and provide an encoding of the descriptors that produce a single signature comparable to other aggregation scheme, only that it contains local spatial information.

However, in such scheme, many false matches are considered. In particular, when using the dot product, small non-zero similarities are added for all descriptors in Ω. Since the number of such matches increases quadratically with the size of the support, a huge amount of noise is added. To circumvent this issue, we propose to use the same codebook based strategy as in VLAD. We thus propose to match only descriptors that correspond to the same entry of the codebook (*i.e.*, that have the same nearest neighbor among the entries of the codebook). Let D be a codebook of m entries and $h(\mathbf{x}_{ri})$ a vector with 1 on the component corresponding to the entry of D associated with $\mathbf{x}_{ri}$ and 0 on all other components, the restricted matching similarity between 2 descriptors is then:

$$k_r(\mathbf{x}_{ri}, \mathbf{x}_{sj}) = \langle h(\mathbf{x}_{ri}), h(\mathbf{x}_{sj}) \rangle k(\mathbf{x}_{ri}, \mathbf{x}_{sj}) \tag{9}$$

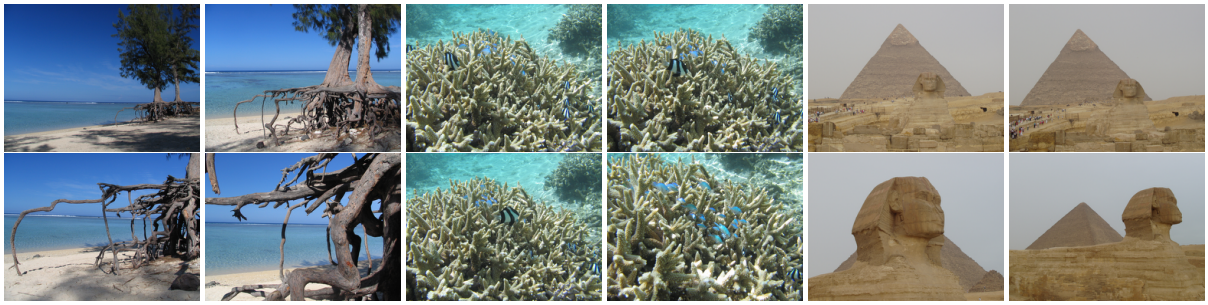

Fig. 1. Images from Holidays dataset.

Applying this to k_Ω leads to the following similarity:

$$k_{r\Omega}(\mathbf{x}_{ri}, \mathbf{x}_{sj}) = \sum_{\substack{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri}) \\ \mathbf{x}_v \in \Omega(\mathbf{x}_{sj})}} \Big(\langle h(\mathbf{x}_{ri}), h(\mathbf{x}_{sj}) \rangle k(\mathbf{x}_{ri}, \mathbf{x}_{sj}) \times \langle h(\mathbf{x}_u), h(\mathbf{x}_v) \rangle k(\mathbf{x}_u, \mathbf{x}_v) \Big) \quad (10)$$

When using the dot product, this can easily be linearized using tensors:

$$\begin{aligned} k_{r\Omega}(\mathbf{x}_{ri}, \mathbf{x}_{sj}) &= \sum_{\substack{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri}) \\ \mathbf{x}_v \in \Omega(\mathbf{x}_{sj})}} \Big(\langle h(\mathbf{x}_{ri}), h(\mathbf{x}_{sj}) \rangle \langle \mathbf{x}_{ri}, \mathbf{x}_{sj} \rangle \times \langle h(\mathbf{x}_u), h(\mathbf{x}_v) \rangle \langle \mathbf{x}_u, \mathbf{x}_v \rangle \Big) \\ &= \Big\langle \sum_{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri})} h(\mathbf{x}_{ri}) \otimes h(\mathbf{x}_u) \otimes \mathbf{x}_{ri} \otimes \mathbf{x}_u, \\ &\quad \sum_{\mathbf{x}_v \in \Omega(\mathbf{x}_{sj})} h(\mathbf{x}_{sj}) \otimes h(\mathbf{x}_v) \otimes \mathbf{x}_{sj} \otimes \mathbf{x}_v \Big\rangle \end{aligned} \quad (11)$$

Which is the dot product between 4th order tensors. The first 2 blocks of dimensions correspond to the entries of the codebook, while the last 2 blocks of dimension correspond to the second order raw moments between descriptors belonging to the corresponding entries.

The similarity between 2 images is then simply the sum of similarities over all descriptors of both the query and the target:

$$K_r(B_i, B_j) = \sum_{\substack{\mathbf{x}_{ri} \in B_i \\ \mathbf{x}_{sj} \in B_j}} \Big\langle \sum_{\mathbf{x}_u \in \Omega(\mathbf{x}_{ri})} h(\mathbf{x}_{ri}) \otimes h(\mathbf{x}_u) \otimes \mathbf{x}_{ri} \otimes \mathbf{x}_u, \sum_{\mathbf{x}_v \in \Omega(\mathbf{x}_{sj})} h(\mathbf{x}_{sj}) \otimes h(\mathbf{x}_v) \otimes \mathbf{x}_{sj} \otimes \mathbf{x}_v \Big\rangle \quad (12)$$

Using the linearity, K_r can be obtained by computing the following 4th order tensor:

$$T(B_i) = \sum_{\substack{\mathbf{x}_{ri} \in B_i \\ \mathbf{x}_u \in \Omega(\mathbf{x}_{ri})}} h(\mathbf{x}_{ri}) \otimes h(\mathbf{x}_u) \otimes \mathbf{x}_{ri} \otimes \mathbf{x}_u \quad (13)$$

We name this tensor the *Spatial Tensor Aggregation* (STA). This tensor is then flattened into a vector and the similarity between images is computed using the dot product:

$$K_r(B_i, B_j) = \langle vec(T(B_i)), vec(T(B_j)) \rangle \quad (14)$$

An efficient way of computing $T(B_i))$ is to loop over all pairwise combination (c, d) of entries in D, and then compute the second order raw moment matrix of descriptors associated with c and their neighbors associated with d. This leads to $m \times m$ second order raw moment matrices that are flattened and concatenated.

Three remarks are worth mentioning here. First, STA can be computed using any local descriptor. In the experiments section, we show we obtained good performances both using well known SIFT descriptors and the output of the convolutional layers of a deep CNN. Second, many normalization tricks that are widely used with other aggregation schemes can be applied here. In fact, we obtained the best results using centering and intra-normalization on the descriptors and power normalization on the tensors as in [18]. Third the geometry of the spatial support Ω allows to take into account spatial transform invariance. For instance, using a line for Ω allows to consider scale invariance, since a pair of descriptors of the query can be matched with a pair of descriptors in the target with a closer or larger distance but in the same direction. Similarly, using a circle for Ω leads to rotational invariance since the distance between matching pairs is preserved but not the angle.

4. EXPERIMENTS

In this section, we present experiments on the Holidays dataset [3], which consists of 1491 images divided into 500 groups of the same scene. Images from the dataset are shown in Figure 1. We mainly compare our method to VLAT [17] since it is the most closely related method in the state of the art. We use 2 types of local descriptors, namely dense SIFT extracted using the vlfeat softare [19], and the output of the convolutional layers of the AlexNet deep CNN [2] obtained using the matconvnet software. To have a fair comparison between SIFT and CNN based local features, we rescaled the image to have a fixed height of 256 pixels and took the union of descriptor sets extracted on several regularly spaced crops of 227x227 pixels. This setup is based on the constraints of the CNN, and we acknowledge that better results could have been obtained for SIFT using larger images.

All descriptors are first projected to a fixed 56 dimensions space using PCA and then normalized. In all our tests, 56 dimensions is enough to retain more than 90% of the variance for both the SIFTs

	SIFT	cnn-1	cnn-2	cnn-3	cnn-4	cnn-5
VLAT	67.4	56.8	64.6	70.1	70.5	64.7
STA	69.6	58.6	67.6	72.0	72.3	66.8
Ω	8	1	1	2	3	2

Table 1. Comparison in mean Average Performance (mAP) on Holidays between VLAT and Spatial Tensor Aggregation using various descriptors and the same codebook. Ω denotes the size of the spatial window leading to these results.

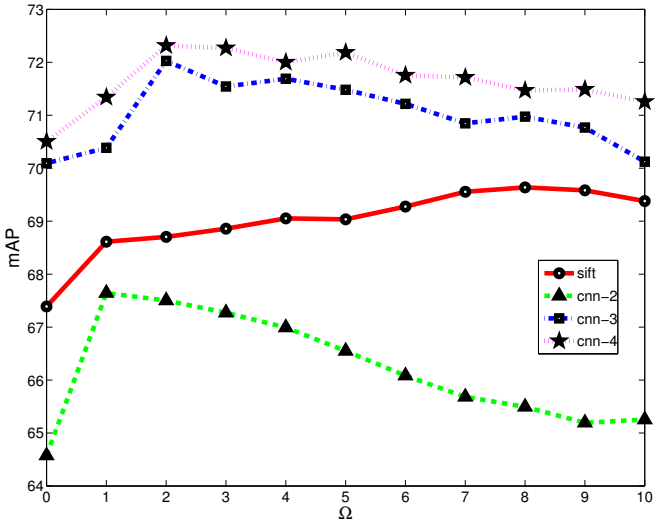

Fig. 2. mAP against the length of the support Ω. 0 ist equivalent to VLAT.

and the convolutional layers of the CNN. After reduction, all descriptors are normalized to unit ℓ_2-norm. We then used a random subset of 150k descriptors to train the codebook using k-means. The number of entries in the codebook was fixed to 8. In all of our tests, Ω is a descending vertical line starting from $\mathbf{x}_{ri}$ with a length measured in number of adjacent descriptors.

In Table 1, we show the comparison of the VLAT methods compared to our best results using spatial tensor aggregation both on SIFT and using various convolution layers. As we can see, incorporating spatial information allows to gain between 1% and 3%, which is significant on this dataset. We also show the support size that led to the results. It should be remarked that this size is relative to the spatial coverage of the descriptors. As such, even though the best support size remains fairly constant for all layers of CNN, the corresponding spatial coverage increases since deeper layers encode larger equivalent filters.

We show in Figure 2 the variation in mAP against the size of the spatial support. A support of 0 is equivalent to VLAT, and only the diagonal components of the tensor corresponding to the same entries in the codebook are non-zero. As we can see, increasing the size of the spatial support steadily increases the mAP up to an optimal size after which the spatial aggregation begins to add noise. For reasonable relative support sizes, the performances are improved without much variation, which shows the robustness of the method to the parameters.

Finally, we show in Table 2 a comparison with other existing methods. We first report result using the fully connected layers of AlexNet using matconvnet. As already shown in many studies, fully connected layers provide a strong baseline in many image retrieval applications. It should be remarked that FC layers contains spatial

Method	params	mAP
AlexNet FC	dim	-
layer-6 conv	4k	66.3%
layer-6 relu	4k	61.0%
layer-7 conv	4k	60.0%
layer-7 relu	4k	61.8%
VLAD [21]	rootsift, m=256	65.3%
Fisher Vectors [7]	sift, m=256	62.5%
VLAT		
[22]	sift, m=64	70.0%
(this paper)	sift, m=8	67.4%
(this paper)	cnn-4, m=8	70.5%
FAemb [20]	rootsift, m=8	72.7%
STA	desc	-
	sift, m=8, $\Omega = 8$	69.6%
	rootsift, m=8, $\Omega = 8$	72.1%
	cnn-4, m=8, $\Omega = 3$	72.3%

Table 2. Comparison with existing methods. Cited results are reported from their corresponding paper, while others are computed using the same code base as in our method.

information comparable to that of SPM that is outperformed by our local spatial aggregation with a massive gain of 6%. One should also note that the best performances are obtained by the first FC layer without the rectification. Indeed, the rectification decreases the performances by a colossal 5%, which hints that the FC layers are too much tailored toward the classification objectives of the full network.

We report the same results as in [20] which is to our knowledge the latest aggregation method proposed in the literature. The improvements obtained by our approach over VLAD, VLAT and Fisher Vectors is consequent, especially when compared to the results obtained by our implementation of these methods. The relative gain over VLAT is comparable to that obtained by FAemb [20], although FAemb performs better then STA with comparable descriptors. However, it should be noted that the computational cost of FAemb greatly exceeds that of STA since an optimization problem has to be solved for each descriptor in the aggregation. In our case, although the number of aggregations is much higher (depending on the support size), it still requires only a constant number of vector operations for each descriptor. Moreover, since FAemb proposes an encoding scheme that linearizes a matching kernel, it can be included in our framework in replacement of the dot product.

5. CONCLUSION

In this paper, we considered the local spatial information in image similarity. We start from the binary product of similarities in nearby matching pairs of local descriptors, and show it can be linearized using tensors. The obtained aggregation scheme can be used with any local descriptors. Depending on the spatial support where the aggregation is performed, invariance to geometric transforms can be obtained. We perform experiments on the Holidays dataset using SIFT descriptors and the outputs of the convolutional layers of AlexNet deep CNN. We show our method is able to obtain comparable results with the state of the art, while being much simpler to implement and requiring less fine tuning of the parameters.

Further work include using non linear matching kernels such as the ones provides by FAemb encoding, and evaluating the impact on the results of the invariance obtained by different support geometry.

6. REFERENCES

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 2, no. 60, pp. 91–110, 2004.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European Conference on Computer Vision*. Springer, 2008, pp. 304–317.

[4] Hervé Jégou, Matthijs Douze, and Cordelia Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, jan 2011.

[5] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, 2003, vol. 2, pp. 1470–1477.

[6] David Picard and Philippe-Henri Gosselin, "Efficient image signatures and similarities using tensor products of local descriptors," *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 680–687, 2013.

[7] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 3304–3311, 2012, QUAERO.

[8] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2010.

[9] Romain Negrel, David Picard, and Philippe-Henri Gosselin, "Dimensionality reduction of visual features using sparse projectors for content-based image retrieval," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2192–2196.

[10] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE International Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2006, pp. 2169–2178, IEEE Computer Society.

[11] Piotr Koniusz and Krystian Mikolajczyk, "Spatial coordinate coding to reduce histogram representations, dominant angle and colour pyramid match," in *IEEE International Conference on Image Processing*, 2011.

[12] George E Dahl, Tara N Sainath, and Geoffrey E Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8609–8613.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[14] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio, "Why does unsupervised pre-training help deep learning?," *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[15] Joan Bruna and Stéphane Mallat, "Invariant scattering convolution networks," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1872–1886, 2013.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[17] David Picard and Philippe-Henri Gosselin, "Improving image similarity with vectors of locally aggregated tensors," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. pages–669.

[18] Philippe-Henri Gosselin, Naila Murray, Hervé Jégou, and Florent Perronnin, "Revisiting the fisher vector for fine-grained classification," *Pattern Recognition Letters*, vol. 49, pp. 92–98, 2014.

[19] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," `http://www.vlfeat.org/`, 2008.

[20] Thanh-Toan Do, Quang D Tran, and Ngai-Man Cheung, "Faemb: a function approximation-based embedding method for image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3556–3564.

[21] Relja Arandjelovic and Andrew Zisserman, "All about vlad," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1578–1585.

[22] Romain Negrel, David Picard, and Philippe-Henri Gosselin, "Web-scale image retrieval using compact tensor aggregation of visual descriptors," *MultiMedia, IEEE*, vol. 20, no. 3, pp. 24–33, 2013.