ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Machine Learning and Content-Based Multimedia Retrieval

Philippe-Henri Gosselin[1] and David Picard[2]

1- INRIA Rennes Bretagne Atlantique
Rennes - France

2- ETIS / ENSEA - Université de Cergy-Pontoise - CNRS
Cergy-Pontoise - France

Abstract. This paper presents an overview of popular retrieval techniques based on machine learning for content based multimedia retrieval. Furthermore, we also propose to highlight current gaps and required improvement in this context. We first introduce common retrieval problems, and the usual models and assumptions made on multimedia data. Thanks to these assumptions, techniques based on machine learning can be used in many application cases. In this scope, we present popular methods for indexing multimedia data, like the ones based on the training of visual dictionaries. Then, we present supervised techniques that use labeled data to train and design retrieval components. We show how this last topic could benefit from many improvement from the machine learning community. Finally, this paper presents interesting perspective and new paradigms for multimedia retrieval based on machine learning.

1 Introduction

With the globalization of internet, collections with tremendous amounts of multimedia documents are available. For instance, more than 6 billions images were hosted on Flickr in 2011, and one hour of video is added to Youtube every second. In order to retrieve content in these large repositories, the need for learning-based system has become essential. Among these systems, the ones based on statistical and machine learning have known a great success thanks to their ability to effectively *and* efficiency index datasets.

In this paper, we propose an overview of current content-based retrieval systems based on machine learning techniques. We first present the common objectives of these systems, and the basic assumption on the visual descriptors processed by learning methods. Then, we show the popular indexing techniques for context-based retrieval based on the implicit or explicit matching of low-level visual descriptors. Finally, we present supervised techniques, or how to use labels and annotations to classify datasets and further improve index.

2 Retrieval Systems

In this section, we present the usual assumptions one can make about the common retrieval problems and solutions. Thanks to these assumptions, we will be able to work in the following sections with a generalist model, theoretically independent from specific applications.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

2.1 Retrieval objectives

In order to exploit multimedia datasets, different kind of searches can be imagined, depending on the target application and environmental concerns. Among theses, we can first cite the ones based on *similarity search*. For instance, *target search* aims at retrieving a specific document or object. Another example in this scope is *duplicate search* which aims at retrieving visual copies of a specific document, usually for copyright considerations. We can also cite *category search* as an extension of similarity search, where the aim is to retrieve a set of documents with common semantics, for instance the healthy organs in medical imaging. Let us note that these retrieval problems can be considered for complete documents, but also for part of a document. For instance, one can aim at retrieving the specific location of cars are inside pictures or video. This idea can also be considered for smallest part of documents, for instance in remote sensing images, a common problem is the classification of pixels. Furthermore, in the case where the time axis is available, tracking can be considered, as in action retrieval in video.

In all cases, a key component is the *index*, *e.g.* a metadata created from multimedia content that allows many kind of searches. There are many possibilities for indexing a multimedia dataset, but in most cases, we can assume that they are able to respond to the retrieval problems we presented. In order to create such data structures, we have to consider the content of multimedia documents, usually thanks to the extraction of low-level *descriptors* which encode specific properties of the underlying signal (for example, shape or color in images).

2.2 Low-level descriptors space and similarity measure

Most retrieval problems can be solved using low-level signal processing specific to the target application, and then using generalist tools based on machine learning techniques. Actually, in most cases we can assume that it exists a method that turns any low-level multimedia content into a model that better fits mathematical frameworks. In this scope, a popular scheme is to assume that low-level descriptors are extracted from the document as vectors. For instance, in the case of images, these descriptors can be the description of parts of the image, usually as Histogram of Gradient (HoG). Among all image descriptors, we can cite the very popular SIFT descriptors [1], that describe neighborhood of a pixels with a vector of 128 dimensions. We can find similar descriptions for video including time information [2], and descriptors for 3D objects [3].

Low-level descriptors must be defined with a comparison measure in order to be used. Actually, when one mentions a descriptor, it always implicitly includes the corresponding measure. Such measure can be a distance $d(\mathbf{b}_r, \mathbf{b}_s)$, in which case two descriptors $\mathbf{b}_r$ and $\mathbf{b}_s$ are close if distance is close to zero, or different as long as distance increases. The measure can also be a similarity $s(\mathbf{b}_r, b_s)$, in which case two descriptors are close if similarity is high, or different if similarity is close to zero. Since most popular descriptors are vectors, the corresponding measure is usually the Euclidean distance, or a dot product when the vectors

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

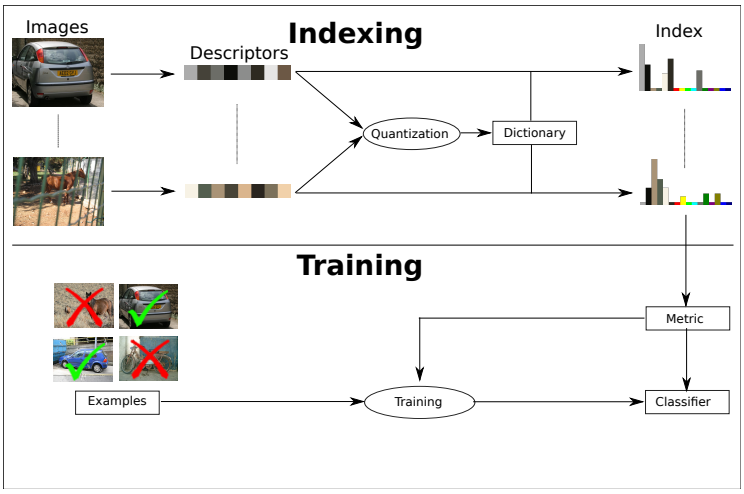

Fig. 1: Example of retrieval system architecture.

lie on the unit hypersphere. Nevertheless, non-linear measure can be used, but also non-vectorial descriptors, using the kernel function framework [4]. Let us note that this later case is considered during the design of the index similarity, described in Section 3.1.

2.3 Architecture

Retrieval problems can not be solved using a single processing chain, and many tools must be connected to shape a retrieval system architecture. There are as many architecture as there are target applications. However, similar parts and techniques can be seen in many cases. To illustrate some of these, we present in Fig. 1 an example of architecture for image categorization. Firstly, one can see there are two main parts: the indexing part, and the training part. This split is very common in retrieval systems, where we first index documents, and then exploit them in many ways. Now focusing on the indexing part, one can see on the left side that colors are extracted from images, in other words low-level descriptors are extracted from documents. Then, a dictionary is learned thanks to a quantization process. Note that dictionary is a very popular approach in multimedia indexing, but other tools could be considered. The last part of the indexing in our example is the projection of colors into a single histogram per image, in other words each set of descriptors are summarized into a single vector. The second main part of our example is the training part, where we aim at creating a classification function. More specifically, the objective is to use binary examples of cars and non-cars and the underlying index similarity to train a car detector.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

3 Unsupervised Indexing

In this section, we present indexing methods based on the assumptions we made in the previous section. That means they can be used to index any multimedia document as long as they is a tool to extract low-level descriptors. Furthermore, these techniques learn index structures from a representative set of multimedia documents, usually about several thousands of documents randomly drawn from the main dataset.

3.1 Index similarities

One of the main component for multimedia retrieval is the index metric, e.g. the operator that allows the comparison between two documents. Thanks to this metric, most retrieval problems can be solved, from similarity search to classification.

A lot of different metrics can be used, depending on low-level descriptors and the targeted application. In this paper, we focus on a popular approach based on kernel functions [4]. In this case, we consider equivalently the dot product as a similarity function, or similarly the Euclidean distance thanks to the mathematical relation between these two operators. Using this approach, we can assume that, whatever are the low-level descriptors, the similarity between two documents is a dot product, a.k.a. the kernel function. For instance, if we denote $\mathbf{B}_i = \{\mathbf{b}_{ri}\}_r$ the bag (or set) of low-level descriptors for document i, then we can build a kernel function that corresponds to a dot product in some Hilbert space [4]:

$$K_{softmax}(\mathbf{B}_i, \mathbf{B}_j) = \sum_{\mathbf{b}_{ri} \in \mathbf{B}_i} \sum_{\mathbf{b}_{sj} \in \mathbf{B}_j} \langle \mathbf{b}_{ri}, \mathbf{b}_{sj} \rangle \tag{1}$$

Using this strategy, we can turn a non-linear space (for instance, the space of descriptor bags) into an Hilbert space.

Coming back to multimedia retrieval problems, such kernel-based similarities can be used for interactive retrieval [5] or classification [6]. However, these techniques lead to a very high computational complexity. Consequently, an efficient strategy is to compute index whose similarity is very close to the kernel-based one. For instance, in the case of Eq. (1), a solution is an index $\mathbf{x}_i$ with the embedding function ϕ defined as $\mathbf{x}_i = \phi(\mathbf{B}_i) = \sum_r \mathbf{b}_{ri}$. In that case, the index metric is also the dot product, without any approximation: $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = K_{softmax}(\mathbf{B}_i, \mathbf{B}_j)$. In the following section, we will present several techniques in this scope.

3.2 Dictionary-based Indexing

A very effective way of comparing two multimedia documents is to use vote-based techniques. The main idea of these techniques is to count the number of matching low-level descriptors between two documents, and use this count as

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

similarity. The most basic case is:

$$K_{vote}(\mathbf{B}_i, \mathbf{B}_j) = \sum_r \sum_s matches(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \tag{2}$$

with *matches* a function that returns 1 if the two descriptors match, 0 otherwise.

In order to save computational time, dictionary-based methods can be used to build efficient index. The most popular is the Bag-of-Words (BoW) method [7]. This strategy first learns a clustering $\{S_c\}_{c\in[1,C]}$ of descriptor space using a random sampling of descriptors in database. Then, only descriptors in the same cluster S_c (called visual word, or codeword) are compared:

$$K_{BoW}(\mathbf{B}_i, \mathbf{B}_j) = \sum_c \sum_{\mathbf{b}_{ri}\in S_c} \sum_{\mathbf{b}_{sj}\in S_c} matches(\mathbf{b}_{ri}, \mathbf{b}_{sj}) \tag{3}$$

The learning of the codebook is a very challenging problem in itself, since the effect the the codebook (in particular the cluster assumption) on the retrieval performances is not clearly known.

In the case where the function *matches* always return 1, the corresponding index is then an histogram of words occurrences in the document. This idea is generalized using Coding/Pooling schemes, where we first consider a *coding* function that encodes any descriptors $\mathbf{b}_{ri}$ onto the codebook, and then we consider a *pooling* that aggregates all codes into a single index. The aim of these techniques is the same as all other ones: build an index whose metric is the closest possible to a straight matching process. In order to achieve such aim, the authors of [8] propose to compute sparse codes. This is achieved by optimising the following cost function:

$$code(\mathbf{b}) = \arg\min_{\mathbf{c}\geq 0} \parallel \mathbf{b} - \mathbf{D}^\top \mathbf{c} \parallel_2 + \alpha \parallel \mathbf{c} \parallel_1 \tag{4}$$

with $\mathbf{b}$ the descriptor to encode and $\mathbf{D}$ the dictionary. Note that this problem leads to sparse codes thanks to the ℓ_1 norm constraint, and adjusted thanks to the α parameter. Other techniques of descriptor encoding can be used, for example Wang et al. propose to perform a coding based on local linear approximation [9]. In all cases, codes are pooled using a sum or a max function. The problem can be extended to learn jointly both the encoding and the dictionary and is then known as *Dictionary Learning*.

3.3 Deviation-based Indexing

An interesting idea that was proposed next to dictionary-based techniques is the one based on model deviation. This idea is the following one. We first create a general model of data, usually the low-level descriptor space. Then, we build a document specific model for each document, and consider their index as the deviation between their model and the general model.

A good way to detail this idea and the underlying theoretical motivations, is to present the Fisher Vectors (FV) indexing method [10]. We first consider

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

a generative model $u_\lambda(\mathbf{b})$ for descriptors space, usually Gaussian Mixture Models (GMM). Then, we consider the bag of descriptors B_i from document i as a sampling of u_λ, and compute the gradient of the log-likelihood:

$$G_\lambda^{B_i} = \frac{1}{|B_i|} \nabla_\lambda \log u_\lambda(B_i) \tag{5}$$

The Fisher Kernel (FK) is then defined as:

$$K_{Fisher}(B_i, B_j) = (G_\lambda^{B_i})^\top F_\lambda^{-1} G_\lambda^{B_j} \tag{6}$$

with $F_\lambda = E_{\mathbf{b}\sim u_\lambda}(\nabla_\lambda \log u_\lambda(\mathbf{b}) \nabla_\lambda \log u_\lambda(\mathbf{b})^\top)$ the Fisher information matrix. Since F_λ is symmetric and positive definite, it can be factorized as $F_\lambda = L_\lambda^\top L_\lambda$, and K_{Fisher} can be rewritten as the dot product between Fisher Vectors (FV) $\mathcal{G}_\lambda^{B_i} = L_\lambda G_\lambda^{B_i}$ and $\mathcal{G}_\lambda^{B_j} = L_\lambda G_\lambda^{B_j}$. In the case where u_λ is a diagonal GMM, effective index can be computed thanks to this method, with very high performance for image categorization [11].

The idea of model deviation is also used in the Vectors of Locally Aggregated Tensors (VLAT), in addition to other propositions [12]. The first idea is the same as for most dictionary-based methods, with the novelty that a kernel function on bags K_B for each cluster c is considered, with $\mathbf{B}_{ic} = \{\mathbf{b}_{ric}\}_r = \mathbf{B}_i \cap \mathcal{S}_c$ the subset of descriptors in image i and cluster c:

$$K(\mathbf{B}_i, \mathbf{B}_j) = \sum_c K_B(\mathbf{B}_{ic}, \mathbf{B}_{jc}) \tag{7}$$

The second idea is the linearization of kernel functions on bags, in order to get an explicit expression of this metric. More specifically, the kernel on bags are considered with a Gaussian kernel and normalized descriptors (a usual tuning), expanded using Taylor series, and finally linearized using tensor products:

$$\begin{aligned} K_B(\mathbf{B}_{ic}, \mathbf{B}_{jc}) &= \textstyle\sum_r \sum_s e^{-\frac{1}{\sigma^2}||\mathbf{b}_{rci} - \mathbf{b}_{scj}||^2} \\ &= e^{-\frac{2}{\sigma^2}} \textstyle\sum_r \sum_s e^{\langle \mathbf{b}_{rci}, \mathbf{b}_{scj} \rangle} \\ &= e^{-\frac{2}{\sigma^2}} \textstyle\sum_r \sum_s \sum_p \frac{\alpha_p}{\sigma^{2p}} \langle \mathbf{b}_{rci}, \mathbf{b}_{scj} \rangle^p \\ &= e^{-\frac{2}{\sigma^2}} \textstyle\sum_p \frac{\alpha_p}{\sigma^{2p}} \langle \sum_r \otimes^p \mathbf{b}_{rci}, \sum_s \otimes^p \mathbf{b}_{scj} \rangle \end{aligned}$$

with $\otimes^p \mathbf{x}$ the tensor product of order p of a vector $\mathbf{x}$.

The third idea for VLAT is the deviation idea, i.e. the computation of the difference between descriptors and cluster centers $\boldsymbol{\mu}_c$. In the case $p = 2$, the index component for image i and cluster c is $\mathcal{T}_{ic} = \sum_r (\mathbf{b}_{rci} - \boldsymbol{\mu}_c)(\mathbf{b}_{rci} - \boldsymbol{\mu}_c)^\top - \mathcal{T}_c$ with $\mathcal{T}_c$ the mean tensor of cluster c.

Further extensions of deviation models consist in learning more accurate models of the descriptors space, as well as finding efficient similarity measures encoding the deviation between document specific models and the general model.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

4 Supervised Learning

In the previous methods, we only considered raw samples of multimedia documents. In this section, we assume that sampled documents are provided with additional data. In this paper, we will only consider category labels, the most popular annotations in multimedia retrieval. That means each training sample is considered as couple *(document,label)*. Let us note that the same document can be repeated in several samples.

4.1 Categorization

Multimedia document categorization aims at detecting an object or a high-level semantic concept, or category. The first solutions are based on a collection of binary classifiers, one for each category. In this scope, the most popular techniques are the discriminative ones that focuses on the visual content that brings out differences between categories. We can first cite usual classification method in machine learning, like k-Nearest-Neighbors (kNN), Fisher Discriminant Analysis, or Support Vector Machines (SVM), which can be easily used with indexes in Hilbert spaces.

Beside these usual classification methods, we can remark the ones based on the Boosting approach [13]. This approach is particularly interesting in multimedia retrieval context, because it allows an easier design of component in the retrieval system. Thanks to the idea of *weak classifiers* combination, it is no more required to design very complex indexes and corresponding similarities, as simple ones are sufficient. For instance, in the very popular face detector of Viola & Jones [14], the weak classifiers are based on a single Haar atoms in image space. The complexity of these weak components have to be compared to the high dimensional visual descriptors like the SIFT we presented earlier. The Boosting approach in still recent, and many improvements to the AdaBoost forerunner are still proposed every year, like Real AdaBoost, Gentle AdaBoost, RankBoost, AnyBoost, BrownBoost, ...

The usual way to train these classifiers is to use a random sample of labeled documents. However, using this basic approach requires large training set in order to improve the chance to get relevant labels for each category. As a result, the building of such retrieval systems can be very costly because of the time required to find and label multimedia documents. An effective strategy to reduce labeling time is to use *Active Learning*. This strategy is able to find inside large unlabeled datasets documents that, if they are labeled, may improve the most the results. A popular approach in this scope is version space reduction. The idea is to consider the version space, e.g. the space with all possible labeling, and to select documents that, if they are labeled, will shrink the most the version space. This approach have been proposed for SVM [15], Boosting [16], and may certainly be used with other classification techniques.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

4.2 Multiple Kernel Learning (MKL)

In Multimedia retrieval, a single index is usually not enough to respond to any kind of query. A common strategy is then to concatenate several indexes, and thus several low-level characteristics, to shape a single large index. Although this improves results, better performance can be achieved thanks to machine learning techniques.

In this scope, a popular techniques in Multimedia retrieval is the combination of kernel functions. There are several ways to combine these kernels, and the most used is a linear combination. In this context, first propositions suggest to perform a joint optimization of the kernel combination and the classifier, for instance a SVM classifier which is known as *Multiple Kernel Learning* (MKL) [17]. Recently, this approach have been criticized for several reasons, and two-stage optimization schemes are proposed. This can be also performed with SVM classifiers, where the kernel combination is first learned, and then hyperplane parameters are trained [18]. As for binary classification, the Boosting approach can also be used to learn kernel combination, with the specificity that weak kernels are combined [19].

The MKL techniques are still recent, and many improvements can be proposed. For instance, one of the key component of these methods is the evaluation criterion. Actually, for most optimization techniques, a function is required to evaluate the relevance of a kernel combination. The most popular one is the Kernel Alignment, defined as the cosine between two kernel matrices [20]. Even if improvement have been proposed, like centering for handling unbalanced category sizes, this criterion may be discussed, for instance because it can lead to overfitting. Furthermore, linear combinations have been first studied for their simplicity, however other kind of combinations could lead to better kernels [21].

5 Conclusion

Machine learning is only used for a short time for content-based multimedia retrieval. Nevertheless, we highlighted that first models and assumptions appeared. For instance, it is more and more common to assume that multimedia document are modeled as a set of low-level descriptors in an Euclidean space. Thanks to this kind of assumptions, it is relevant to create generalist methods, that can be used in many application cases. Furthermore, we can also observe similar results for document index. For instance, it is more and more common to assume that document index are vectors in a Hilbert space, even for multimedia document with a complex structure. Moreover, we also showed that this assumption can be also made in the case of non-linear index, using for example the kernel function framework. All of this results in an interesting foundation for machine learning research, and therefore it will lead to significant improvement in multimedia retrieval. In this scope, we presented in the last part several methods, from binary classification to multiple kernel learning. Moreover, emerging strategies such as *Deep Learning* jointly optimizing features extraction, mid-level representation and classification by using multiple layers of convolutional neu-

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

ral networks seem very promising according to recent benchmarks [22]. These works show that supervised learning approaches can be used for any stage of the multimedia indexing process.

Beyond the retrieval problem we presented, other paradigms are recently explored or should be studied. For instance, computational and memory complexity are major concerns for retrieval systems. If the former is already notably studied, the latter has only been targeted very recently, for instance using techniques based on encoding. Another aspect that is poorly considered is the computing architecture where retrieval systems are deployed. The usual assumptions made about these architectures are, for instance, an unlimited access to data, or the presence of a central control unit. However, in order to manage and store the largest datasets, these architectures can not be used. The current solution is to used distributed and decentralized architectures. As a result, the most current learning models are no more relevant, and need to be adapted. Let us note that these new constraints do not necessarily depend on the kind of multimedia data. It means that it highlights new generalist machine learning problems, and new solutions to be invented.

References

[1] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, 2004.

[2] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *European Conference on Computer Vision*, 2004.

[3] Hedi Tabia, Mohamed Daoudi, Olivier Colot, and Jean-Philippe Vandeborre. 3d-object retrieval based on vector quantization of invariant descriptors. *Journal of Electronic Imaging*, 21, 2012.

[4] J. Shawe-Taylor and N. Cristianini. *Kernel methods for Pattern Analysis*. Cambridge University Press, ISBN 0-521-81397-2, 2004.

[5] P.-H. Gosselin, M. Cord, and S. Philipp-Foliguet. Kernel on bags for multi-object database retrieval. In *ACM International Conference on Image and Video Retrieval*, pages 226–231, Amsterdam, Netherlands, July 2007.

[6] Z. Harchaoui and F. Bach. Image classification with segmentation graph kernels. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.

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

[8] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, volume 19, pages 801–808, 2007.

[9] Jingjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.

[10] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(7):1243–1256, July 2008.

[11] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, pages 143–156, 2010.

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0.
Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

[12] David Picard and Philippe-Henri Gosselin. Improving image similarity with vectors of locally aggregated tensors. In *IEEE International Conference on Image Processing*, Brussels, Belgique, September 2011.

[13] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[15] S. Tong and D. Koller. Support vector machine active learning with application to text classification. *International Journal on Machine Learning Research*, 2:45–66, November 2001.

[16] X. Li, L. Wang, and E. Sung. Improving adaboost for classfication on small training sample sets with active learning. In *Asian Conference on Computer Vision*, Jeju, Korea, January 2004.

[17] Francis R. Bach and Gert R. G. Lanckriet. Multiple kernel learning, conic duality, and the smo algorithm. In *International Conference on Machine Learning*, 2004.

[18] C. Cortes, M. Mohri, and A. Rostamizadeh. Two-stage learning kernel algorithms. In *International Conference on Machine Learning*, 2010.

[19] A. Lechervy, P.H. Gosselin, and F. Precioso. Linear kernel combination using boosting. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Brussels, Belguim, April 2012.

[20] N. Cristianini, J. Shawe-Taylor, A. Elisseff, and J. Kandola. On kernel target alignement. In *Advances in Neural Information Processing Systems*, pages 367–373, Vancouver, Canada, December 2001.

[21] David Picard, Nicolas Thome, Matthieu Cord, and Alain Rakotomamonjy. Learning geometric combinations of gaussian kernels with alternating quasi-newton algorithm. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2012.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.