Pattern Recognition 44 (2011) 2343–2357

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

SALSAS: Sub-linear active learning strategy with approximate *k*-NN search

D. Gorisse [a,*], M. Cord [b], F. Precioso [a]

[a] ETIS, ENSEA/CNRS UMR8051/Univ. Cergy-Pontoise, France
[b] LIP6, UPMC - PARIS VI, France

ARTICLE INFO

Available online 21 December 2010

Keywords:
Image retrieval
Active learning
Relevance feedback
Locality sensitive hashing
Scalability
Support vector machines

ABSTRACT

With the democratization of digital imaging devices, image databases exponentially grow. Thus, providing the user with a system for searching into these databases is a critical issue. However, bridging the *semantic gap* between which (semantic) concept(s) the user is looking for and the (semantic) content is quite difficult. In content-based image retrieval (CBIR) systems, a classic scenario is to formulate the user query, at first, with only one example (*i.e.* one image). In order to address this problem, active learning is a powerful technique which involves the user in interactively refining the query concept, through *relevance feedback* loops, by asking the user whether some strategically selected images are relevant or not. However, the complexity of state-of-the-art active learning methods is linear in the size of the database and thus dramatically slows down retrieval systems, when dealing with very large databases, which is no longer acceptable for users. In this article, we propose a strategy to overcome scalability limitations of active learning strategies by exploiting ultra fast *k*-nearest-neighbor (*k*-NN) methods, as locality sensitive hashing (LSH), and combining them with an active learning strategy dedicated to very large databases. We define a new LSH scheme adapted to χ^2 distance which often leads to better results in image retrieval context. We perform evaluation on databases between 5 K and 180 K images. The results show that our interactive retrieval system has a complexity almost constant in the size of the database. For a database of 180 K images, our system is 45 times faster than exhaustive search (linear scan) reaching similar accuracy.

1. Introduction

In content-based image retrieval (CBIR), the search is usually initiated using one example as query of targeted data. Most of CBIR systems are considering retrieval task as a binary classification task: the relevant class represents the class of data similar to the query and the irrelevant class is the rest of the database. The system presents to the user the most relevant images (the top ranked images) with respect to their similarity to the query, using the learned classifier. All retrieval methods are thus deeply depending on the relevance of the training set $\mathcal{A}$ on which the optimal classification function is learned. Supervised learning techniques are concerned with optimizing this training set in order to improve the classification. Semi-supervised learning algorithms focus on how to incorporate unlabeled data into the training process to build a better training set. For instance, Cheng et al. [1] propose to learn two distinct SVM classifiers, one in color feature subspace, another one in texture feature subspace, then unlabeled data which are classified in different classes in the two subspaces are selected to be annotated. On-line learning gathers all the methods which focus on optimizing the ranking of results by improving the training set $\mathcal{A}$ in order to update the classification function. Among on-line learning approaches, we consider here active learning strategies [2] which aim at minimizing classification error over the whole set $\mathcal{B}$ of images by selecting which elements from the unlabeled data pool are the most informative and thus should be annotated to improve the most the classification. This specific process, compared to simple classification methods, is called *sampling strategy* in [3]. The user, considered as an expert, must then iteratively annotate as positive (relevant) or negative (irrelevant) strategically selected images [4], in a process called *relevance feedback loop* [5,6]. Such strategies are particularly relevant in image interactive retrieval context since only few annotations should be required from the user to define the training set. As a consequence the training set is small, new annotated data must thus provide highest classification improvements. Many interaction strategies between the user and the system have been proposed [7]. In [8], an hybrid strategy exploits together semi-supervised learning technique and active learning strategy. In this work, two learners (one using l_1 distance and the other l_2 distance) estimate relevance of some unlabeled images. Then, the selection sampling gathers all the resulting labeled images with high uncertainty after merging classification ranking of both learners (relevance summation).

However, for the aforementioned methods, ranking the whole database at each iteration is thus required to show the results.

* Corresponding author.
E-mail addresses: david.gorisse@ensea.fr (D. Gorisse), matthieu.cord@lip6.fr (M. Cord), frederic.precioso@lip6.fr (F. Precioso).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.12.009

When the database becomes very large, this sorting process becomes intractable and represents the first scalability lock of active learning retrieval systems.

The strategy selection has, at best, a linear computational complexity in the number of images in the database and is also processed at each relevance feedback loop. This is the second scalability lock of active learning retrieval systems when the database becomes large.

Active learning strategy can be improved with transductive approaches that take into account unlabeled data in the optimization scheme. However, these optimizations usually lead to slow algorithms and intractable solutions when dealing with large databases. Even if, as in [9], greedy solutions to build better scalable transductive approach exist, the search is at best linear with the size of the database.

None of these methods achieve to break the complexity down to sub-linear scheme in the size of the database.

In our previous work [10], we proposed a strategy to quickly select, in a large database, relevant images to be annotated using an Euclidean LSH scheme. Our first results on addressing both scalability locks, sample selection and data ranking, of CBIR systems for large databases were promising. In this paper, we present a new sub-linear search approach which outperforms the previous one: we build a pool of relevant data on which focusing both the sample selection and the ranking process to make active learning strategies scalable in very large database context; we propose a strategy to quickly update this pool in order to explore the feature space; we propose a brand new LSH scheme defining new hash functions dedicated to χ^2 distance which proved to often lead to better results for image retrieval task [11]; we test our approach on a database of 180 K images and show how our active learning strategy, combined with a kernel-based SVM approach, can effectively address interactive content-based image retrieval in very large databases.

2. Active learning for CBIR

Since, in this paper, we consider global descriptors of images (color or texture histograms, etc.), we will use equivalently the terms vectors or images indexes.

In CBIR classification framework, retrieving classes of images is usually considered as a two-class problem: the relevant class, the set of images corresponding to the user query concept, and the irrelevant class composed by the remaining of the database. Let $\{\mathbf{x}_i\}_{1,n}$ be the n image indexes of the database. A training set is expressed from any user label retrieval session as $\mathcal{A}=\{(\mathbf{x}_i,y_i)_{i=1,n}|y_i\neq 0\}$, where $y_i=1$ if the image $\mathbf{x}_i$ is labeled as relevant, $y_i=-1$ if the image $\mathbf{x}_i$ is labeled as irrelevant. The classifier is then trained using these labels, and a relevance function $f_{\mathcal{A}}(\mathbf{x}_i)$ is determined in order to be able to rank the unlabeled images. The set of unlabeled images is denoted by $\mathcal{U}=\{(\mathbf{x}_i,y_i)_{i=1,n}|y_i=0\}$.

We use a support vector machine (SVM) [12] as learning algorithm for its effectiveness to learn, in a binary classification context, with very few examples. The relevance function $f_{\mathcal{A}}$ obtained with SVM define an hyperplane that separates the relevant and irrelevant images in the training set $\mathcal{A}$ by a maximal margin. All images lying on the positive side of the hyperplane are considered as relevant and all images on the other side (negative one) are considered as irrelevant. The closer to 0 an image score is, the greater its label uncertainty. Higher the score of an image is, higher its relevance to the target class. As such a linear separation (hyperplane) does not often exist in the input space $\mathcal{R}^d$, SVMs are generally combined with kernel functions. A kernel K is a similarity function that maps image indexes in a Hilbert space $\mathcal{H}$ (often higher dimensional) called *feature space* where a linear separator is more likely to exist. A popular kernel function is the RBF kernel $K(\mathbf{x}_i,\mathbf{x}_j)=\exp(||\mathbf{x}_i-\mathbf{x}_j||^2/2\sigma^2)$ which maps image indexes into an infinite dimensional space that enforces the existence of a such hyperplane. The relevance function $f_{\mathcal{A}}$ is then defined by $f_{\mathcal{A}}(\mathbf{x})=\sum_{i=1}^{|\mathcal{A}|}y_i\alpha_i K(\mathbf{x}_i,\mathbf{x})$. The learning procedure determines the scalar weights, α_i, associated with the training instances. In active learning classification, the user is considered as an expert and is thus involved in the process of minimizing the classification error over the whole database $\mathcal{B}$. The user iteratively annotates strategically selected images among unlabeled data $\mathcal{U}$ in $\mathcal{B}$ in order to build an "optimal" training set $\mathcal{A}$. This expert can be represented by a function $s:\mathcal{B}\rightarrow\{-1,1\}$, which assigns a label to an image of the database. In the case where only one image $\mathbf{x}_i$ has to be selected, this turns to the minimization of classification error on $\mathcal{B}$, over all the functions $f_{\mathcal{A}}$ of classification on the previous training set $\mathcal{A}$ augmented with the annotation $s(\mathbf{x}_i)$ of the image $\mathbf{x}_i$:

$$i^\star=\arg\min_{i\in\mathcal{U}}R_{test}(f_{\mathcal{A}\cup\{(\mathbf{x}_i,s(\mathbf{x}_i))\}}) \tag{1}$$

with $R_{test}(f_{\mathcal{A}})$ a risk function whose definition depends on the approximation introduced in its evaluation. For instance, Roy and Mc Callum [13] propose a technique to determine the data $\mathbf{x}_i$ which, once added to the training set $\mathcal{A}$ with the user annotation $s(\mathbf{x}_i)$, minimizes the error of generalization. This problem cannot be directly solved, since the user annotation $s(\mathbf{x}_i)$ of each $\mathbf{x}_i$ image is unknown. Roy and Mc Callum [13] thus propose to approximate the risk function $R_{test}(f_{\mathcal{A}})$ for both possible annotations, positive and negative. The labels $s(\mathbf{x}_i)$, unknown on $\mathcal{U}$, are estimated by training 2 classifiers for both possible labels on each unlabeled data $\mathbf{x}_i$.

Another selection strategy has been proposed by Tong et al. [14]. Their SVM_{active} method is based on the minimization of the set of separating hyperplanes. Their idea is to focus on the most uncertain data $\mathbf{x}$: $f_{\mathcal{A}}(\mathbf{x})\sim 0$. The solution to the minimization problem in Eq. (1) is then:

$$i^\star=\arg\min_{\mathbf{x}_i\in\mathcal{U}}(|f_{\mathcal{A}}(\mathbf{x}_i)|) \tag{2}$$

For all these methods, once the image $\mathbf{x}_{i^\star}$ is labeled, this image is added to the training set $\mathcal{A}$. The SVM is then retrained to update the relevance function $f_{\mathcal{A}}$.

3. Scalability issues in CBIR

As long as the number of iterations is reasonable, the training set $\mathcal{A}$ is small thus the complexity of retraining the classifier is negligible. However, when $|\mathcal{A}|$ increases too much, the complexity of this step becomes not negligible, then efficient SVM formulation as LASVM [15]can be considered. Wang et al. [16] propose a quite different approach based on a quick test scheme to select samples to be annotated. They extend an existing on-line kernel learning algorithm adapted to training samples arriving one-by-one. $f_{\mathcal{A}_{t+1}}$ is immediately obtained from a correction of $f_{\mathcal{A}_t}$ at tth relevance feedback loop and an upper bound determines the maximum test scope required in each loop without sacrificing retrieval performance.

In the context of on-line learning $\mathcal{A}$ remains small thus, we do not face such scalability issue.

The second scalability lock of CBIR systems, in the context of very large databases, concerns the ranking of the database to retrieve the top N most relevant images, called TOPN. Retrieving the TOPN when the query is defined by only one image example, is considered as a "solved problem" since efficient solutions, based on fast k-NN search like LSH [17], have been proposed.

When the query concept is refined by adding new positive query images, the similarity between positively labeled images and unlabeled data is more complex to evaluate. In this context, several approaches have been proposed to address the scalability issue of

retrieving quickly TOPN most relevant images. In [18], a One-class SVM is trained on positive samples in order to estimate the distribution of relevant image class. In [19], in order to speed up retrieving relevant images from this distribution, this process exploits an M-tree in the feature space. In [20], authors propose to use another index structure KVA-file to provide an efficient nearest neighbor search using kernel distances. However, if we consider the number of relevance feedback loops required to completely identify the target class then 2-class SVM with active learning significantly outperforms One-class SVM.

When the relevant function is defined with a 2-class SVM, the strategy to quickly retrieve the TOPN is more complex. Indeed, the most relevant images are both closest to the positively labeled images and farthest from negative examples. In [21], authors propose to approximate the frontier of the 2-class SVM by defining regions in the input space (space of the descriptors) with bounding boxes (hyper-rectangles) delimited by positive and negative images. They adapt an indexing structure based on R-tree to quickly search images that are close to the center of theses bounding boxes to efficiently retrieve the TOPN. However, the input space can be high-dimensional. Consequently, the proposed R-tree based approach becomes intractable.

Wang et al. [16] propose a quite different approach based on a quick test scheme to only update the relevance of images that can belong to the top of the search at the next iteration. $f_{\mathcal{A}_{t+1}}$ is hence obtained from a correction of $f_{\mathcal{A}_t}$. The quick test scheme allows to compute, at the tth iteration, an upper bound defining the maximal correction for the next iteration. This bound allows to select images that can have a higher score, at the $(t+1)$th iteration, than the TOPN of the tth iteration. This method allows to decrease the complexity of the ranking step after the first iteration. However, at the first iteration, the ranking of all the database is required thus the complexity of the first iteration is still linear regarding the database size.

In [22], in order to avoid ranking all the database, authors build a new index structure KDX [23] designed for providing most relevant data (farthest from the query hyperplane on "relevant side"). This index structure has important drawbacks. Indeed, KDX has no stopping rules. Then, the search terminates when the wall-clock time assigned to run the TOPN expires. Moreover, there are some critical cases where the search does not converge to the real nearest neighbor. Furthermore, the search quality depends a lot on the distance between the query and the central instance in the induced space which controls the clustering stage.

Focusing on active learning strategies for CBIR systems in the context of very large databases, selecting "optimal" images for annotation becomes intractable. Indeed, the complexity of state-of-the-art techniques, described in previous section, is directly depending on $\mathcal{U}$, the set of unlabeled data. Roy and Mc Callum [13] approach implies a computational complexity of $O(|\mathcal{U}|^2)$, while Tong et al. [14] approach, even though more efficient, has still a computational complexity of $O(|\mathcal{U}|)$.

In [24], Crucianu et al. propose a search method based on an M-tree built in the feature space associated with a kernel-based 2-class SVM. The M-tree structure allows to quickly retrieve the k-NN images closest to the frontier hyperplane resulting from the SVM. However, when the number of examples increases, the frontier becomes very complex and the number of M-tree leaves to be visited increases highly which slows down the process. In [22] for the sampling selection, the authors propose to first explore the feature space by clustering it. Then they consider the nearest clusters to the discrimination frontier based on query hyperplane, defined by the the SVM, in order to select the most uncertain data for annotation. This last process is still problematic when considering very large database without control over the number of clusters.

To the best of our knowledge, only Panda et al. [22] address both scalability issues of active learning strategies for 2-class-SVM-based retrieval systems in the context of very large databases, *i.e. sampling selection* and *database ranking*. However, as already mentioned, this method suffers from important drawbacks.

In this paper, we propose a new approach to break both active learning scalability locks, in very large database context. Our scalable fast selection strategy allows to exploit active learning strategies combined with a kernel-based SVM for retrieving images based on their content in very large databases. We exploit ultra fast index structure as locality sensitive hashing (LSH) which has already proven its efficiency for similarity search in huge databases.

4. Sub-linear scheme

In interactive CBIR, one can notice that most of the time, the user is interested in the top of the ranking of the whole database. Only the rank of the N most relevant images, called TOPN, is useful (usually, N is fixed by the user). We would like to exploit this specificity of on-line learning process in interactive CBIR to overcome the problem of scalability related to the database size. The ranking of the whole database implies "to see" at least once each image of the database. Our idea is to shortcut this process by selecting a pool of images, called $\mathcal{S}$, which, thanks to heuristics, are more than likely to be among the TOPN. In order to be really sub-linear and not seeing (even shortly) all images of the database, we will need to carry out efficient indexing structure of the database.

When considering only this pool $\mathcal{S}$ for the following interactive learning process, we can break the complexity of the search algorithm.

We are going to detail our selective subsampling strategy based on heuristics before explaining the active learning process working on $\mathcal{S}$.

4.1. Selective subsampling strategy

The relevance of an image $\mathbf{x}$ is estimated by the relevance function $f_{\mathcal{A}}$ (see Section 2): $f_{\mathcal{A}}(\mathbf{x}) = \sum_{i=1}^{|\mathcal{A}|} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x})$. Looking for TOPN images means finding the N highest values for $f_{\mathcal{A}}(\mathbf{x})$. This function may be split into:

$$f_{\mathcal{A}}(\mathbf{x}) = f_{\mathcal{A}^+}(\mathbf{x}) - f_{\mathcal{A}^-}(\mathbf{x}) = \sum_{p=1}^{|\mathcal{A}^+|} \alpha_p K(\mathbf{x}_p, \mathbf{x}) - \sum_{n=1}^{|\mathcal{A}^-|} \alpha_n K(\mathbf{x}_n, \mathbf{x}) \tag{3}$$

where $\mathcal{A}^+$ denotes positive labeled images: $(\mathbf{x}, +1) \in \mathcal{A}$ and $\mathcal{A}^-$, the negative ones: $(\mathbf{x}, -1) \in \mathcal{A}$.

Our strategy is to replace the $f_{\mathcal{A}}$ optimization by focusing on $f_{\mathcal{A}^+}$. More precisely, we assume that if an image $\mathbf{x}$ is close to one positive training example $\mathbf{x}_p \in \mathcal{A}^+$, $\mathbf{x}$ has good chances to get a high $f_{\mathcal{A}^+}(\mathbf{x})$ value, and thus a high $f_{\mathcal{A}}(\mathbf{x})$ score. Of course, this assumption is not true for every $\mathbf{x}$, since α coefficients and $f_{\mathcal{A}^-}(\mathbf{x})$ may affect a lot the initial score, this is why candidates are filtered in a second step.

The first step of our process, called *selection*, is to get images from the unlabeled image set $\mathcal{U}$ which are close to one of positive training examples $\mathcal{A}^+$. We carry out this step by using a k-NN search for all data in $\mathcal{A}^+$ which can be achieved with a complexity sub-linear regarding the size of the whole database $\mathcal{B}$ (or the size of $\mathcal{U}$). This is the main motivation of our *selection* strategy. Therefore, we quickly collect a lot of candidates. Even though many of them are not good, the filtering process will clean this set.

The second step of our selective subsampling strategy, called *pruning*, aims at filtering the k-NN search output. As already noted, a candidate $\mathbf{x}$ may finally have a poor $f_{\mathcal{A}}(\mathbf{x})$ score. This step consists in computing the exact relevance of the selected images by using $f_{\mathcal{A}}$ on the selected images and keeping only the p most relevant images. When the resulting pool of images $\mathcal{S}$ of size $p = |\mathcal{S}|$ is large enough ($p \geq N$), the approximate pool of the TOPN images is simply

extracted from $\mathcal{S}$. The pool $\mathcal{S}$ will be also used in our active learning strategy presented in the next section.

Our strategy is based on heuristics, we have no theoretical guarantee to really find exact TOPN, but this solution is scalable when the database is too large to allow linear search.

As aforementioned, one of the crucial conditions for our scheme to be effective is that the k-NN search has to be fast. Instead of doing a linear scanning, we use an efficient indexing scheme based on *LSH*, which will be detailed in the next section. As *LSH* is based on l_1 and l_2 metric, to be consistent with the learner, the kernel considered must be based on these distances. However, it was observed that the quality of retrieval achieved using l_2 is not always satisfactory while χ^2 distance proved to often lead to better results for image and video retrieval task [25,11]. In the next section, we extend *LSH* to the χ^2 distance that allows us to use any kernel based on χ^2 distance.

4.2. Active learning labeling strategy

The strategy described above provides a solution to overcome the scalability lock of the ranking stage, we are now focusing in this section on the second lock mentioned in Section 3: the scalability issue of the sampling stage for on-line labeling during the retrieval process for interactive CBIR systems. Our aim is to define a strategy in order to select, from the unlabeled dataset $\mathcal{U}$, the best image(s) that, once labeled and thus added to the training set, will lead the relevance function to optimally improve for the next search round.

The state-of-the-art active sampling strategies have been described in Section 2. As aforementioned in Section 3, all these strategies are computationally demanding when the size of the database increases. One of the most efficient techniques is Tong's strategy [14], given by Eq. (2). Its computational complexity is linear in size of the dataset $\mathcal{U}$. The image $\mathbf{x}_i^*$ is the closest one to the boundary between the relevant and irrelevant classes estimated using the current classifier $f_{\mathcal{A}}$.

In order to be scalable, we propose to take benefit of our previous process defined to approximate the top ranking images. Actually, our strategy is based on the use of the set $\mathcal{S}$ instead of $\mathcal{U}$ to find the best candidates for labeling. As explained before, this strategy is very computationally efficient, but there are also other motivations to focus on $\mathcal{S}$:

- First of all, note that $\mathcal{S}$ is used to approximate the TOPN but is not restricted to the TOPN. In other words, the set $\mathcal{S}$ may be quite larger than the approximate TOPN extracted from $\mathcal{S}$. Actually, we keep the size p of $\mathcal{S}$ as one of the main parameters of our system to tune the trade-off between efficacy and efficiency.
- The classification problem considered here is very unbalanced. Indeed, in huge databases, we know that the size of relevant image subset is a lot smaller than the size of irrelevant image subset. It follows that a positively annotated image is more likely to be close to the center of the relevant class than a negatively annotated image. By focusing not too far from positive examples, *i.e.* in $\mathcal{S}$, we then increase our chances to select positive images and thus to re-balance the problem.
- As long as the user is not satisfied by the results, this means that the relevance function $f_{\mathcal{A}}$ is not enough accurate to retrieve enough relevant images. Consequently the pool $\mathcal{S}$ contains several uncertain images which will help to improve the relevant function $f_{\mathcal{A}}$ through interactive process.

For the optimization scheme in $\mathcal{S}$, we do not solve the optimization proposed in Eq. (2), but we follow instead the extension of this work, proposed in [26]. The author incorporates a diversity metric into sampling selection that outperforms previous methods. This method is named *angle diversity* and represents now, for sampling selection, the state of the art. The main idea is to select the image that is the most uncertain while at the same time is the less similar to already labeled images $\mathcal{A}$.

In our scheme, the solution to the minimization problem in Eq. (1) is rewritten over $\mathcal{S}$ as

$$i^* = \arg\min_{\mathbf{x}_i \in \mathcal{S}} \left(\lambda * |f_{\mathcal{A}}(\mathbf{x}_i)| + (1-\lambda) \left(\max_{\mathbf{x}_j \in \mathcal{A}} \frac{|K(\mathbf{x}_i, \mathbf{x}_j)|}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i) K(\mathbf{x}_j, \mathbf{x}_j)}} \right) \right) \quad (4)$$

λ is a parameter which allows to balance the diversity and the uncertainty. We fix this parameter to $\frac{1}{2}$ in all our experiments.

The active learning labeling strategy is reprocessed at each relevance feedback loop.

4.3. SALSAS algorithm

The scheme of our strategy is summarized in Fig. 1 and each block is described as follows: The system is initialized with some *query images* provided by the user which are added to the training set $\mathcal{A}$ (*labeled dataset*) to learn a relevant function $f_{\mathcal{A}}$ with a SVM classifier (*training*). At the same time, the positively labeled images ($y > 0$) are used to perform a fast *k-NN search* in order to quickly retrieve images from the *unlabeled dataset* $\mathcal{U}$ close to positive examples.

For the initial loop, the *selection* step consists just in initializing the pool $\mathcal{S}$ with the images retrieved by the k-NN search. The pool $\mathcal{S}$ is then ranked in the *pruning* block using the relevance function $f_{\mathcal{A}}$. Only the p most relevant images are kept. The *pruning* block ensures thus that the size of $\mathcal{S}$ remains constant by removing the most

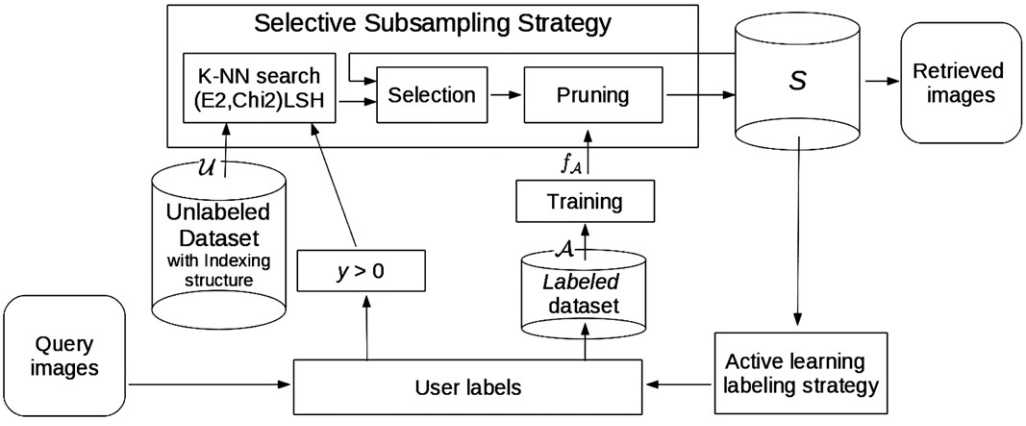


Fig. 1. Scheme of fast active learning.

irrelevant images. The TOPN is then shown to the user (*Retrieved images*). The *Active learning labeling strategy* block selects the most uncertain images of $\mathcal{S}$ to be labeled by the user. The system iterates as long as the user is not satisfied by the search.

After each loop, the relevance function $f_{\mathcal{A}}$ is retrained thus the pool $\mathcal{S}$ has to be updated. The update of $\mathcal{S}$ is done in *selection* block by approximating $f_{\mathcal{A}^+}$ of Eq. (3). Images of $\mathcal{U}$ close to one of the positive training examples from $\mathcal{A}^+$ should hence be retrieved. However, noticing that $\mathcal{S}$ already contains such images of $\mathcal{U}$ from previous iterations, this step is speeded up by only adding the k-NN of the new positively labeled images ($y>0$) to $\mathcal{S}$. Then, the *pruning* step filters the pool $\mathcal{S}$. The TOPN images of $\mathcal{S}$ can then be presented to the user as intermediate result. Finally, the *Active learning labeling strategy* block selects the most uncertain image of $\mathcal{S}$ to be labeled by the user. The system iterates as long as the user is not satisfied by the search.

4.4. Algorithm and complexity

More details for implementation are given in algorithm 1. We consider here that the search is initialized with only one query image example I_q, considered as positively annotated. On line 2, we first fill the unlabeled data pool $\mathcal{S}$ with p-NN images of the query image. The complexity of this stage is $O(n^{\rho})$ with $\rho<1$ that depends on the index structure used. At each feedback iteration, depending on the presence (or not) of negative labels, the relevant function $f_{\mathcal{A}}$ is updated, from line 4 to 8. The complexity of the training stage is $O(\mathcal{A}^2)$. As the number of images annotated by the user is small ($|\mathcal{A}| \ll n$), the complexity of this stage is negligible. On line 9, the pool $\mathcal{S}$ is re-ranked with respect to this new $f_{\mathcal{A}}$. The complexity of the ranking stage is $O(|\mathcal{S}|.log(|\mathcal{S}|))$. At this stage, $\mathcal{S}$ contains at most $(p+k) \ll n$ images. The complexity of this stage is thus negligible. In order to preserve a constant size for $\mathcal{S}$, we only keep the p most relevant images, on line 11. On line 13, the TOPN of $\mathcal{S}$ is then presented to the user as preliminary retrieval results. Our active learning labeling strategy, on line 15, then selects b images in this pool $\mathcal{S}$ to be presented to the user for annotation using Eq. (4). The complexity of the selection stage is $O(|\mathcal{S}|.log(|\mathcal{S}|))$ with $|\mathcal{S}|=p$. The complexity of this stage is negligible. The relevant function $f_{\mathcal{A}}$ is thus improved. On line 19, the pool $\mathcal{S}$ is then updated by adding the k-NN images for each new positively annotated images. The complexity of this stage is $O(i_p.n^{\rho})$ with i_p the number of positively annotated images at the current feedback iteration ($0 \le i_p \le b$). To put it in a nutshell, the complexity of our fast scheme is $O(n^{\rho})$ and only depends on the indexing structure used to perform the k-NN search.

Algorithm 1.

Require: $I_q, \mathcal{U}, k, p, b$; /* Query image, Unlabeled set, number of NN, pool size, batch size */
1 $\mathcal{A} \leftarrow (I_q, +1)$
2 $\mathcal{S} \leftarrow p-\text{NN}(I_q)$
3 **loop**
4 **if** $\forall(\mathbf{x}_s,y_s) \in \mathcal{A}$, $\exists s \mid y_s<0$ **then**
5 $f_{\mathcal{A}} \leftarrow$ 2-class SVM($\mathcal{A}$)
6 **else**
7 $f_{\mathcal{A}} \leftarrow$ 1-class SVM($\mathcal{A}$)
8 **endif**
9 sort($\mathcal{S}$) (by computing $f_{\mathcal{A}}(\mathbf{x}_i)$ $\forall \mathbf{x}_i \in \mathcal{S}$)
10 **if** $|\mathcal{S}|>p$ **then**
11 remove $\{\mathcal{S}_r\}_{r \in [p,\ldots,|\mathcal{S}|]}$
12 **endif**
13 show TOPN of $\mathcal{S}$
14 **for** $a=0$ to b **do**
15 $\mathbf{x}_s \leftarrow \arg\min_{\mathbf{x}_i \in \mathcal{S}}(\frac{1}{2}*|f_{\mathcal{A}}(\mathbf{x}_i)| + \frac{1}{2}(\max_{\mathbf{x}_j \in \mathcal{A}} \frac{|K(\mathbf{x}_i,\mathbf{x}_j)|}{\sqrt{K(\mathbf{x}_i,\mathbf{x}_i)K(\mathbf{x}_j,\mathbf{x}_j)}}))$
16 $y_s \leftarrow$ user label $\{-1,+1\}$
17 $\mathcal{A} \leftarrow \mathcal{A} \cup (\mathbf{x}_s,y_s)$
18 **if** $y_s>0$ **then**
19 $\mathcal{S} \leftarrow \mathcal{S} \cup k-\text{NN}(\mathbf{x}_s)$
20 **endif**
21 **endfor**
22 **endloop**

5. Indexing structure

In this section we present an overview of LSH before providing details on our new hash function for χ^2 distance. The intuition behind LSH is to use hash functions to map vectors into buckets, such that nearby vectors are much more likely to map into the same bucket than vectors that are far apart. A similarity search consists in finding the bucket B that the query q hashes into, selecting candidates, *i.e.* vectors in B, and returning the k-Nearest Neighbors (k-NN) candidates of q.

5.1. Basics

LSH was first introduced by Indyk and Motwani in [27] for the Hamming metric. They defined the requirements for a hash function family to be considered locality-sensitive. Let D be the domain of vectors and d the distance measure between vectors.

Definition. A function family $\mathcal{H}=\{h: D \rightarrow U\}$ is called $(r,cr,p_1,p_2)-$sensitive, with $c>1$ and $p_1>p_2$, for d if for any $p,q \in S$

- if $d(q,p) \le r$ then $Pr_{\mathcal{H}}[h(q)=h(p)] \ge p_1$ and
- if $d(q,p)>cr$ then $Pr_{\mathcal{H}}[h(q)=h(p)] \le p_2$.

Intuitively, the definition states that nearby vectors (those within distance r) are more likely to collide ($p_1>p_2$) than vectors that are far apart (those with a distance greater than cr).

To decrease the probability of false detection p_2, several sensitive functions h are concatenated to define a hash function g. For an integer M, let us defining a hash function family $\mathcal{G}=\{g: \rightarrow U^M\}$ such that $g(p)=(h_1(p),\ldots,h_M(p))$, where $h_i \in \mathcal{H}$. However, as a consequence, the probability of good detection p_1 decreases too. To compensate the decrease in p_1, several function g are used. For an integer L, choose $g_1,\ldots,g_L$ from $\mathcal{G}$, independently and uniformly at random. Each of the L functions g_i is used to construct one hash table, resulting in L hash table.

We now present the two sensitive functions used in this paper.

5.2. Euclidean metric hashing

Datar et al. [28] proposed LSH families for the Euclidean metric. In this article, we will call this method *E2LSH*.

The sensitive function works on tuples of random projections of the form: $h_{\mathbf{a},b}(\mathbf{p})=\lfloor(\mathbf{a}.\mathbf{p}+b)/W\rfloor$ where $\mathbf{a}$ is a random vector whose each entry is chosen independently from a Gaussian distribution, b is a real number chosen uniformly in the range $[0,W]$ and W specifies a bin width.

Each projection splitting the space by a random set of parallel hyperplanes; the value of the hash function indicates in which slice of each hyperplane the vector has fallen.

The three parameters chosen for this algorithm are the size of the search windows W, the number of projections M and the number of hash tables L.

5.3. χ^2 metric hashing

If the Euclidean locality sensitive hashing algorithm, approximating k-NN search in an euclidean space with a sub-linear

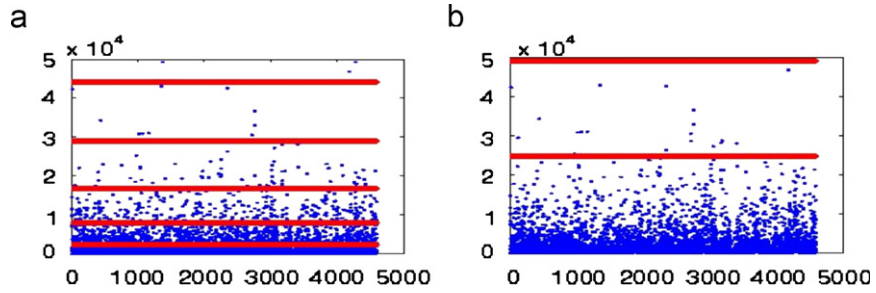


Fig. 2. Partition of 4500 feature vectors (spread along x-axis) extract from VOC2006 database according to a line $\mathcal{R}^+$: (a) distance χ^2 and (b) distance l_2.

complexity, is probably the most popular approach, euclidean metric does not always provide as accurate and relevant results, when considering similarity measure, as χ^2 distance. In this paper, we define a new LSH scheme adapted to χ^2 distance, preserving the same efficiency than Datar for E2LSH. Therefore, we map all the vectors in a space of smaller dimension and cluster this sub-space. The clusterization must ensure that the probability of two vectors falling into the same bucket (a cluster cell) is higher, when the distance between these two vectors is small, than when the distance is upper a certain threshold. In our case, the sub-space is the line $\mathcal{R}^+$. This line is obtained by projecting all vectors on a random vector $\mathbf{a}$ whose each entry is chosen independently from a Gaussian distribution. We uniformly partition this line with respect to χ^2 distance, *i.e.* each partition has the same length, W. As shown in Fig. 2b, the distance between two consecutive bounds X_i and X_{i+1} of the line $\mathcal{R}^+$ is not constant if considering the distance $\forall i$, $l_2(X_i,X_{i+1}) \neq W$. However, the same interval is constant considering the distance χ^2 (Fig. 2a):

$$\forall i \chi^2(X_i,X_{i+1}) = \sqrt{\frac{(X_i - X_{i+1})^2}{X_i + X_{i+1}}} = W \tag{5}$$

This partition in the sense of χ^2 distance ensures that when two vectors are close (at a distance less than W after mapping to $\mathcal{R}^+$), the probability of collision is higher.

We are looking for a sensitive function $h_{\mathbf{a}}$ such that:

$$h_{\mathbf{a}}(\mathbf{p}) = n \quad \text{iff } X_{n-1} \leq \mathbf{a}.\mathbf{p} < X_n \tag{6}$$

where the sequence $(X_n)_n$ satisfies Eq. (5) with initial value set to zero: $X_0 = 0$. $\mathbf{a}$ is a random vector whose each entry is chosen independently from a Gaussian distribution with positive value $\mathcal{N}^+(0,1[$.

Eq. (5) leads to the relation between X_n and X_{n-1}:

$$X_n = X_{n-1} + W^2 \frac{\sqrt{8X_{n-1}/W^2 + 1} + 1}{2} \tag{7}$$

By fixing $X_0 = 0$, we obtain $X_n = (n(n+1)/2)W^2$.

Let us first introduce the following function

$$y : \mathcal{R}^+ \to \mathcal{R}^+$$

$$x \mapsto \frac{\sqrt{\frac{8x}{W^2} + 1} - 1}{2} \tag{8}$$

Proposition P1. *For a function h defined by Eq.* (6)*, a function y defined by Eq.* (8) *and adding an offset b on the projected line to avoid boundary effect, we have*

$$h_{\mathbf{a},b}(\mathbf{p}) = \lfloor y(\mathbf{a}.\mathbf{p}) + b \rfloor \tag{9}$$

with $\mathbf{p} \in \mathcal{R}^{+d}$, $\mathbf{a}$ *a random vector whose each entry is chosen independently from a Gaussian distribution with positive value* $\mathcal{N}^+(0,1)$ *and b a real number chosen uniformly at random in the range* $[0,1]$. *Let* $\mathcal{H}$ *the family of function defined by the set of function h.*

In the next part, we may consider $b = 0$ without loss of generality. To prove *P1*, we first define the sequence $(Y_n)_n$:

$$Y_n = y(X_n) : \frac{\sqrt{8\frac{X_n}{W^2} + 1} - 1}{2} \tag{10}$$

We prove for all n, (H_n):$Y_n = n$.

Proof by induction on *n*. As setup (H_0) is true: $Y_0 = 0$. Assuming (H_{n-1}) true, $Y_{n-1} = n-1$, from Eq. (10), we have: $Y_n^2 + Y_n = 2X_n/W^2$ with Eq. (5), we obtain

$$Y_n^2 + Y_n = 2\frac{X_{n-1}}{W^2} + 1 + \sqrt{8\frac{X_{n-1}}{W^2} + 1} \tag{11}$$

from (H_{n-1}) and Eq. (10) we deduce $2X_{n-1}/W^2 = n(n-1)$ and then replaced in Eq. (11), we obtain $Y_n(Y_n+1) = n(n+1)$. The only positive solution of this equation is $Y_n = n$.

Then, it is straightforward to see that $h_{\mathbf{a},b}(\mathbf{p}) = n$ in the general case (using the strict monotony of the function y) ending to prove our proposition P1. □

It follows that with adding an offset b on the projected line to avoid boundary effect, we obtain the sensitive function:

$$h_{\mathbf{a},b}(\mathbf{p}) = \frac{\sqrt{\frac{8\mathbf{a}.\mathbf{p}}{W^2} + 1} - 1}{2} + b \tag{12}$$

with $\mathbf{p} \in \mathcal{R}^{+d}$, $\mathbf{a}$ a random vector whose each entry is chosen independently from a Gaussian distribution with positive value $\mathcal{N}^+(0,1)$ and b a real number chosen uniformly at random in the range $[0,1[$.

5.4. Multi-probe LSH

The main drawback of LSH is the memory cost. Indeed, each hash table must be stored in main memory. Moreover, a large number of hash tables is required to reach high accuracy.

Lv et al. in [29] propose a new indexing scheme for l_2-metric called Multi-Probe LSH (MPLSH) that overcomes this drawback. MPLSH is based on E2LSH principle. Indeed, this approach also uses hash functions to map vectors into buckets and the pre-process (hash table construction) is therefore identical. However, the exploration stage is quite different. Instead of exploring only one bucket by hash table, success probabilities are computed for several buckets and buckets which are most likely to hold k-NN vectors are examined. Given the property of locality sensitive hashing, we know that if an object is close to a query object but not hash in the same bucket, it is likely to be in a neighboring bucket. The authors defined a *hash perturbation vector* $\Delta = (\delta_1, \ldots, \delta_M)$ where $\delta_i \in \{-1,0,1\}$. For a query q and a hash function $g(x) = (h_1(x), \ldots, h_M(x))$, the success probabilities for $g(q) + \Delta$ are computed and the T most likely buckets of each hash tables are visited. As a result, the authors reduce the number of hash tables by a factor of 14 to 18 for similar search accuracy and query time than *E2LSH*.

Since our new χ^2-based *LSH* scheme respects the same algorithmic steps as *E2LSH*, it is fully compliant with the MPLSH framework originally defined for *E2LSH* algorithm.

The four parameters chosen for this algorithm are the size of the search windows W, the number of projections M, the number of hash tables L and the number of probe T.

6. Experiments

Our experiments aims to prove that our active learning scheme applied to a subset of relevant images and based on χ^2 distance, named SALSAS, is as accurate as the state-of-the-art approach (Tong method with angle diversity [30]) while drastically decreasing the computational complexity of image retrieval task.

Experiments are reported as follows:

1. we first prove the efficiency of SALSAS,
2. we show the interest of using χ^2 distance,
3. we validate the usefulness of our LSH optimization for χ^2 distance, and
4. we show influence of parametrization.

To evaluate accuracy and efficiency of our method, experimental comparisons were performed with Tong's approach combined with angle diversity [26,3] on several databases from 5K to 180K images for the sake of scalability analysis. As this method has a linear complexity *w.r.t.* the size of the database, we name it linear method (LIN) in our experiments. Efficiency was only evaluate against linear method because existing proposals to speedup active learning process either only consider one of the scalability problems such as in [24,20,18], or use indexing structure known to be ineffective for high-dimensional input space, as in [21]. To the best of our knowledge, only Panda et al. [22] address both scalability issues of *sampling selection* and *database ranking*. However, the very brief description of the clustering stage to boost the sampling selection does not enable reimplementation.

6.1. Databases

We first perform evaluation on the VOC06 database [31] which contains 5,304 images for 10 categories. The use of a ground truth is required because user interactions are simulated during the active learning sessions: the membership of images to ground truth classes must be known by the "virtual user" in order to simulate reliable feedback. Moreover, ground truth is also required to evaluate the search quality.

In order to measure the evolution of the complexity of our algorithm *w.r.t* the size of the database, we have increased the size of the database by incorporating new images. It allows us to estimate the search time of our algorithm in very huge database (between million and thousand million of images).

Experiments have thus been performed on a 5 K, 20 K, 60 K, 100 K and 180 K images databases. The 20 database has been built by merging VOC2007 and VOC2008 databases. The ground truth has been obtained by merging classes of VOC databases. The three other databases are, respectively, built by adding key-frames of TrecVid 2007, 2008 and 2009 databases. We have considered all these images as irrelevant (not belonging to any of the 10 VOC classes considered).

In all of our experiments, results were provided by averaging 100 runs per category. Experiments were first performed on VOC06 and each run was initialized by a query image picked at random in the considered class. Then, experiments were performed on the other databases with the exact same query images as those for VOC06 experiments.

Each image is represented by a high dimensional histogram of 128 bins obtained by the concatenation of two histograms: one of 64 chrominance from CIE L∗a∗b and one of 64 textures from Gabor filters.

6.2. Evaluation protocol

To evaluate the accuracy, we compare the Mean Average Precision of TOPN images retrieved by the linear method with the one reached by our fast scheme. The efficiency of our method is evaluated by measuring how much our scheme is faster than the linear method.

The Mean Average Precision of TOPN is computed as follows: for each query image, we evaluate the average precision of the TOPN (AP_N). This value is the average of the precision value obtained after the N first images are retrieved by the system. Let $R^N = \{r_1, r_2, \ldots, r_N\}$ be a ranked version of the answer set. At any given rank j, let $|C \cap R^j|$ be the number of relevant images in the top j of R^N, where C is the total number of relevant images in the whole database $\mathcal{B}$. Then AP_N is defined as

$$AP_N = \frac{1}{N}\sum_{j=1}^{N}\frac{|R^j \cap C|}{j}\Delta(r_j) \tag{13}$$

where $\Delta(r_j) = 1$ if $r_j \in C$ and 0 otherwise. We first compute the mean value over the set of queries for each class and we take the mean value over the 10 classes to compute the MAP_N. N is a parameter that must be chosen by the user as a function of the number of images that he wants to retrieve. In our experiments, we fixed $N = 200$ by considering that a user is never interested in more than 200 images. A lower value of N would be an advantage for our method because it would decrease the accuracy difference with the linear method while emphasizing on our method speed.

Experiments are performed on a machine with a 3.2 GHz processor and 8 GB of memory.

6.3. Parametrization

We now detail the parametrization of the classical active learning strategy (called linear method in our experiments) and our fast scheme. The parameters have been fixed by leading preliminary tests on the VOC2006 database without using any annotation. We begin with presenting the parameter settings common to the linear method and to our fast scheme, then we detail the parameters specific to our fast scheme.

Active learning strategy with relevant feedback requires to fix as parameters:

- the batch size b (number of labels by iteration) set at 1 (by default) and
- the similarity kernel function.

We test two RBF-kernels ($K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-d(\mathbf{x}_i,\mathbf{x}_j)^2/2\sigma^2)$): the classical l_2-RBF kernel often employed by default, where $d(\mathbf{x}_i,\mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_2$ and the χ^2-RBF kernel, where $d(\mathbf{x}_i,\mathbf{x}_j) = \chi^2(\mathbf{x}_i,\mathbf{x}_j)$. SALSAS is defined with the second kernel. We set σ values with the following heuristics preformed on the VOC2006 database:

- we compute the central vector $\mathbf{x}_c$ of the input space,
- we compute the mean distance d_m between $\mathbf{x}_c$ and all vectors of the database, and
- we set σ so that d_m is equal to the half of the maximal value of K: $\sigma = d_m/2\sqrt{(2\log(2))} \approx d_m/2.35$.

We can notice that a good estimation of $\mathbf{x}_c$ and d_m can be given with a subsample of the database. We obtain $\sigma = 213$ for the χ^2 distance

and $\sigma = 97\,503$ for the l_2 distance. σ values have been rounded to $\sigma = 200$ for the χ^2-RBF kernel and $\sigma = 10^5$ for l_2-RBF kernel. This heuristic is preferred, in interactive retrieval context, to cross-validation because of the very small size of the training set. Moreover, as the training set is iteratively built for each search, it is difficult to identify it from preliminary test.

Our fast scheme needs two more parameters: the pool size p (by default $p=N=200$) and the number of k-NN to update the pool, k is set at $p/2$.

We must then parametrize the index structure LSH to perform k-NN search. The main parameter is the bin width W. W is the minimal radius containing the k-NN images of all query images. As it is impossible to compute this radius for all image of the database, we use a statistical method to estimate this search radius.

From the theorem 6.33 (ranks on random subsets) in [32, Section 6.5]: Denote by $M := \{x_1, \ldots, x_m\} \subset \mathbb{R}$ a set of m elements, and $\tilde{M} \subset M$ a random subset of size $\tilde{m}$. Then the probability that the best element of $\tilde{M}$ is better than n elements of M is at least $1-((m-n)/m)^{\tilde{m}}$.

It is then sufficient to consider $\tilde{M}$ by randomly selecting $\tilde{m}$ elements of M to ensure with a probability η that the best element of $\tilde{M}$ is better than the nth best element of M:

$$\tilde{m} = \frac{\log(1-\eta)}{\log(\frac{m-n}{m})} \tag{14}$$

On the dataset VOC2006, $m=5304$, then it is sufficient to randomly select $\tilde{m} = 158$ images to ensure with a probability of 95% that the best elements of the random subset belongs to 100-NN of the dataset. We randomly select 1000 queries. For each query, we keep the minimal distance obtained from 158 randomly selected images. We then sort distances and take the 95th value. We obtain a distance of 387 for χ^2 distance and 173,520 for l_2 distance. We round this value to fix $W=400$ for χ^2 and $W = 1.75 \times 10^5$ for l_2.

For the three others LSH parameters, as in [29], we fix $L=4$ hash tables, $M=24$ projections by hash functions and $T=100$ probes.

6.4. Evaluation of the proposed scheme

For all the experiments, one iteration represents one relevance feedback loop.

We first evaluate our method SALSAS comparing it with the linear χ^2-RBF kernel. Results on VOC2006 database are reported in Fig. 3. As we can see in Fig. 3a, our method provides better results for the 20 first iterations and is slightly less efficient for the next iterations than exhaustive search (LIN_CHI2). At the 50th iteration, our method reaches an accuracy of 46.06 that is less than 3% worse than linear scheme which reaches 48.84%. This slight deterioration of the ranking is counter-balanced by the speedup of our algorithm that is more than twice faster on this database of only 5K images.

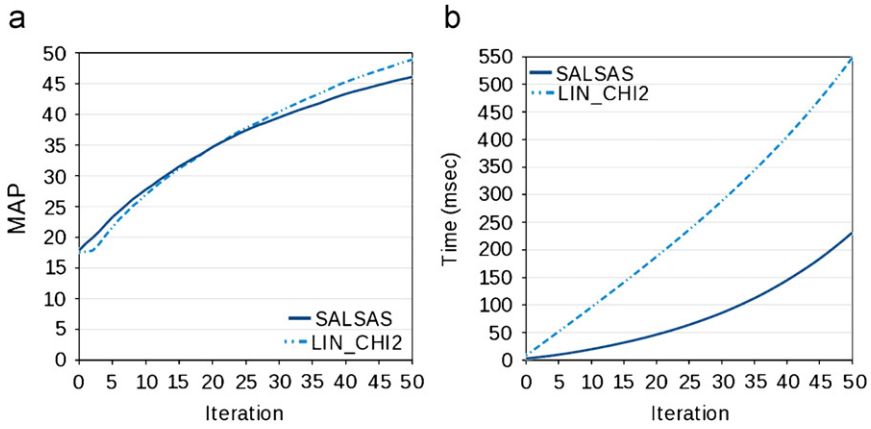


Fig. 3. Evaluation of the accuracy and the efficiency on VOC2006 vs the number of iterations: (a) MAP of TOP200 vs number of iterations on VOC06 and (b) time vs number of iterations on VOC06.

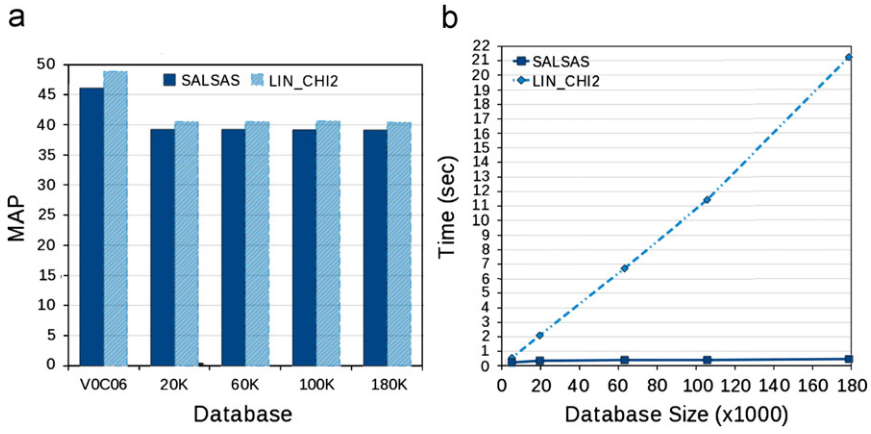


Fig. 4. Evolution of the accuracy and the efficiency with the size of the database for 50 iterations with 1 label by iteration: (a) MAP of TOP200 at 50th iteration vs database size and (b) time at 50th iteration vs database size.

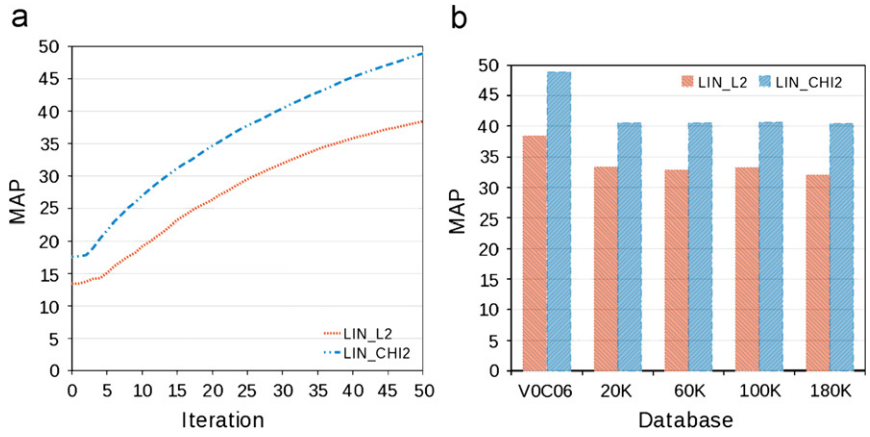


Fig. 5. Accuracy of linear method with χ^2-RBF kernel and l_2-RBF kernel: (a) MAP of TOP200 vs number of iterations on VOC06 and (b) MAP of TOP200 at 50th iteration vs database size.

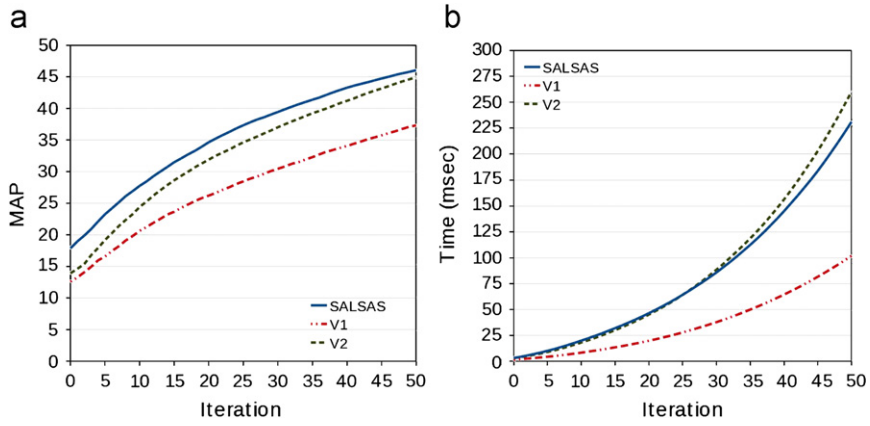


Fig. 6. Comparison of the accuracy and the efficiency on VOC2006 vs the number of iterations with several implementations of the fast scheme: (a) MAP of TOP200 vs number of iterations on VOC06 and (b) time vs number of iterations on VOC06.

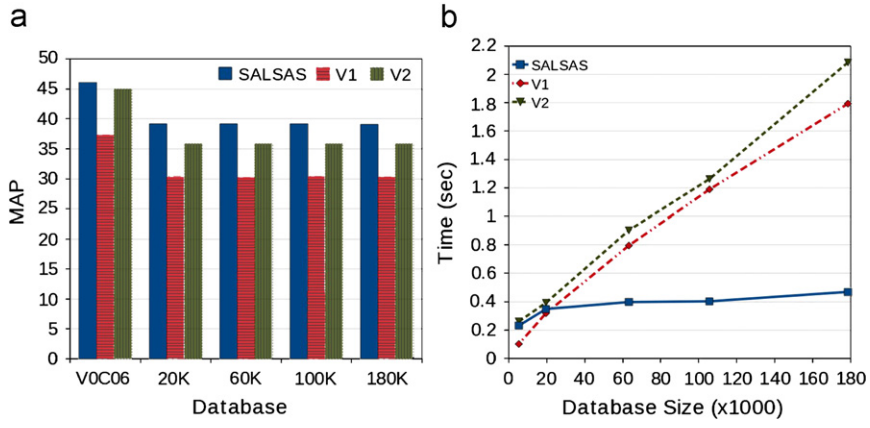


Fig. 7. Evolution of the accuracy and the efficiency with the size of the database for 50 iterations with 1 label by iteration: (a) MAP of TOP200 at 50th iteration vs database size and (b) time at 50th iteration vs database size.

Indeed, as shown in Fig. 3b, for a search of 50 iterations, our algorithm takes 230 ms against 550 for the linear method.

The second evaluation aims at measuring the evolution of search time and the precision of our method when the database size increases. As shown in Fig. 4a, the accuracy of the ranking decreases for the two methods between VOC06 and the 20 K database. The accuracy decreases because we added new images more difficult to identify in each classes which complicates the search process. It may be noticed that the accuracy of our method decreases less than that of linear method. Indeed, the gap between the two methods is only of 1.3%. For bigger databases, obtained by adding TrecVid images without changing classes, the precision does not decrease. We have shown that the size of the pool $\mathcal{S}$ can remain constant even if the number of images in the database increases highly. Moreover, the increase of search duration is small when the database grows (Fig. 4b). Indeed, for the database of 180 K

images, our method takes only 0.47 s against 21.21 s for the linear method. Then, it only takes twice as long time (0.23 s for 5 K images) to search in a more than 33 times bigger database. Our methods is then 45 times faster than the linear method for 180 K images. Furthermore, curves in Fig. 4b show that this gap should increase for larger databases.

We have then shown that SALSAS is able to drastically decrease the computational time of active learning while keeping similar accuracy for RBF χ^2 kernel.

6.5. Linear χ^2-RBF *vs* linear l_2-RBF

This experiment illustrates the interest of using RBF with χ^2 distance over using classical RBF kernel with l_2 distance. As shown in Fig. 5, χ^2 distance reaches better results on all databases and for each relevant feedback iterations. At the end of the search (50th iterations), the gap between these two kernels is between 7% and 10%.

6.6. Comparison of indexing structure block

In this set of experiments, we show the contribution of our indexing structure optimized for χ^2 distances. We compare SALSAS with two other implementations of our fast scheme. These two implementations, named V1 and V2, use *E2LSH* as indexing structure to perform k-NN search for l_2 distance. V1 is combined with a l_2-RBF kernel and V2 with a χ^2-RBF kernel.

Results are reported in Figs. 6 and 7. On Figs. 6a and 7a, we can see that V2 is more accurate than V1. The combination between k-NN search with l_2 distance for the selection stage and kernel on χ^2 distance is not absurd. However, SALSAS provide better accuracy than V2. Our optimization of χ^2 distance for the k-NN search is more relevant. At the 50th iteration, the gap between these two implementations is more than 3% for the four bigger databases. Indeed SALSAS reaches a MAP of 39.2% against 35.8% for V2. Moreover, SALSAS is more efficient than V2. For the 180 K image database, the 50 iterations only take 0.47 s for SALSAS against 2.08 s for V2.

Table 1
MAP of TOP200 at the 50th iteration for each category for the 180 K database.

Categories	LIN_CHI2	SALSAS
Bicycle	44.38	45.28
Bus	30.33	29.74
Car	81.59	86.73
Cat	20.51	18.38
Cow	42.66	36.02
Dog	17.4	17.83
Gorse	19.87	20.74
Motorbike	31.12	30.36
Person	62.04	64.9
Sheep	54.34	40.98
Mean	40.42	39.1

6.7. Detailed analysis of results

In Table 1, we report the MAP for the 10 categories obtained on the 180 K database. For seven categories over 10, the accuracy of SALSAS and the linear method are very similar, the difference between MAP does not exceed 2%. However, for categories *car*, *cow* and *sheep* the differences are more than 5%. Our method reaches better accuracy for the category *car* than the linear method. However, for categories *cow* and *sheep*, our method does not allow to reach same ranking.

Our investigation to understand this difference highlighted that these two categories have a set of images strongly concentrated in the feature space and some images are scattered into the feature space. As illustrated in Fig. 8, when an isolated image is used to initialize the search, the active learning system takes several iterations before retrieving an image which allows to improve the ranking (10 iterations are required for the illustration of Fig. 8). During these iterations, the sampling stage only proposed irrelevant images and the active learning thus explores a big area of the space before finding an image close to the center of the class. Once this image is found, it allows to easily access to a larger number of images of the class. As a result, the MAP increases significantly. On the other hand, as our method does not update the pool of images $\mathcal{S}$ for negative labels the exploration is limited. As a result, our method does not allow to find the center of class when the system is initialized with a too difficult query image (representing a very

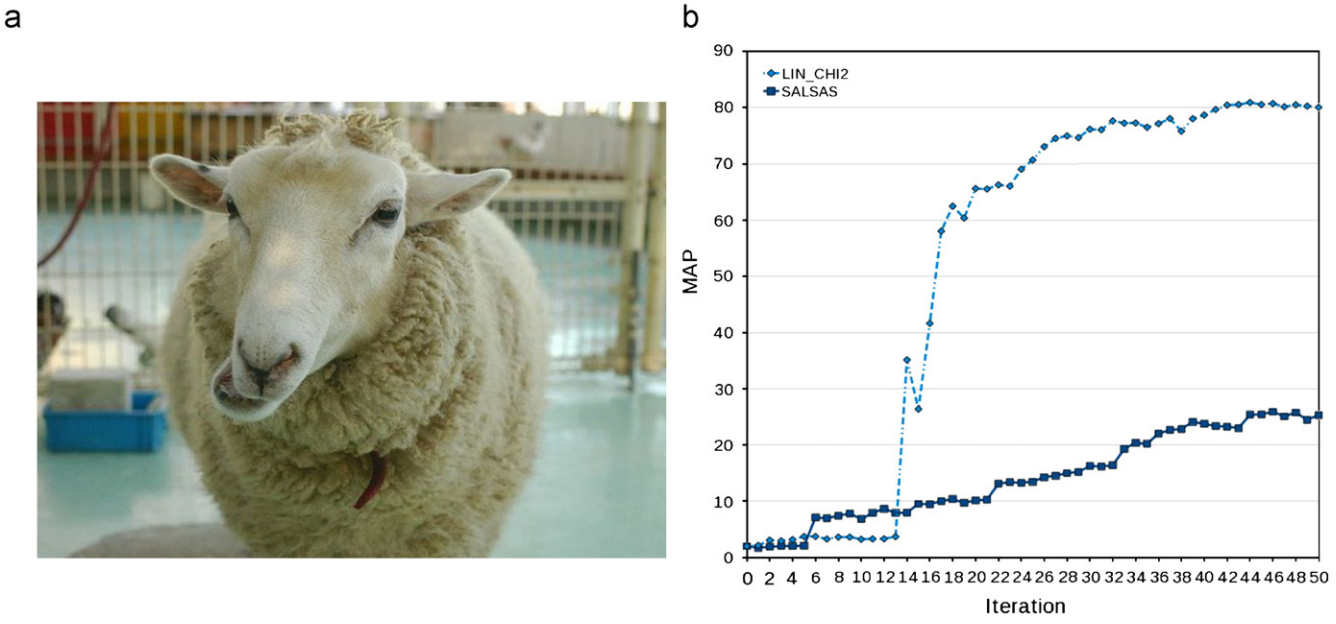


Fig. 8. Illustration of a disadvantageous case for our method: (a) query image and (b) MAP of TOP200 vs number of iterations on 180 K database.

specific modality of the targeted semantic concept) and the results are then deteriorated.

6.8. Parametrization impact

6.8.1. Iteration parametrization

In active learning, it may be interesting to perform multiple pool-queries at once. Indeed, by presenting one image at a time to the user, he or she is likely to loose patience after a few rounds. To avoid this, multiple images (say, b) can be presented at the same time for labeling. Moreover, the user does not necessarily hold a query image to present to the system to initialize the algorithm. In this case, the search can be started with a text search and returns several images. Then, the user labels theses images which are used to launch the active learning process. It may thus be interesting to test our system by starting the algorithm with several images (relevant and irrelevant).

Exhaustive experiments were performed by varying both the number of image requests and the number of images to annotate in order to evaluate our system. Results with five query images (three positives and two negatives) and five labels per feedback loop are reported in Figs. 10 and 9. As we can see in Figs. 9 and 10, the accuracy of SALSAS is quite similar to the linear method. Indeed, by increasing the number of images to initialize the system, the case illustrated in Fig. 8 is less probable to occur. Moreover, results prove that our system is able, for the sampling stage, to find in the pool $\mathcal{S}$ several images to be labeled as relevant, as the linear system does. Therefore, the updating process of the pool $\mathcal{S}$ is also relevant when several images are positively labeled.

As shown in Fig. 10, although the speedup of SALSAS goes down by increasing the batch size with five labels by iteration, our system is still 20 times faster than the linear method. Furthermore the complexity is still sub-linear in the size of the database.

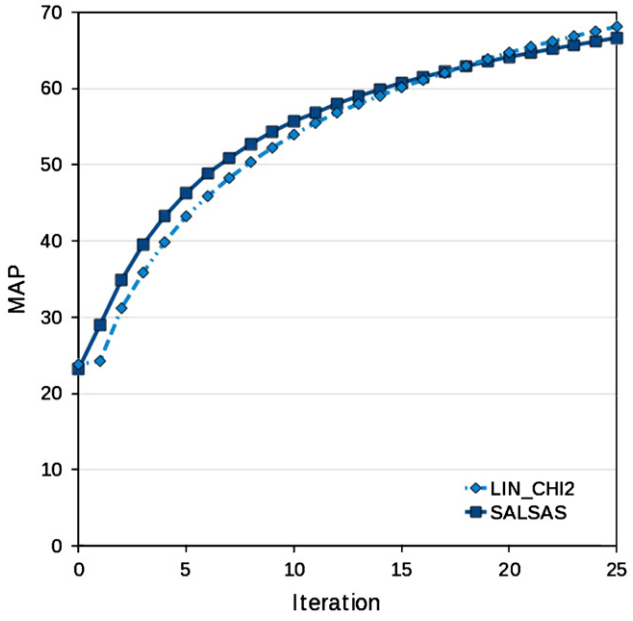


Fig. 9. Evolution of MAP of TOP200 vs the iterations on VOC06 database with five labels by iteration and five query images.

6.8.2. Pool size

In this set of experiments we evaluate the influence of the pool size. As we can see in Fig. 11a, multiplying by 2 the size of the pool does not allow to increase the accuracy of SALSAS. Moreover, the bigger the pool size, the slower our method is (Fig. 11b).

6.8.3. LSH parametrization

In this set of experiments, we evaluate the influence of the two main parameters of LSH, W and T, which allow to tune the trade-off between accuracy and efficiency.

We first control that $W=400$, previously fixed through preliminary tests and heuristics on the database VOC2006, is still valid on the 180 K dataset. In other words, we evaluate the quality of our heuristics. Secondly, we verify that $T=100$, default value provided by [29], is also still valid on our dataset.

Impacts of the window size W and of the number of probe T are, respectively, reported in Figs. 12 and 13.

As shown in Fig. 12a, increasing W allows to improve the search quality until reaching a MAP of 39.1% for $W=400$. Our heuristic based on preliminary tests on VOC2006 database is thus still valid on 180 K database. This parameter allows to increase the probability of finding the exact nearest neighbors of each query images. However, the higher the precision, the longer the search time. A trade-off between accuracy and efficiency has to be found.

As shown in Fig. 13, the number of probes T is the second parameter which allows to tune the trade-off between accuracy and efficiency. The parameter provided in [29] is tuned in order to favor accuracy.

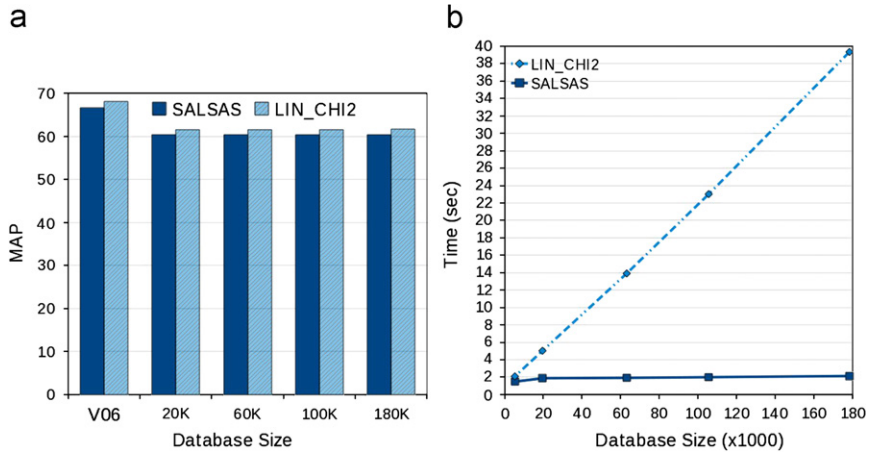


Fig. 10. Evolution of accuracy and efficiency with the size of the database for 25 iterations of five labels and five query images: (a) MAP of TOP200 at 25th iteration vs database size and (b) time at 25th iteration vs database size.

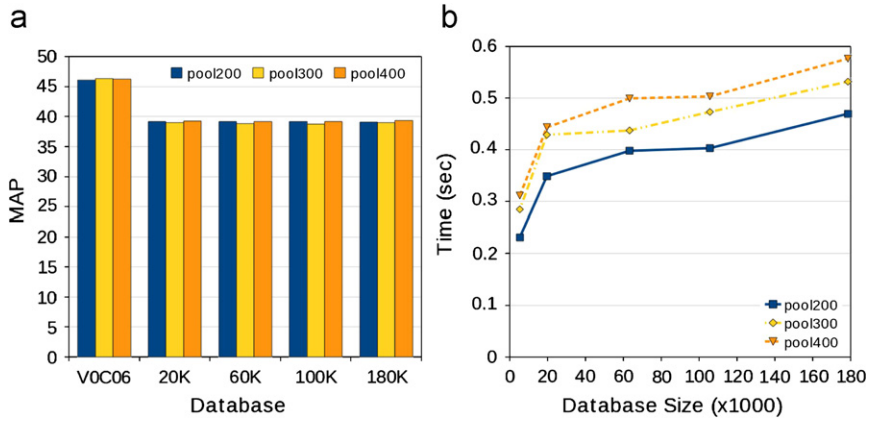


Fig. 11. Influence of pool size for 50 iterations with one label by iteration: (a) MAP of TOP200 at 50th iteration vs database size and (b) time at 50th iteration vs database size.

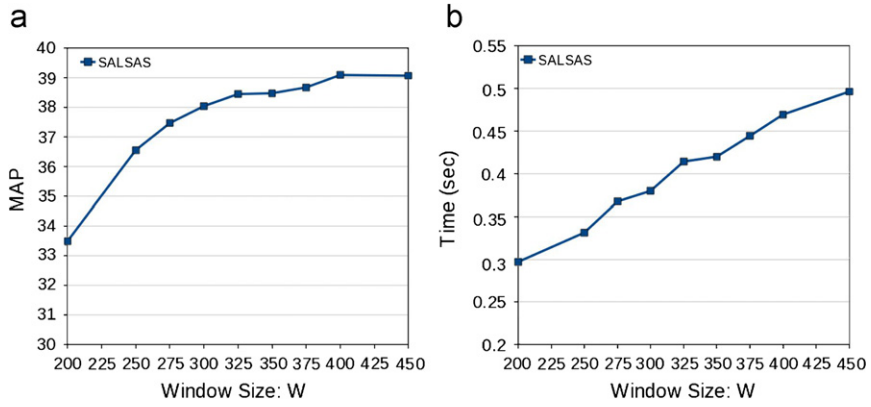


Fig. 12. Impact of LSH *W* parameter (window size) for 50 iterations of one label by iteration on 180 K database: (a) MAP of TOP200 at 50th iteration vs W and (b) time at 50th iteration vs W.

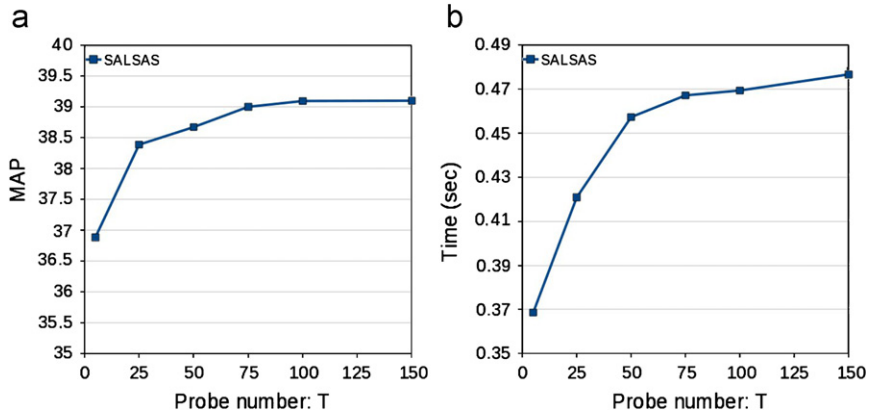


Fig. 13. Impact of MPLSH *T* parameter (number of probes) for 50 iterations of one label by iteration on 180 K database: (a) MAP of TOP200 at 50th iteration vs T and (b) time at 50th iteration vs T.

6.9. Evaluation summary

Our experiments have allowed us to prove that:

- our method provides similar accuracy as Tong method combined with angle diversity [30],
- our method allows to drastically reduce the computational complexity of active learning in large databases,
- χ^2 is a relevant distance for image retrieval task,
- our hash function based on χ^2 distance is accurate and effective.
- SALSAS is still efficient with multiple pool-queries.

7. Fast RETIN search engine

We have developed a CBIR system named RETIN dealing with dictionary-based approaches and on-line statistical learning strategies

[25,33]. We integrate, in RETIN, our fast scheme presented in this paper in order to deal with very large databases. We present an example of retrieval session in Table 2 on the 180 K database, providing the same query, for both our approximate method and the exact search to illustrate how "comparable" the results are.

8. Conclusion

Nowadays, providing the user with a system for searching into very large image databases becomes a critical issue of Content-Based Image Retrieval systems (CBIR). However, bridging the *semantic gap* between which (semantic) concept(s) the user is looking for and the (semantic) content of this unique image is quite difficult. Active learning has proved to be particularly relevant in interactive image retrieval. The image selection for annotation and the ranking of the database are the two key aspects for scalability of active learning strategies of retrieval systems. In this paper, we have proposed a strategy to tackle these scalability issues. Based on LSH strategy we quickly select a pool of relevant images to speedup the sampling and the ranking. A strategy is proposed to quickly update this pool at each feedback iteration. We also design a new LSH scheme on χ^2 distance that allows to use more accurate kernels. Furthermore, taking benefits from this scheme, our method proves to be theoretically sub-linear. Experimental results on a huge database show that our algorithm achieves same accuracy than the reference methods, Tong approach with angle diversity strategy, while dividing the computational complexity by 45. Where on-line search, for 50 iterations, in a 180 K image database takes 21 s, our method takes 0.47 s. This gain will become crucial for the 10 million database we are working on. One of the next step will be to investigate new strategies to explore faster and more efficiently the feature space in order to update the pool of considered relevant data.

Table 2
Comparisons of search sessions between LIN_CHI2 and SALSAS initialized with a bike on the graphical interface of our system. Top part: retrieved images; bottom part: images selected by the active learner; green square: image labeled positively; red square: image labeled negatively.

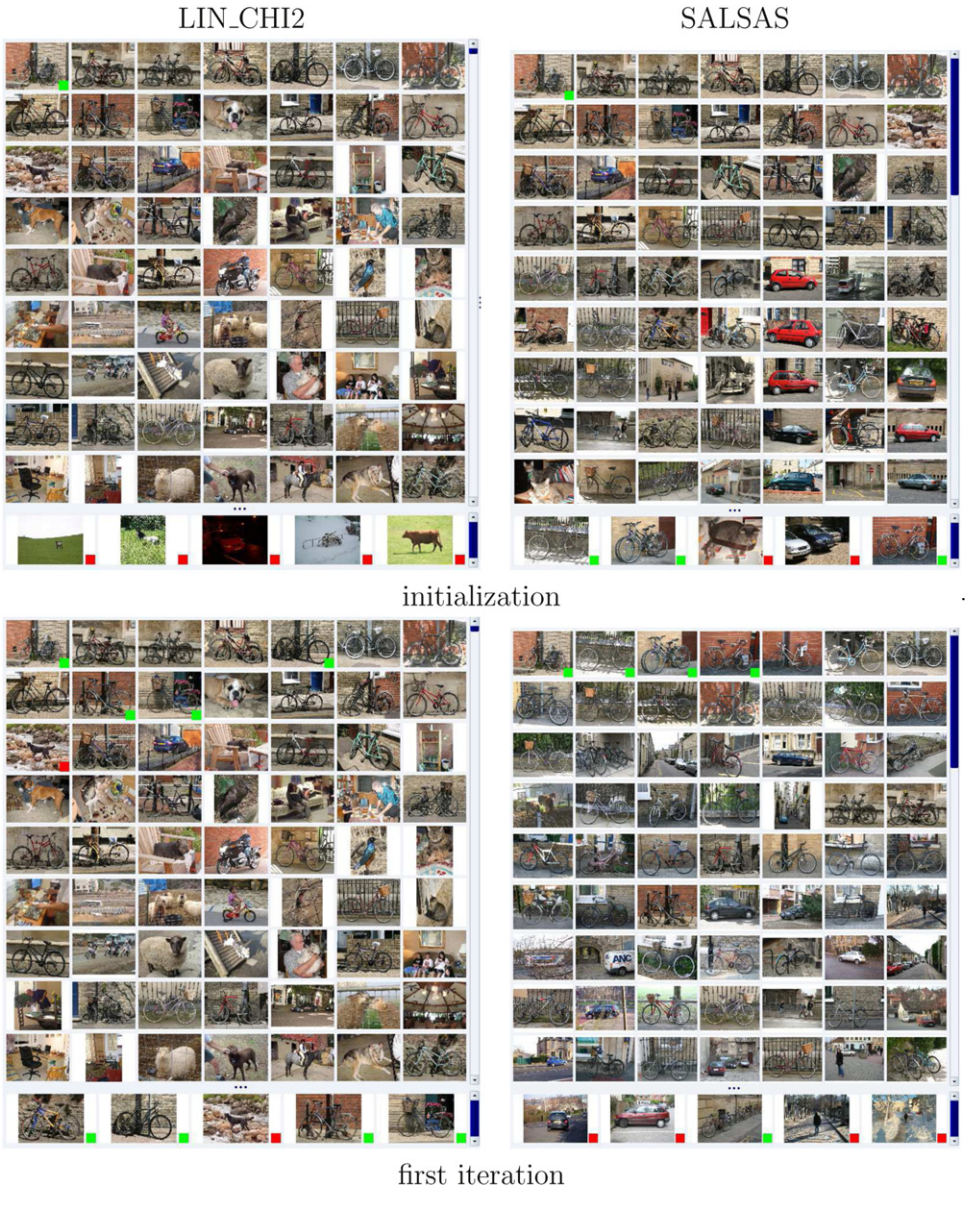
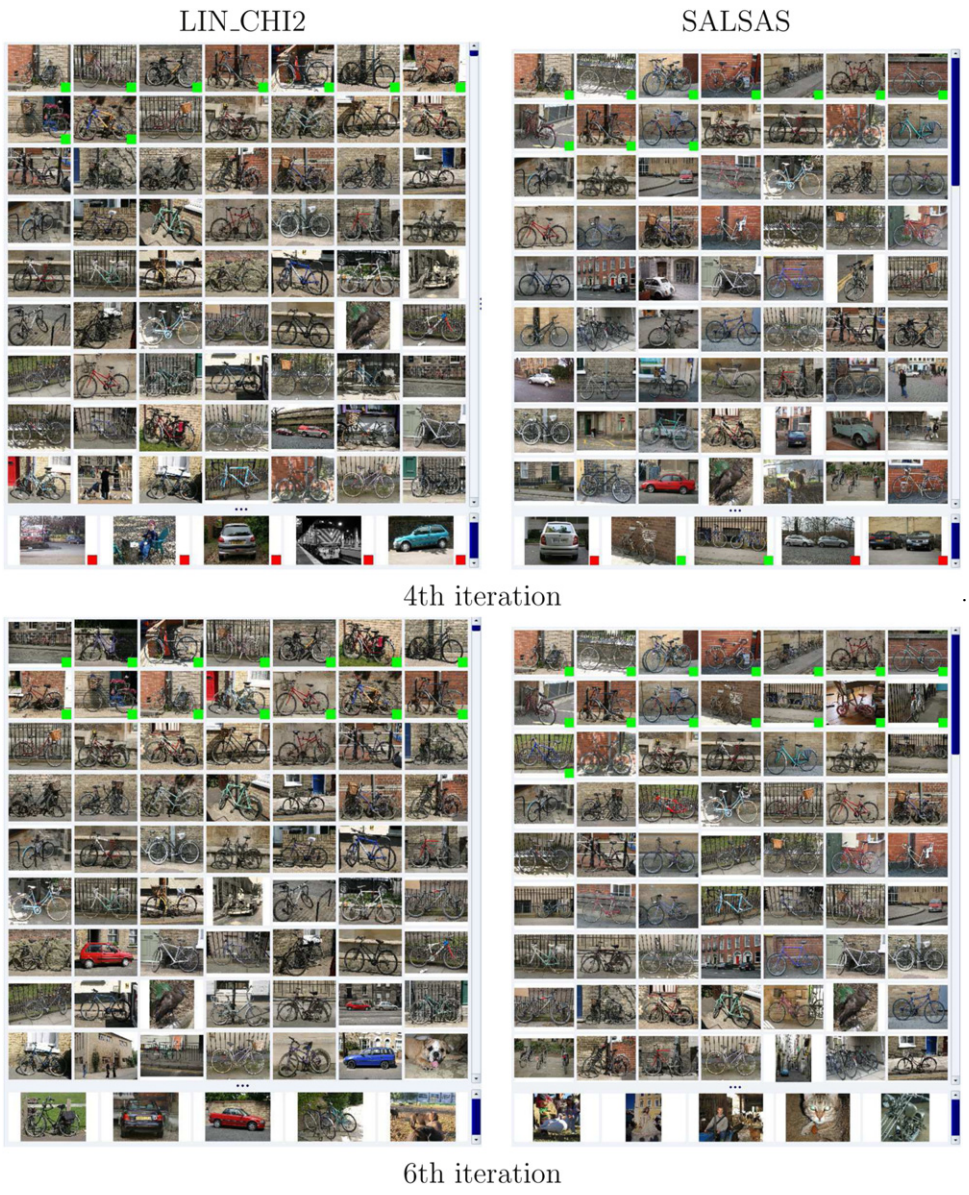


Table 2. (*continued*)



References

[1] J. Cheng, K. Wang, Active learning for image retrieval with CO-SVM, Pattern Recognition 40 (2007) 330–334.

[2] T. Huang, C. Dagli, S. Rajaram, E. Chang, M. Mandel, G. Poliner, D. Ellis, Active learning for interactive multimedia retrieval, Proceedings of the IEEE 96 (4) (2008) 648.

[3] E. Chang, S. Tong, K. Goh, C. Chang, Support vector machine concept-dependent active learning for image retrieval, IEEE Transactions on Multimedia 2 (2005).

[4] S.C. Hoi, R. Jin, J. Zhu, M.R. Lyu, Semi-supervised SVM batch mode active learning for image retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–7.

[5] Y. Rui, T. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Transactions on Circuits and Systems for Video Technology 8 (5) (1998) 644–655.

[6] X. Zhou, T. Huang, Relevance feedback in image retrieval: a comprehensive review, Multimedia Systems 8 (6) (2003) 536–544.

[7] E. Chang, B. Li, G. Wu, K. Goh, Statistical learning for effective visual information retrieval, in: IEEE International Conference on Image Processing, 2003, pp. 609–612.

[8] Z.-H. Zhou, K.-J. Chen, H.-B. Dai, Enhancing relevance feedback in image retrieval using unlabeled data, ACM Transactions on Information System 24 (2) (2006) 219–244 doi:http://doi.acm.org/10.1145/1148020.1148023.

[9] K. Yu, J. Bi, V. Tresp, Active learning via transductive experimental design, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 1081–1088.

[10] D. Gorisse, M. Cord, F. Precioso, Optimization on active learning strategy for object category retrieval, in: IEEE International Conference on Image Processing, IEEE, 2009.

[11] O. Chapelle, P. Haffner, V. Vapnik, Support vector machines for histogram-based image classification, IEEE Transactions Neural Networks (1999) 1055–1064.

[12] V. Vapnik, Statistical Learning Theory, Wiley-Interscience, New York, 1998.

[13] N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 441–448.

[14] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, The Journal of Machine Learning Research 2 (2002) 45–66.

[15] A. Bordes, S. Ertekin, J. Weston, L. Bottou, Fast kernel classifiers with online and active learning, Journal of Machine Learning Research 6 (2005) 1579–1619.

[16] L. Wang, X. Li, P. Xue, K. Chan, A novel framework for SVM-based image retrieval on large databases, in: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM, New York, NY, USA, 2005, pp. 487–490.

[17] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, pp. 518–529.

[18] J. Peng, D.R. Heisterkamp, Kernel indexing for relevance feedback image retrieval, in: IEEE International Conference on Image Processing, 2003.
[19] D.R. Heisterkamp, J. Peng, Kernel va-files for relevance feedback retrieval, in: ACM International Workshop on Multimedia Databases, 2003, pp. 48–54.
[20] D.R. Heisterkamp, J. Peng, Kernel vector approximation files for relevance feedback retrieval in large image databases, Multimedia Tools Application 26 (2) (2005) 175–189, doi:http://dx.doi.org/10.1007/s11042-005-0454-4.
[21] N. Panda, E.Y. Chang, Efficient top-k hyperplane query processing for multimedia information retrieval, in: Proceedings of the 14th Annual ACM International Conference on Multimedia, 2006, pp. 317–326.
[22] N. Panda, K.-S. Goh, E.Y. Chang, Active learning in very large databases, Multimedia Tools and Applications 31 (3) (2006) 249–267.
[23] N. Panda, E.Y. Chang, Exploiting geometry for support vector machine indexing, in: Proceedings of SIAM International Conference on Data Mining, 2005.
[24] M. Crucianu, D. Estevez, V. Oria, J.-P. Tarel, Hyperplane queries in a feature-space m-tree for speeding up active learning, in: Proceedings of Journées Bases de Données Avancées, 2007, pp. 1–2.
[25] P. Gosselin, M. Cord, S. Philipp-Foliguet, Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval, Computer Vision and Image Understanding 110 (3) (2008) 403–417.
[26] K. Brinker, Incorporating diversity in active learning with support vector machines, Machine Learning International Workshop then Conference, 2003, pp. 59–66.
[27] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: Proceedings of the 13th Annual ACM Symposium on Theory of Computing, 1998, pp. 604–613.
[28] M. Datar, N. Immorlica, P. Indyk, V. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the 20th Annual Symposium on Computational Geometry, 2004, pp. 253–262.
[29] Q. Lv, W. Josephson, Z. Wang, M. Charikar, K. Li, Multi-probe LSH: efficient indexing for high-dimensional similarity search, in: Proceedings of the International Conference on Very Large Data Bases, 2007, pp. 950–961.
[30] S. Tong, E. Chang, Support vector machine active learning for image retrieval, in: Proceedings of the Ninth ACM International Conference on Multimedia, 2001, pp. 107–118.
[31] M. Everingham, A. Zisserman, C.K.I. Williams, L. Van Gool, Pascal voc2006, 2006.
[32] B. Scholkopf, A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, 2002.
[33] P. Gosselin, M. Cord, S. Philipp-Foliguet, Active learning methods for interactive image retrieval, IEEE Transactions on Image Processing 17 (2008) 1200–1211.

David Gorisse received the M.S. degrees in Computer Science from the University of Cergy-Pontoise, France, in 2007 and the M.Sc. in Electrical Engineering and Telecommunications by ISEN, France, in 2006.

Currently he is a Ph.D. Student in Computer Sciences by the University of Cergy-Pontoise at ETIS joint laboratory of CNRS/ENSEA/Univ Cergy-Pontoise, France. His research interests include Computer Vision, Machine Learning, Kernel Design for Multimedia Information Retrieval and Recognition.

Matthieu Cord obtained his Ph.D. degree in Image Processing in 1998 by the University of Cergy-Pontoise, France, and was a post-doc in 1999 at the Katholieke Universiteit Leuven, Belgium. Then, he joined the ETIS labs in France to create the image indexing research group. In 2006, he joined the UPMC-Paris 6 University, where he got a full professor position. He is involved in several French and international research programs and projects and has been recently nominated to the prestigious French Research Institute (IUF) for 5 years. His research interests include Computer Vision, Image Processing, Machine Learning and their applications to Multimedia Information Retrieval and Multimedia Processing.

Frédéric Precioso has a Ph.D. in Signal and Image Processing, obtained from University of Nice-Sophia Antipolis,France, in 2004. After a year of Post-Doctorate at CERTH-Informatics and Telematics Institute, Thessaloniki, Greece, where he worked on semantic methods for object extraction and retrieval, he became Associate professor at ENSEA since 2005. He is involved in several French and international research programs. He used to work on video and image segmentation, active contours and his current main topics of interest concern video object detection and classification, content-based video indexing and retrieval systems, scalability of such systems.