Information Sciences 000 (2015) 1–15

Contents lists available at ScienceDirect

Information Sciences

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

Improving bag-of-visual-words image retrieval with predictive clustering trees

Ivica Dimitrovski[a,*], Dragi Kocev[b], Suzana Loskovska[a], Sašo Džeroski[b]

[a] Faculty of Computer Science and Engineering, University Ss. Cyril and Methodius, Skopje, Macedonia
[b] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 12 April 2014
Revised 22 March 2015
Accepted 1 May 2015
Available online xxx

Keywords:
Image retrieval
Feature extraction
Visual codebook
Predictive clustering

ABSTRACT

The recent overwhelming increase in the amount of available visual information, especially digital images, has brought up a pressing need to develop efficient and accurate systems for image retrieval. State-of-the-art systems for image retrieval use the bag-of-visual-words representation of images. However, the computational bottleneck in all such systems is the construction of the visual codebook, i.e., obtaining the visual words. This is typically performed by clustering hundreds of thousands or millions of local descriptors, where the resulting clusters correspond to visual words. Each image is then represented by a histogram of the distribution of its local descriptors across the codebook. The major issue in retrieval systems is that by increasing the sizes of the image databases, the number of local descriptors to be clustered increases rapidly: Thus, using conventional clustering techniques is infeasible. Considering this, we propose to construct the visual codebook by using predictive clustering trees (PCTs), which can be constructed and executed efficiently and have good predictive performance. Moreover, to increase the stability of the model, we propose to use random forests of predictive clustering trees. We create a random forest of PCTs that represents both the codebook and the indexing structure. We evaluate the proposed improvement of the bag-of-visual-words approach on three reference datasets and two additional datasets of 100 K images and 1 M images, compare it to two state-of-the-art methods based on approximate *k*-means and extremely randomized tree ensembles. The results reveal that the proposed method produces a visual codebook with superior discriminative power and thus better retrieval performance while maintaining excellent computational efficiency.

1. Introduction

The amount of visual information becoming available in various digital archives is in constant growth. For instance, the widely used social web sites such as Facebook[1] and Flickr[2] store several billion images for their users. The improvement of digital cameras and user interfaces for upload of images will further increase the amount of available images. The value of the

* Corresponding author.
E-mail addresses: Ivica.Dimitrovski@finki.ukim.mk (I. Dimitrovski), Dragi.Kocev@ijs.si (D. Kocev), suzana.loshkovska@finki.ukim.mk (S. Loskovska), Saso.Dzeroski@ijs.si (S. Džeroski).

[1] Facebook©– http://www.facebook.com.
[2] Flickr from Yahoo!©– http://www.flickr.com.

http://dx.doi.org/10.1016/j.ins.2015.05.012
0020-0255/

information obtained from an image depends on how easily it can be found, retrieved, accessed, filtered and managed. Considering this, the development of systems for efficient archiving, browsing and searching images is a necessity. Such systems are being developed within the research area of image retrieval.

Image retrieval is an inter-disciplinary research area that cross-fertilizes the following research areas: multimedia research, information retrieval, machine learning, computer vision, and human–computer interaction. The methods for image retrieval can be categorized into two categories [16]: text-based image retrieval (TBIR) and content-based image retrieval (CBIR). The former group of methods require some meta-data for each image in textual format (i.e., image tags) which allow retrieval to be performed by providing textual queries. These methods perform well and are efficient as long as the images are correctly tagged. However, they have two serious limitations: the large amount of human labor required for manual annotation (which increases severely in the case of large image databases) and the inherent inaccuracy resulting from the subjectivity of human annotators. To alleviate these limitations, CBIR methods were introduced. They describe the images by their visual content, such as color, texture, shapes and local descriptors. The CBIR methods heavily rely on extracting appropriate image descriptors and a good similarity measure between images.

CBIR methods can be readily applied in several practically relevant domains [25]. To begin with, they can be used for the creation of a web-scale visual search engine where the query will be a visual object. Second, they can be used for searching through personal image databases (e.g., select the photos that contain an image of the Eiffel Tower). Next, performing product search will strongly benefit from such systems: The user can take a photo of a given product, perform a search on the web and compare its prices from the given store to the price in the on-line shops. Furthermore, these methods will facilitate the automatic tagging of images uploaded to social media, such as Flickr and Facebook. Finally, these methods can be used for creation of augmented reality and visual guides for museums and art galleries. For instance, a user can take a photo of a given sculpture and look for information about the given sculpture on the web.

Several systems for content-based image retrieval have been proposed in the literature [13,21,25,30]. These systems are inspired from the text retrieval systems and use the analogy between bag-of-words and bag-of-visual-words representations [29]. An architecture of such a system for image retrieval is depicted in Fig. 1. It consists of three phases: creation of visual codebook, image description and similarity definition. The creation of the visual vocabulary starts by detecting interesting points in the images. From these points, then, local invariant descriptors are extracted. Finally, the visual codebook is obtained by clustering the large set of descriptors obtained from all of the images. Each resulting cluster represents a visual word, while all the visual words comprise the visual dictionary. The image description phase consists of assigning all of the local image descriptors to the visual words from the visual dictionary. Each image is then described with a high-dimensional histogram, where each component of the histogram is the number of descriptors that are assigned to a given visual word. Finally, the images are ranked using term frequency inverse document frequency (*tf-idf*) scores which discount the influence of visual words which occur in many images. The search is then performed efficiently using a fast and tree-based inverted index structure [4].

The systems for content-based image retrieval face several challenges [25]. To begin with, the changes in lighting, image scale and rotation can hurt the performance of the retrieval systems. Second, viewpoint changes can make previously unseen parts of the object visible and also may include obstructions which will cover parts of the object. Finally, the systems need to be scalable with respect to the size of the image database, while preserving the retrieval accuracy. Moreover, the construction of the visual codebook should be also performed efficiently.

To address these issues, we propose an improvement of the bag-of-visual-words approach for efficient and simple image retrieval that does not only constructs a visual codebook, but also simultaneously constructs a complete indexing/search structure. The indexing structure can be readily used for indexing, ranking and retrieval of relevant images. More specifically, the proposed system is based on the predictive clustering framework [5] that unifies predictive modeling and clustering through methods that partition the instances into subsets, such as decision trees and decision rules. The task of predictive clustering is to identify clusters of instances that are close to each other both in the target and in the descriptive space.

In this work, we use predictive clustering trees (PCTs) to construct the indexing/search structure. In particular, we use PCTs for predicting multiple targets in which the descriptive attributes are also considered as clustering/target attributes. Using a single PCT is a fast and efficient approach to the construction of an indexing/search structure. However, learning PCTs (and decision trees in general) is unstable, i.e., the learned tree can change substantially for small changes in the data. To further improve the discriminative power and the robustness of our system, we use an ensemble (random forest) of PCTs.

We evaluate the proposed system on three benchmark databases for image retrieval: *Oxford5K* [31], *Paris* [32] and *Pythia* [35]. The first and the second database contain images of 11 famous Oxford landmarks and 12 famous Paris landmarks, respectively, while the third database contains general purpose images. Using these databases, we explore the different parameter settings for the proposed system and compare its performance with that of two state-of-the-art approaches for image retrieval - approximate *k*-means [1,25] and extremely randomized tree ensembles (ExtraTrees) [18]. Furthermore, we use spatial re-ranking of a shortlist of top ranked results to further boost the retrieval performance [3,11]. The spatial re-ranking is a post-processing step and is independent of the method used for retrieval.

The remainder of this paper is organized as follows. Section 2 briefly presents the related state-of-the-art methods for content-based image retrieval. The predictive clustering framework is described in Section 3. Section 4 introduces the proposed system for constructing an indexing structure for content-based image retrieval. Section 5 outlines the experimental design, while Section 6 presents the results from the experimental evaluation. Finally, the conclusions and a summary are given in Section 7.

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

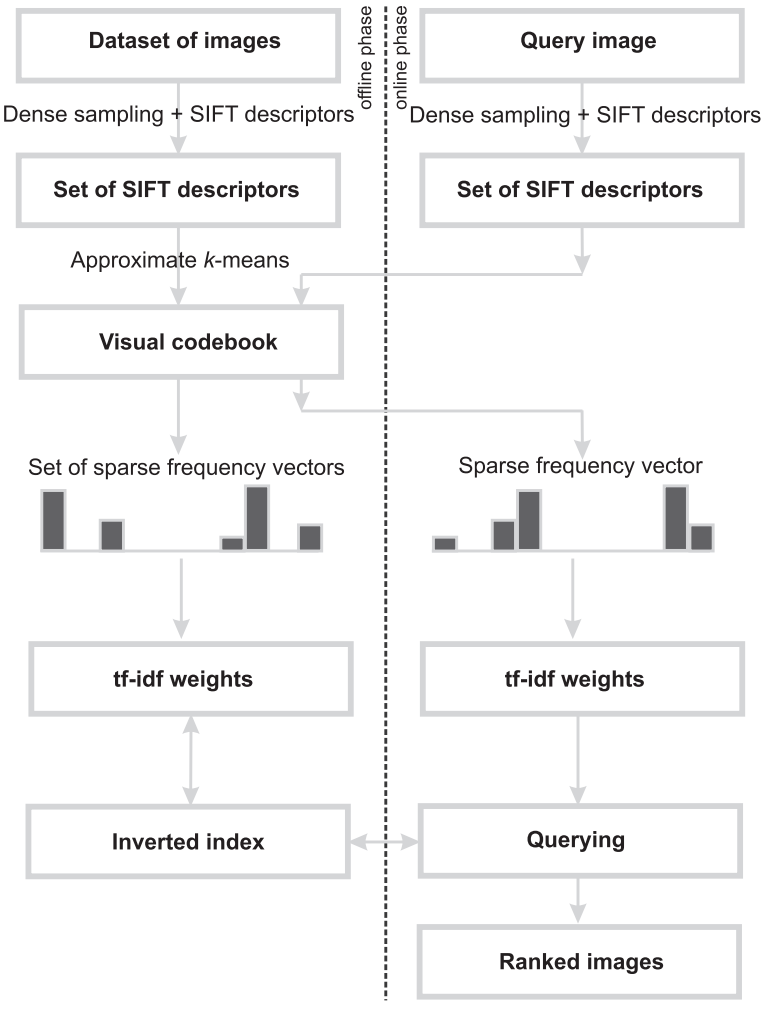


Fig. 1. The architecture of a typical system for image retrieval based on clustering methods, such as k-means or approximate k-means.

2. Related work

In this section, we briefly present the most widely used and state-of-the-art methods for CBIR based on the bag-of-visual-words approach. Many studies have shown that the bag-of-visual-words approach exhibits high retrieval performance for image retrieval given a query image of some particular object or an entire image [9,22]. A crucial step in the bag-of-visual-words approach is the codebook construction. The best results are achieved by using a large codebook that contains one million or even more entries/visual words, which requires clustering tens or even hundreds of millions of high-dimensional feature descriptors into one million or more clusters. The methods for CBIR mainly differ in the process of the visual codebook construction.

In the remainder of this section, we first describe the baseline method that is centered around k-means clustering. Next, we give an overview of the hierarchical k-means method. We then discuss the approximate k-means, which gives state-of-the-art retrieval results. Furthermore. we present a tree-based approach, which is related to the one presented here. Finally, we briefly discuss the differences between the method presented here and the state-of-the-art.

The visual codebook is typically constructed by applying k-means clustering to the local descriptors (e.g., Scale Invariant Feature Transform, i.e., SIFT, descriptors) extracted from the images [17,34]. The resulting clusters are actually the visual words. Although this method works very well for smaller databases (and consequently a small number of descriptors), it has very serious limitations when applied to the problem of large scale object retrieval [14]. It works with small visual codebooks, i.e., with thousands of visual words, while many datasets may have tens of thousands of visual words.

The hierarchical k-means (HKM) approach [21] addresses the issue of visual codebook size. HKM performs the clustering in a hierarchical manner with a predefined number of levels (n). At the first level, the descriptor space is clustered into k_1 clusters, then at the second level, each of the k_1 clusters is re-clustered into k_2 clusters, and so on, until level n. The final

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

visual codebook then consists of $k_1 \cdot k_2 \cdot \cdots \cdot k_n$ visual words. However, a major drawback of this method is that it is not a priori clear how many levels to use and how to choose the appropriate values for $k_1 \ldots, k_n$.

Philbin [25] proposed the approximate *k*-means (AKM) algorithm. In AKM, the exact nearest neighbor search is replaced with approximate nearest neighbor search in the assignment step when searching for the nearest cluster center for each point. In particular, the current cluster centers in each *k*-means iteration are organized by a forest of *k-d* trees to perform an accelerated approximate nearest neighbor search.

Most closely related to our approach is the CBIR method based on indexing random sub-windows (extracted from the images) with extremely randomized trees (ExtraTrees) [18]. These sub-windows are sampled from the images at random positions and random sizes. The sampled sub-windows are resized using bilinear interpolation to size 16×16. Each of the sub-windows is then described with HSV values (thus resulting in a 768 feature vectors). Next, these feature vectors are used to construct extremely randomized tree: each node split is selected randomly; thus, these trees are constructed in an unsupervised manner. These trees are then used as search structures for the retrieval phase. Furthermore, the extremely randomized trees method can be used to construct ensembles and thus further improve their retrieval performance [20].

Uijlings et al. [33] have performed experimental comparisons of visual codebooks constructed using *k*-means and tree-based approaches. The main conclusions from their study is that the tree-based approaches are more efficient than the approaches based on *k*-means. However, the improvement of the computational efficiency comes at the price of decreasing the discriminative power of the codebook. Similarly, in a direct comparison, Mareé et al. [18] compare their approach to hierarchical *k*-means and approximate *k*-means and conclude that their method is much more efficient than the competing methods, but exhibits lower retrieval performance (especially on larger image databases).

In this paper, we propose a method for image retrieval that does not just constructs a visual codebook, but also constructs the complete indexing/search structure. The proposed method is based on the predictive clustering framework. More specifically, we propose to construct the indexing structure using PCTs and random forests of PCTs. The major difference between the method proposed here and the one proposed by Mareé et al. [18] is in the selection of splits. Mareé et al. select the splits randomly, while we chose an informed approach. Namely, the split selection in the proposed method is performed by considering the compactness of the produced clusters. This split selection is the main reason for obtaining an indexing/search structure with high discriminative power.

This paper extends our previous work [8] in three major directions. First, we upgrade the retrieval method to replace both the clustering algorithm for visual codebook construction and the indexing/search structure. Namely, instead of using a random forest of smaller PCTs to obtain the visual codebook and then performing *tf-df* weighting, we use a random forest of fully grown PCTs which yield a larger visual codebook and use the codebook to calculate the distance between the query image and an image from the database. Note that, for each image descriptor, we keep a unique index/identifier for easier calculation of the distance. Second, we evaluate our approach on two additional benchmark datasets for image retrieval. Third, we implement a post-processing procedure, called spatial re-ranking that further enhances the retrieval results. All in all, we propose a method for image retrieval that can obtain state-of-the-art retrieval performance while being at least as computationally efficient as the competition.

3. Predictive clustering

In this section, we first outline the predictive clustering framework, which is the foundation of our codebook generation approach. We then briefly describe the predictive clustering trees for predicting multiple continuous targets. Finally, we present the method for construction of random forests of predictive clustering trees.

3.1. Predictive clustering framework

The notion of *predictive clustering* was first introduced by Blockeel [5]. The predictive clustering framework unifies two machine learning paradigms, predictive modeling and clustering, usually viewed as completely different. The connection between these is made by machine learning methods that partition the instances into subsets, such as decision trees and decision rules. These methods can be considered both as predictive and as clustering methods.

The task of predictive clustering is to identify clusters of instances that are close to each other both in the target and in the descriptive space. More specifically, in predictive modeling, the clusters of examples obtained using the target space only (e.g., a normal decision/regression tree focuses only on the target variable) are homogeneous in the target space (the target variables of the instances belonging to the same cluster have similar values). On the other hand, in clustering, the clusters obtained using the descriptive space only (e.g., *k*-means and *k*-nearest neighbors method look only the descriptive variables) are homogeneous in the descriptive space (the descriptive variables of the instances belonging to the same cluster have similar values). Predictive clustering can combine these two and produce clusters that are homogeneous both in the target and in the descriptive space.

In this work, we use a specific setting from the predictive clustering framework where only the descriptive space is provided. The descriptive space is thus equal to the target space, i.e., the descriptive variables are used to both calculate the statistics needed to partition the examples and provide symbolic descriptions of the obtained clusters. In other words, we

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

are constructing predictive clustering trees in an unsupervised setting. This focuses the predictive clustering setting on the task of clustering. This approach has two major advantages over classical clustering (such as k-means). First, we obtain the clusters much more efficiently as compared to standard clustering algorithms. Second, there are cluster descriptions for each of the clusters. The cluster description is the conjunction of the tests starting from the root node of the tree and following the path to the leaf (or the conditions used in a predictive clustering rule). This also improves the efficiency when new examples need to be projected into the clusters.

3.2. PCTs for predicting multiple continuous variables

The predictive clustering framework is implemented using decision trees (called predictive clustering trees - PCTs) and decision rules (called predictive clustering rules) as predictive models. The predictive clustering framework sees a decision tree as a hierarchy of clusters: the top-node corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. PCTs can be induced with a standard *top-down induction of decision trees* (TDIDT) algorithm [7]. The algorithm is presented in Table 1. It takes as input a set of examples (E) and outputs a tree. The heuristic (h) that is used for selecting the tests (t) is the reduction in variance caused by partitioning ($\mathcal{P}$) the instances (see line 4 of BestTest procedure in Table 1). By maximizing the variance reduction the cluster homogeneity is maximized, improving predictive performance. If no acceptable test can be found (see line 6), that is, if the test does not significantly reduce the variance, then the algorithm creates a leaf and computes the prototype of the instances belonging to that leaf.

The main difference between the algorithm for learning PCTs and a standard decision tree learner is that the former considers the variance function and the prototype function (that computes a label for each leaf), as parameters that can be instantiated for a given learning task. So far, the predictive clustering framework has been used for the prediction of multiple continuous variables, prediction of multiple discrete variables, hierarchical multi-label classification (HMC) and prediction of time series [15]. The predictive clustering framework is implemented in the CLUS system, which is available for download at http://clus.sourceforge.net/.

The variance and prototype functions of PCTs for predicting multiple continuous variables are instantiated as follows. The variance is calculated as the sum of the variances of the target variables, i.e., $Var(E) = \sum_{i=1}^{T} Var(Y_i)$. The variances of the target variables are normalized, so that each target variable contributes equally to the overall variance, given that the target variables can have completely different ranges. Namely, if one of the target variables is in the range (0, 1) and another in the range (10, 100) and normalization is not used, then the values of the second variable will contribute much more to the overall score than the values of the first variable. In addition, weighting of the (normalized values of the) target variables (so that the variance function gives more weight to some variables and less to others) is supported. The prototype function (calculated at each leaf) returns as a prediction the tuple with the mean values of the target variables, calculated by using the training instances that belong to the given leaf.

4. Codebook construction using predictive clustering

The architecture of the proposed system for image retrieval, based on random forest of PCTs, is presented in Fig. 2. The systems consists of an off-line phase and an on-line phase. The off-line phase implements the image description and the construction of the indexing/search structure for the retrieval. The on-line phase, on the other hand, implements the construction of the query image description, the querying of the indexed image database and the presentation of the results of the retrieval. In the remainder of this section, we discuss these phases in more detail.

Table 1
The top-down induction algorithm for learning predictive clustering trees.

procedure PCT(E) **returns** tree
1: $(t^*, h^*, \mathcal{P}^*) =$ BestTest(E)
2: **if** $t^* \neq$ *none* **then**
3: **for each** $E_i \in \mathcal{P}^*$ **do**
4: $tree_i =$ PCT(E_i)
5: **return** node(t^*, $\bigcup_i \{tree_i\}$)
6: **else**
7: **return** leaf(Prototype(E))

procedure BestTest(E)
1: $(t^*, h^*, \mathcal{P}^*) = (none, 0, \varnothing)$
2: **for each** possible test t **do**
3: $\mathcal{P} =$partition induced by t on E
4: $h = Var(E) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var(E_i)$
5: **if** $(h > h^*) \wedge$ Acceptable($t, \mathcal{P}$) **then**
6: $(t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
7: **return** $(t^*, h^*, \mathcal{P}^*)$

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

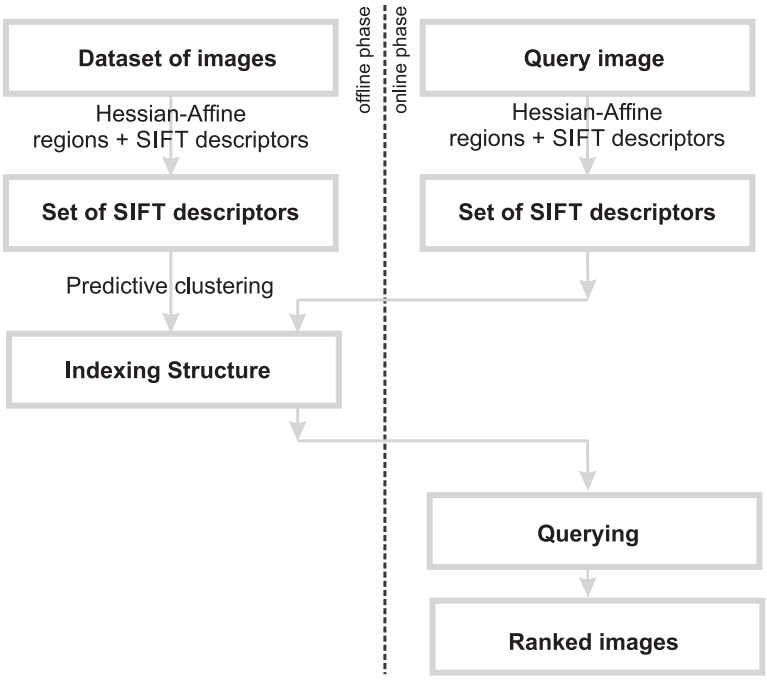


Fig. 2. The architecture of the proposed system for image retrieval based on (a random forest of) predictive clustering trees.

The off-line phase starts with the generation of the local descriptors for the images. For each image in the database, affine-invariant Hessian regions are located [19,25]. Typically 3300 such regions are detected in an image with a size of 1024×768. For each of these affine regions, a 128-dimensional Scale Invariant Feature Transform (SIFT) descriptor is then computed [17]. Next, the computed descriptors are used to create the visual codebook and the indexing structure, which is the central part of the image retrieval system.

The proposed method for constructing the indexing structure is as follows. First, we randomly select a subset of the local (SIFT) descriptors from all of the training images. Next, the selected local descriptors constitute the training set used to construct a PCT. For the construction of a PCT, we set the descriptive attributes (i.e., the 128 dimensional vector) to be also target and clustering attributes. Note that this feature is a unique characteristic of the predictive clustering framework.

The PCTs are computationally efficient: it is very fast to both construct them and use them to make a prediction. However, tree learning is unstable, i.e., the structure of the learned tree can change substantially for small changes in the training data [6]. To overcome this limitation and to further improve the discriminative power of the indexing structure, we use an ensemble (i.e., random forest) of PCTs (as in Moosmann et al. [20]).

We use the random forest method to create the base classifiers in the ensembles [6,15]. Each PCT from the ensemble is obtained by first creating a bootstrap sample of the training set and then randomly changing the feature set during learning. More precisely, at each node in the decision tree, a random subset of the input attributes is taken, and the best feature is selected from this subset (instead of the set of all attributes). The number of attributes that are retained is given by a function f of the total number of input attributes x (e.g., $f(x) = x$, $f(x) = \sqrt{x}$, $f(x) = \lfloor \log_2 x \rfloor + 1$).

In the proposed system, PCTs (or a random forest of PCTs) represents the search/indexing structure. Namely, for each image descriptor (i.e., each training example used to construct the PCTs), we keep a unique index/identifier. The identifier consists of the image ID from which the local descriptor was extracted coupled with a descriptor ID. For example, if a local descriptor can be identified with the following ID: *Im*27891*D*2258 then it was extracted from the image with an ID of *Im*27891 and it was the 2258-th descriptor from that image. This indexing allows for faster computation of the image similarities.

The majority of CBIR systems [13,21,25] would proceed by applying the standard *tf-idf* weighting scheme to create and index structure. However, our method does not perform this step and the indexing is represented by the PCTs. With the creation of the PCTs, the off-line phase of the retrieval is completed.

In the on-line phase, we produce a ranked list of images. This phase starts by extracting N_Q local SIFT descriptors from the query image I_Q. The N_Q local SIFT descriptors are then propagated through each PCT from the ensemble. Next, for the query image, we calculate its similarity to each reference/training image I_i, using the similarity definitions provided bellow. The calculation of the similarity is one of the most important parts of the proposed method. Note that this specific part can be further modified and updated depending of the application at hand.

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

For the definition of the similarity measure between two images, we use the recommendations given by Mareé et al. [18]. We begin by defining the similarity between two local SIFT descriptors d_1 and d_2 using a single PCT $\mathcal{T}$. This similarity is defined as follows:

$$sim(d_1, d_2)_{\mathcal{T}} = \begin{cases} \frac{1}{N_{\mathcal{L}}}, & d_1, d_2 \text{ belong to the same leaf } \mathcal{L} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where $N_{\mathcal{L}}$ is the number of descriptors in the leaf $\mathcal{L}$. Next, the similarity computed over a random forest of T PCTs is defined as the sum over all the similarities from the base predictive models:

$$sim(d_1, d_2)_{RF} = \frac{1}{T} \sum_{t=1}^{T} sim(d_1, d_2)_{\mathcal{T}_t} \quad (2)$$

Finally, given the similarity measure between descriptors, the similarity between two images I_1 and I_2 is computed as follows:

$$sim(I_1, I_2) = \frac{1}{|D_1||D_2|} \sum_{d_1 \in D_1, d_2 \in D_2} sim_{RF}(d_1, d_2) \quad (3)$$

where D_1 and D_2 are the sets of all descriptors that are extracted from the images I_1 and I_2, respectively. The similarity between two images is thus the average similarity between all pairs of their descriptors. In other words, the similarity measure considers two images more similar to each other if their descriptors are more frequently encountered together in the same leaves across the different PCTs in the ensemble.

At first glance, it may seem that calculating the similarity between two images is computationally expensive: It depends quadratically on the number of descriptors extracted. However, recall that we are keeping identifiers for the descriptors that belong to the leaves of the PCTs. Considering this, when the query image descriptors are sorted down a PCT, we calculate the similarities between the descriptors from the query image that belong to the given leaf and the descriptors of the other training images. The calculation is repeated and accumulated for all of the PCTs in the ensemble. The similarity calculation is thus much more efficient than it seems at first glance: It mainly depends of the number of leaves that the query image descriptors are sorted into.

Finally, we briefly compare the designs of a typical retrieval system (depicted in Fig. 1) and and the proposed system (depicted in Fig. 2). The two designs are equal in two aspects: (1) they both have an off-line and an on-line phase, and (2) they both perform (dense) sampling of key-points and calculate local (SIFT) descriptors for the two phases. However, the design we propose in this paper is much simpler than the design of a typical retrieval system. Namely, instead of the separate processes of visual codebook construction, image projection for obtaining the histograms, calculation of *tf-idf* weights, and calculation of the inverted index, in the proposed system we have a single process that does everything simultaneously. These simplifications are valid for both phases.

5. Experimental setup

In this section, we present the experimental design we used to evaluate the proposed algorithm and compare it to other approaches. First, we present the datasets of images that we use. Second, we describe the evaluation metric that we use to assess the retrieval performance. Next, we state the experimental questions under investigation in this study. Finally, we present a spatial re-ranking algorithm that post-processes the retrieval results and enhances the retrieval performance.

5.1. Image datasets

The *Oxford Buildings* dataset [31] is typically used as a benchmark dataset for large scale particular object retrieval. It consists of 5062 high-resolution images (1024 × 768) automatically downloaded from Flickr by searching for 11 Oxford landmarks. The images are then manually annotated as to provide a solid ground truth for evaluating the performance of retrieval systems. It also defines 55 queries (5 query objects/images for each of the 11 Oxford landmarks) that are used for performance evaluation. Each query consists of an image and query region/object of interest. This dataset is very challenging due to the substantial variations in scale, viewpoint and lighting conditions of the images and the objects that need to be retrieved. Example images from this dataset are shown in Fig. 3.

In addition to this dataset, often called Oxford 5*K*, we use two other datasets to test the scalability of the retrieval systems: Oxford 100*K* and Oxford 1*M* [25]. These datasets are not challenging only in terms of size, but also due to the fact that they cover a much broader spectrum than the Oxford buildings dataset: they include a mixture of different scenes, objects, people, and buildings from all around the world.

The *Paris* dataset contains 6,300 high resolution (1024 × 768) images obtained from Flickr by querying the associated text tags for famous Paris landmarks such as “Paris Eiffel Tower” or “Paris Triomphe”. Example images from this dataset are shown in Fig. 4. More details for this dataset can be found in [32].

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

Fig. 3. A selection of images used for constructing the indexing structure from the *Oxford5K* database.

The *Pythia* image database [35] was designed to overcome the limitations in common benchmark image collection that are introduced by binary relevance judgments. This image database aims at providing a benchmark for user-centered database. The database consists of randomly selected 5555 general purpose images contributed by 19 photographers. Example images from this dataset are shown in Fig. 5. The variance in photographic motifs and styles is ensured by selecting contributors from different demographic groups. Neither have the images within the collection been processed extensively nor have duplicate images been removed. Hence, the data can be considered a realistic sample from a typical user's hard-disk. In addition, the database contains 74 pre-defined query images that can be used to assess the performance of a retrieval system.

5.2. Evaluation measure

The performance measure most widely used for evaluating methods in image retrieval is the *mean average precision* (*mAP*) [9,22]. We thus adopt *mAP* to evaluate the performance of our method for particular object retrieval and the discriminative power of its visual codebook.

The *mAP* is calculated as follows. For each of the query images (for each benchmark dataset the number of query images is different), the average precision (*AP*) is computed as the area under the precision-recall curve (*AUPRC*) for that query. This score combines both precision and recall into a single performance score. Precision is defined as the ratio of retrieved positive images to the total number of images retrieved. Recall is defined as the ratio of the number of retrieved positive images to the total number of positive images in the dataset. An ideal precision-recall curve has precision 1 over all recall levels and this corresponds to an average precision of 1 and also *AUPRC* of 1. The values of the *AUPRC* score are in the range [0,1]: if they are larger then the retrieval performance of the system is better. The overall *mAP* value is obtained by taking the mean of the average precisions of all queries and is used as a single number to evaluate the overall performance.

5.3. Spatial re-ranking

The precision of a retrieval system can be improved by re-ranking of the images based on their spatial consistency with the query image [11,28,36,37]. Note that in order to be able to exploit the spatial information, the retrieval system needs to

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

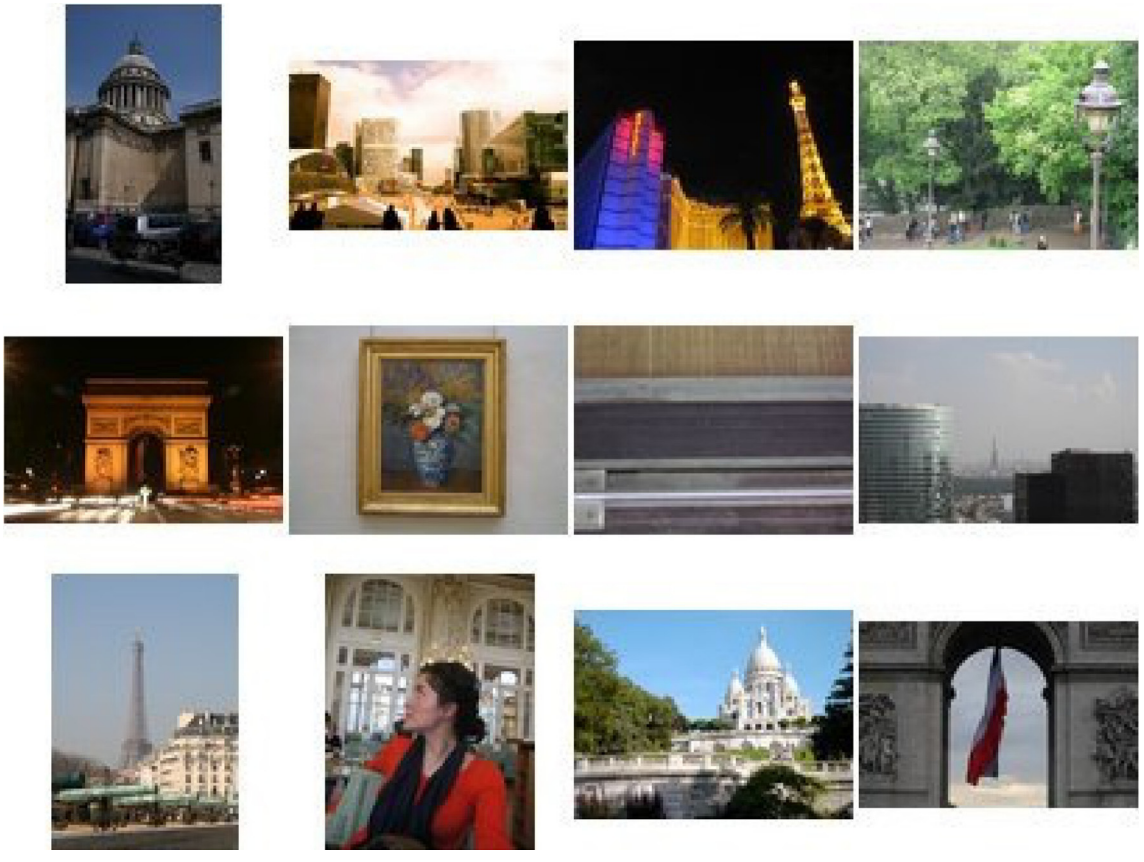


Fig. 4. A selection of images used for constructing the indexing structure from the *Paris* database.

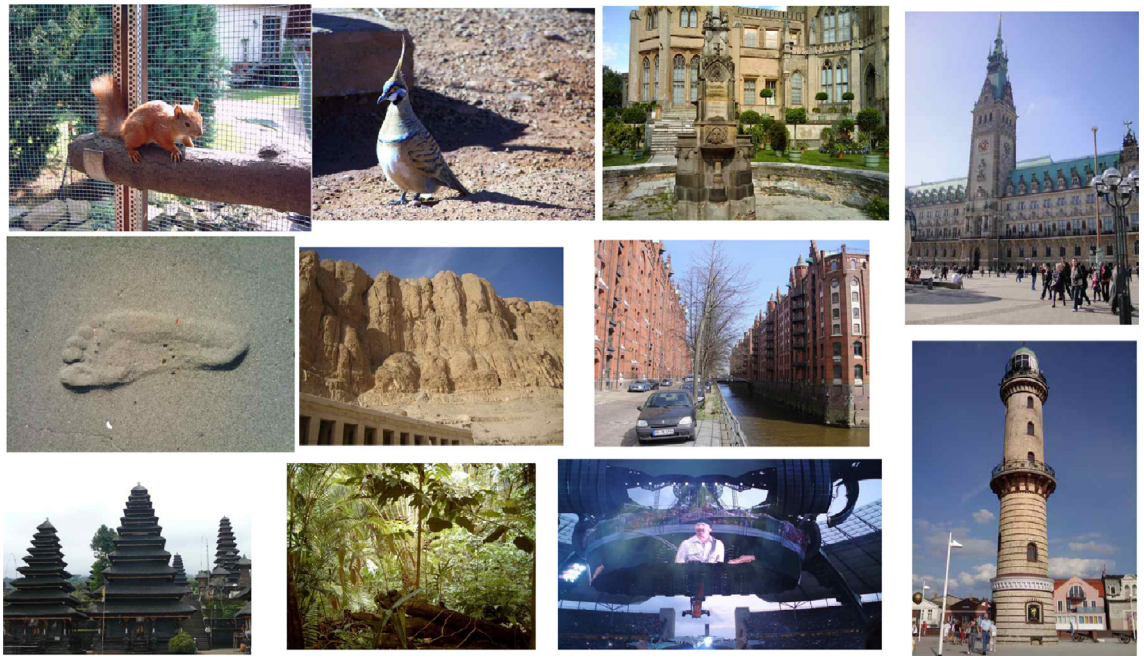


Fig. 5. A selection of images used for constructing of the indexing structure from the *Pythia* database.

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

store additional data on a per-feature basis. Namely, the additional data consists of the location of the feature in the image and usually the scale or the shape of the local region [24]. Since spatial consistency estimation is computationally costly, only a short-list of top ranked results is re-ranked. Spatial re-ranking is a post-processing step and it can be applied independently of the image retrieval method.

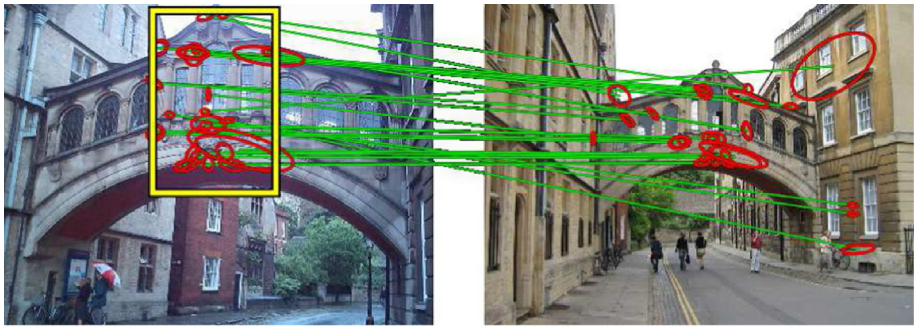
We use the spatial re-ranking method proposed by Philbin et al. [26]. This method re-ranks images based on the number of visual words consistent with an affine transformation (inliers) between the query and the database image. A rigid affine transformation is fitted very efficiently using the iterative random sample consensus (RANSAC) [10] method. Here, we assume that the images are upright and only a single pair of matched visual words is used to propose an affine transformation between the query and a database image by exploiting the local shape of affine co-variant regions [19,27]. Fig. 6 shows an example of a spatially verified pair of images.

5.4. Experimental questions

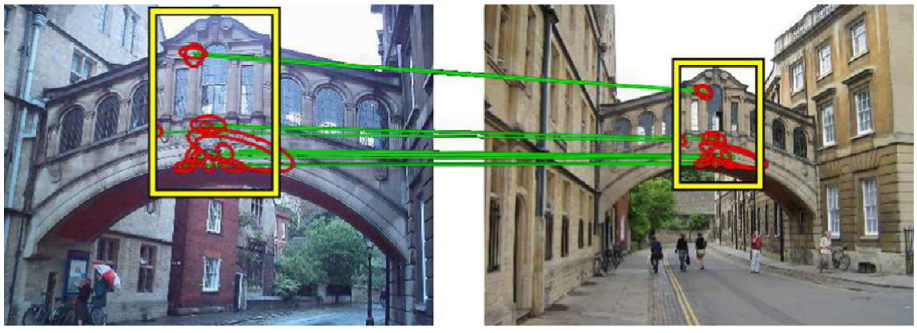
We focus the experimental evaluation of the proposed method on the following four research questions:

1. Does the increase of the number of descriptors and the number of trees in the random forests of PCTs influence the retrieval performance of our system?
2. Does the proposed retrieval system with random forest of predictive clustering trees have better retrieval performance than two state-of-the-art systems using approximate k-means and extremely randomized tree ensembles?
3. Does the use of spatial re-ranking for post-processing of the results improve the performance of the proposed method?
4. Is the proposed system for image retrieval efficient and scalable to larger problems?

In order to answer these four questions, we design the following experimental setup. We address the first question by constructing visual codebooks and indexing structures using different numbers of local descriptors and different numbers of trees. This will help us to optimally parametrize the proposed system. For answering the second question, we compare the performance of the visual codebook constructed using the random forest of PCTs with the performance of the ones constructed using approximate k-means and an ensemble of extremely randomized trees. This comparison is justified by the fact that most current results related to content-based particular object retrieval and content-based retrieval in general are obtained by using large visual codebooks created using these two systems. We follow the standard bag-of-visual-words framework described by Philbin et al. [26]. We use affine-Hessian interest points [19], a vocabulary of $1M$ visual words



(a) All visual word matches



(b) Spatially verified matches

Fig. 6. Spatial verification. (left) The query image and region of interest are shown on the left; (right) A retrieved database image. Figure taken from Arandjelović [1].

obtained using approximate *k*-means, the ExtraTrees ensemble consists of 25 extremely randomized trees constructed using 10*M* descriptors [23], and spatial re-ranking of the top 200 TF-IDF results using an affine transformation. We address the third question by performing spatial re-ranking of the retrieval results obtained by both methods and comparing their retrieval performance without and with the re-ranking. For answering the fourth question, we apply the visual codebook obtained by our method to the Oxford 100*K* and Oxford 1*M* image datasets. The performance results are compared with the results obtained using approximate *k*-means [25].

6. Results and discussion

In this section, we present and discuss the results obtained from the experimental evaluation of the proposed method. First, we discuss the influence of the number of trees in the random forest and the number of local descriptors considered on the retrieval performance of our system. We then compare the performance of the proposed method to that of approximate *k*-means and ensemble of extremely randomized trees. Next, we discuss the scalability of the proposed method by presenting the time needed to construct the indexing structure (in our case to build the random forest of PCTs) and the time needed to propagate the local descriptors for a given query image in all the trees and calculate the similarity. Finally, we present a visual inspection of the retrieval results from our system.

The proposed system has two parameters that influence its retrieval performance and efficiency: the number of local (SIFT) descriptors used for constructing the indexing structure and the number of trees that comprise the indexing structure. To investigate the influence of these parameters, we first construct different indexing structures by varying the number of local SIFT descriptors used to construct the random forest of PCTs: In this case, the number of trees in the random forests is fixed to 12. The results for varying the number of local descriptors from 1 M to 16 M are given in Fig. 7(a). They show that including more local descriptors in the process of building the indexing structure increases the retrieval performance of our system for each of the three image datasets. The increase in performance is most pronounced when increasing the number of descriptors from 1 M to 5 M, while performance begins to saturate after 10 M descriptors are used. After that, increasing the number of local descriptors does not significantly improve the retrieval performance.

Next, we investigate the influence of the number of PCTs in the random forest on the retrieval performance of our system. For these experiments, the number of local SIFT descriptors was set to 10 M (according to the discussion above). The results obtained by varying the number of trees from 4 to 100 are presented in Fig. 7(b). They indicate that the increase of the number of trees does increase the retrieval performance. The retrieval performance saturates after the 25-th PCT is added to the forest. Considering this, we can conclude that the proposed system yields optimal results using 10 M descriptors and a random forest with 25 PCTs. We also observe that a further increase of the retrieval performance is possible by further increasing the values of the two parameters, however, the increase is marginal and comes at the cost of decreased computational efficiency.

Once we have parametrized the proposed system, we proceed to compare its performance with the two state-of-the-art methods in the area of image retrieval - bag-of-visual-words with approximate *k*-means and ensemble of extremely randomized trees. For our system, we selected the optimal parameter values given above and for the competing methods we selected the parameter values as follows. For approximate *k*-means method, we use the recommended optimal values given by Philbin [25]: 16 M local descriptors are used to construct a codebook with a size of 1 M visual words and this is done in 20 iterations. For the ensemble of extremely randomized trees, we used the same optimal values as used for the ensemble of predictive clustering trees. The results are given in Table 2. We can see that the proposed system clearly outperforms both

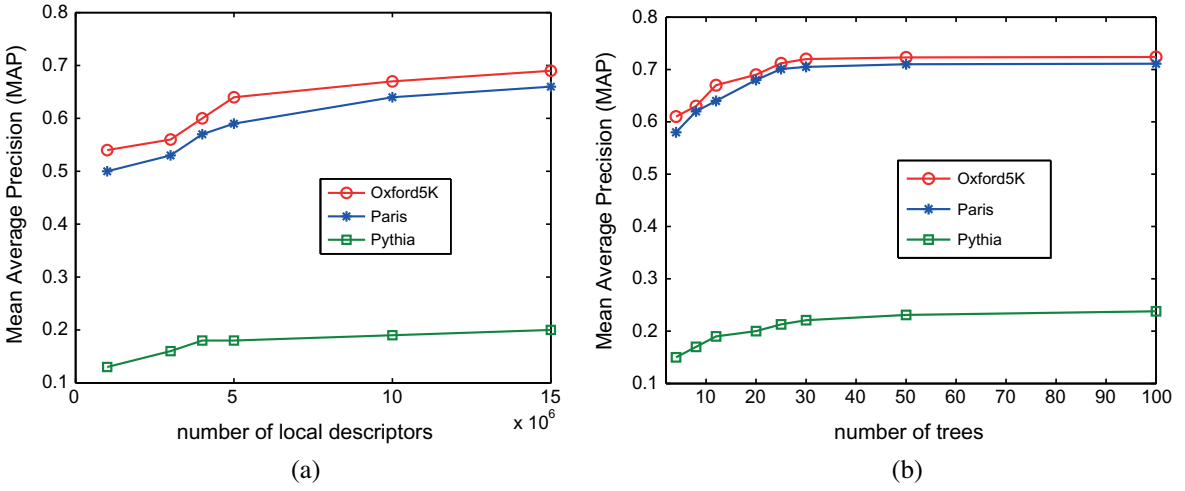


Fig. 7. Retrieval performance of the proposed system (given as mean average precision) for (a) different number of local descriptors with the number of trees set to 12 and (b) different numbers of PCTs with the number of local descriptors set to 10 M.

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

Table 2
Comparison of the retrieval performance (given as mean average precision) of the system based on approximate *k*-means (AKM), the system based on an ensemble of extremely randomized trees (ExtraTrees) and the proposed system based on a random forest of PCTs (RF of PCTs) on the three image datasets without and with spatial re-ranking of the top 200 results.

Image dataset	Without spatial re-ranking			With spatial re-ranking		
	AKM	ExtraTrees	RF of PCTs	AKM	ExtraTrees	RF of PCTs
Oxford5K	0.680	0.675	0.712	0.720	0.710	0.761
Paris	0.687	0.661	0.701	0.688	0.673	0.710
Pythia	0.164	0.172	0.213	0.170	0.189	0.234

Table 3
Comparison of the retrieval performance (mean average precision) of the system based on *AKM* and the proposed system based on random forests of PCTs (RF of PCTs) on the three *Oxford5K* image datasets of different size, without and with spatial re-ranking of the top 200 results.

Image dataset	Without spatial re-ranking			With spatial re-ranking		
	AKM	ExtraTrees	RF of PCTs	AKM	ExtraTrees	RF of PCTs
Oxford5K	0.680	0.675	0.712	0.720	0.710	0.761
Oxford5K + 100K	0.581	0.562	0.592	0.642	0.621	0.653
Oxford5K + 100K + 1M	0.461	0.451	0.482	0.510	0.491	0.531

the approximate *k*-means method and the ExtraTrees method on the three selected image datasets, even though, the approximate *k*-means used more local descriptors than our system.

The performance figures obtained after post-processing the query results with spatial re-ranking are shown in the right-hand side of Table 2. We can note that our system again clearly outperforms both competing systems. Additionally, the increase of the retrieval performance using spatial re-ranking is slightly larger for our system than for the competing systems: 7.7% compared to 7.25% for ExtraTrees and 6.16% for AKM. We would like to note that further improvements of these results are also possible by performing average query expansion or database-side feature augmentation [1,2]. Since these improvements are obtained by post-processing the retrieval results, they are independent from the retrieval method and we expect that they will preserve the relative performance advantage of our method.

Furthermore, we compare the efficiency of the proposed system to the efficiency of the competing systems. The efficiency in the context of image retrieval is measured in terms of the time needed to construct the indexing structure (i.e., for the off-line phase) and the time needed to project the local descriptors through the indexing structure and calculate the similarity to the other images (i.e., for the on-line phase). The off-line phase of both the proposed system and the system based on approximate *k*-means requires ∼100 h of execution time, while for the ensemble of extremely randomized trees ∼62.5 h. For the proposed system, each of the 25 PCTs requires approximately 4 h, thus a total of 25×4 h are required for the complete indexing structure. Each extremely randomized tree requires approximately 2.5 h, i.e., a total of ∼62.5 h. Each of the 20 iterations of the state-of-the-art system based on AKM requires 5 h, yielding a total of 20×5 h. Next, the on-line phase has the following time requirements. Both the proposed system and the system based on ExtraTrees return the result in 0.08 s and the AKM system in 0.1 s. Note that the time needed for retrieving the relevant images does not depend on the database size and the specific method used to construct the retrieval structure. It mainly depends on the number of images returned as a result from the query [26].

We could further improve the efficiency of our system by either reducing the number of trees or pruning the trees (i.e., constructing smaller trees), but that would come at the price of reducing the retrieval performance. We have previously shown a similar effect [8], i.e., we have shown that we can construct a codebook that has slightly better retrieval performance than AKM, but is more efficient. All in all, we can conclude that the proposed system has efficiency comparable to that of the state-of-the-art retrieval system and that it could be successfully scaled to even larger problems.

Next, we evaluate the scalability of our method on the *Oxford5K*, *Oxford5K + 100 K* and *Oxford5K + 100 K + 1 M* datasets. For all three systems, we kept the parameter values given above. The results are given in Table 3. From the results, we can see that the retrieval performance of our method is better than the one of both approximate *k*-means and ensemble of extremely randomized trees. We can also note the decrease in performance with the increase of the size of the image database. This is mainly due to the fact that these datasets now include many more noisy images outside of the domain of Oxford buildings. The spatial re-ranking of the results, as for the other databases, improves the retrieval performance of all systems.

We visually inspect the retrieval results and illustrate part of them in Figs. 8 and 9. The first image in each row is the image that contains the query object, while the remaining four images are part of the retrieved images. The retrieval results given in Fig. 8 reveal that the proposed method performs very well when there are considerable variations in the viewpoint, the image scale, the lighting and partial occlusion of the object.

The retrieval results given in Fig. 9, on the other hand, illustrate examples in which the proposed systems fails to successfully retrieve the particular query object. For the first query (the first row of images), the failure is due to the presence of images that are visually plausible and consistent with the query image (the retrieved images also contain windows with bars).

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

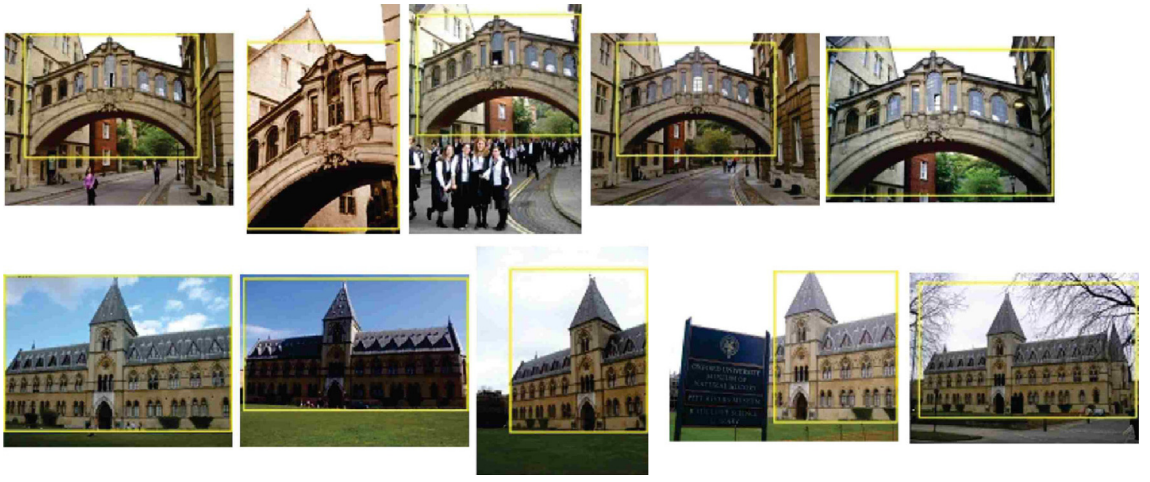


Fig. 8. Examples of searching the 5 K dataset for: Bridge of sighs, Hertford College (first row), Pitt Rivers Museum (second row). The first image in each row is the query image and the selected region with yellow color is the query object/building. The other four images in each row are the query result images obtained by using the proposed system. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

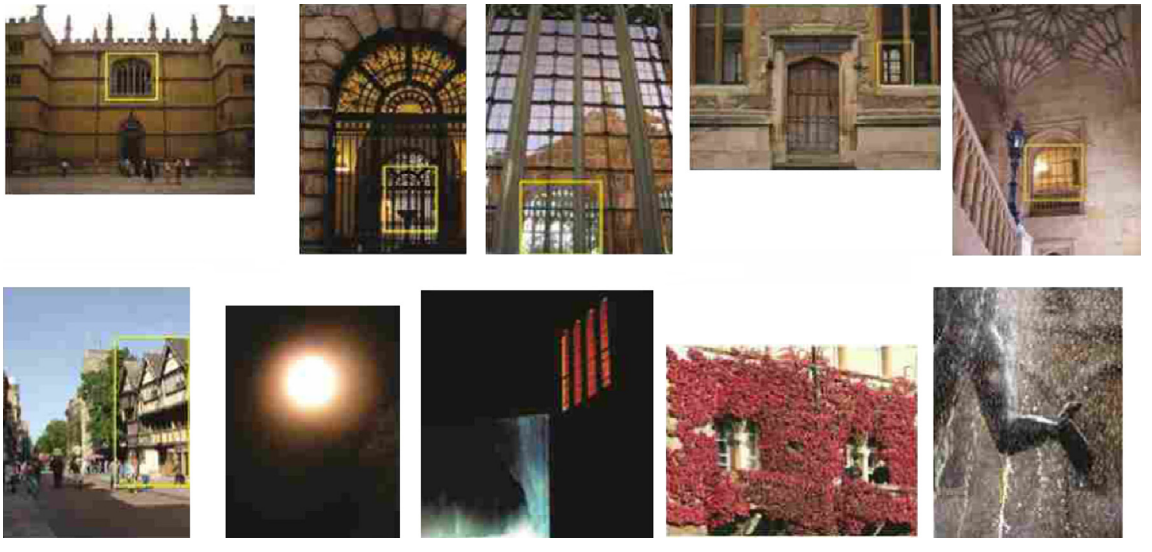


Fig. 9. Examples of errors in retrieval for two query images (the first image in each row). The false positives for the first query are visually plausible, but the false positive for the second query is due to visual ambiguities in the local regions and the low numbers of visual words in the retrieved image.

The failure for the second query (the second row of images) is a result of the 'burstiness' effect [12]. The burstiness effect appears when a given local descriptor appear more frequently in an image than a model would predict. In this context, the burstiness distorts the image similarity measure and thus pollutes the ranking of the images in the retrieval results. In our case, this is a result of the rich texture present in the resulting images such as water-drops and flowers.

7. Conclusion

In this paper, we present a method for fast and efficient construction of visual codebooks for particular object retrieval from large scale image databases. The construction of a codebook is an essential part of the bag-of-visual-words approach to image retrieval. It should thus be efficient and deliver a codebook with high discriminative power. However, the construction of a visual codebook is a bottleneck in the bag-of-visual-words approach, because it typically uses k-means clustering over millions of image patches to obtain several tens of thousands of visual words. While existing approaches are able to solve the efficiency issue, a part of the discriminative power of the codebook is sacrificed for better efficiency. In this paper, we propose to use predictive clustering trees (PCTs) for codebook construction. In this way, we efficiently construct visual codebooks and increase the discriminative power of the dictionary.

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

PCTs are a generalization of decision trees and are capable of performing predictive modeling and clustering simultaneously. More specifically, the method we propose uses PCTs for predicting multiple targets to construct the visual codebook – each leaf in the tree is a visual word. Furthermore, we construct a random forest of PCTs to increase the stability of the codebook and its discriminative power. The overall codebook is obtained by concatenating the codebooks from each tree. This codebook also represents the indexing structure used to retrieve images similar to query images.

We evaluated the proposed method on the Oxford buildings, Paris and Pythia image dataset, which are benchmark datasets for large scale particular object retrieval and content-based image retrieval in general. We used two more variants of the Oxford buildings dataset that include 100 K and 1 M images. We compare the proposed method to two state-of-the-art methods based on approximate k-means clustering and an ensemble of extremely randomized trees.

The results from the experimental evaluation reveal the following. First and foremost, our system exhibits clearly better retrieval performance than both the competing methods. Next, the increase of the number of local descriptors and number of PCTs used to create the indexing structure improve the retrieval performance of the system. More specifically, our system achieves optimal performance (excellent retrieval performance and excellent computational efficiency) when using $10M$ local SIFT descriptors to construct an indexing structure consisting of 25 PCTs. Finally, the proposed and the competing systems have similar efficiency, both in terms of time needed to construct the indexing structure and to perform the retrieval. All in all, the proposed system obtains better retrieval results than the competing systems based on AKM and ExtraTrees, while maintaining good computational efficiency.

Acknowledgments

We would like to acknowledge the support of the European Commission through the project MAESTRA – Learning from Massive, Incompletely annotated, and Structured Data (Grant No. ICT-2013-612944).

References

[1] R. Arandjelović, Advancing Large Scale Object Retrieval, Ph.D. thesis, University of Oxford, 2013.
[2] R. Arandjelovic, A. Zisserman, Three things everyone should know to improve object retrieval, IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 2911–2918.
[3] Y. Avrithis, G. Tolias, Hough pyramid matching: speeded-up geometry re-ranking for large scale image retrieval, Int. J. Comput. Vis. 107 (1) (2014) 1–19.
[4] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, ACM Press, 1999.
[5] H. Blockeel, Top-Down Induction of First Order Logical Decision Trees. Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1998.
[6] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[7] L. Breiman, J. Friedman, R. Olshen, C.J. Stone, Classification and Regression Trees, Chapman & Hall/CRC, 1984.
[8] I. Dimitrovski, D. Kocev, S. Loskovska, S. Džeroski, Fast and scalable image retrieval using predictive clustering trees, Discovery Science, Lecture Notes in Computer Science, vol. 8140, Springer, Berlin Heidelberg, 2013, pp. 33–48.
[9] M. Everingham, L.V. Gool, C. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2010 (VOC2010), 2010. <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html>.
[10] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[11] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004 ISBN: 0521540518.
[12] H. Jégou, M. Douze, C. Schmid, On the burstiness of visual elements, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1169–1176.
[13] H. Jégou, H. Harzallah, C. Schmid, A contextual dissimilarity measure for accurate and efficient image search, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[14] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: IEEE International Conference on Computer Vision, 2005, pp. 604–610.
[15] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Tree ensembles for predicting structured outputs, Pattern Recogn. 46 (3) (2013) 817–833.
[16] Y. Liu, D. Zhang, G. Lu, W.-Y. Ma, A survey of content-based image retrieval with high-level semantics, Pattern Recogn. 40 (1) (2007) 262–282.
[17] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[18] R. Mare, P. Geurts, L. Wehenkel, Content-based image retrieval by indexing random subwindows with randomized trees, Computer Vision ACCV, Lecture Notes in Computer Science, vol. 4844, Springer, Berlin Heidelberg, 2007, pp. 611–620.
[19] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Int. J. Comput. Vis. 60 (1) (2004) 63–86.
[20] F. Moosmann, E. Nowak, F. Jurie, Randomized clustering forests for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1632–1646.
[21] D. Nistér, H. Stewénius, Scalable recognition with a vocabulary tree, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2161–2168.
[22] S. Nowak, ImageCLEF@ICPR contest: challenges, methodologies and results of the photo annotation task, in: International Conference on Pattern Recognition, 2010, pp. 489–492.
[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[24] M. Perd'och, O. Chum, J. Matas, Efficient representation of local geometry for large scale object retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 9–16.
[25] J. Philbin, Scalable Object Retrieval in Very Large Image Collections. Ph.D. thesis, University of Oxford, Oxford, UK, 2010.
[26] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[27] F. Schaffalitzky, A. Zisserman, Multi-view matching for unordered image sets, or how do i organize my holiday snaps? in: European Conference on Computer Vision – ECCV, 2002, pp. 414–431.
[28] X. Shen, Z. Lin, J. Brandt, S. Avidan, Y. Wu, Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking, in: IEEE Conference on Computer Vision and Pattern Recognition, June 2012, pp. 3013–3020.
[29] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: IEEE Conference on Computer Vision, 2003, pp. 1470–1477.

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012

[30] E. Spyromitros-Xioufis, S. Papadopoulos, I. Kompatsiaris, G. Tsoumakas, I. Vlahavas, A comprehensive study over VLAD and product quantization in large-scale image retrieval, IEEE Trans. Multimedia 16 (6) (2014) 1713–1728.
[31] The Oxford Buildings Dataset, 2013. <http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/>.
[32] The Paris Dataset, 2013. <http://www.robots.ox.ac.uk/vgg/data/parisbuildings/>.
[33] J. Uijlings, A. Smeulders, R. Scha, Real-time bag of words, approximately, in: ACM International Conference on Image and Video Retrieval, 2009, pp. 1–8.
[34] K. van de Sande, T. Gevers, C. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.
[35] D. Zellhöfer, An extensible personal photograph collection for graded relevance assessments and user simulation, in: ACM International Conference on Multimedia Retrieval, 2012, pp. 29:1–29:8.
[36] Y. Zhang, Z. Jia, T. Chen, Image retrieval with geometry-preserving visual phrases, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 809–816.
[37] W.-L. Zhao, X. Wu, C.-W. Ngo, On the annotation of web videos by efficient near-duplicate search, IEEE Trans. Multimedia 12 (5) (2010) 448–461.

Please cite this article as: I. Dimitrovski et al., Improving bag-of-visual-words image retrieval with predictive clustering trees, Information Sciences (2015), http://dx.doi.org/10.1016/j.ins.2015.05.012