Hierarchical Annotation of Medical Images

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

[a] *Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[b] *Department of Computer Science and Computer Engineering, Faculty of Electrical Engineering and Information Technologies, Rugjer Boshkovikj bb, 1000 Skopje, Republic of Macedonia*

Abstract

We present a hierarchical multi-label classification (HMC) system for medical image annotation. HMC is a variant of classification where an instance may belong to multiple classes at the same time and these classes/labels are organized in a hierarchy. Our approach to HMC exploits the annotation hierarchy by building a single predictive clustering tree (PCT) that can simultaneously predict all annotations of an image. Hence, PCTs are very efficient: a single classifier is valid for the hierarchical semantics as a whole, as compared to other approaches that produce many classifiers, each valid just for one given class. To improve performance, we construct ensembles of PCTs. We evaluate our system on the IRMA database that consists of X-ray images. We investigate its performance under a variety of conditions. To begin with, we consider two ensemble approaches, bagging and random forests. Next, we use several state-of-the-art feature extraction approaches and combinations thereof. Finally, we employ two types of feature fusion, i.e., low- and high-level fusion. The experiments show that our system outperforms the best-performing approach from the literature (a collection of SVMs, each predicting one label at the lowest level of the hierarchy), both in terms of error and efficiency. This holds across a range of descriptors and descriptor combinations, regardless of the type of feature fusion used. To stress the generality of the proposed approach, we have also applied it for automatic annotation of a large number of consumer photos with multiple annotations organized in semantic hierarchy. The obtained results show that this approach is general and easily applicable in different domains, offering state-of-the-art performance.

Keywords: Automatic Image Annotation, Hierarchical Multi-Label Classification, Predictive Clustering Trees, Feature Extraction from Images

1. Introduction

Digital imaging in medicine is in constant growth due to the increasing availability of imaging equipment in hospitals. Average-sized radiology departments now produce several tera-bytes of data annually. This prompts for efficient systems for image annotation, storage, retrieval and mining. Typically, medical image databases are accessed via textual information through the

[*] Corresponding author (telephone: +389 2 3099 159)

Email addresses: ivicad@feit.ukim.edu.mk (Ivica Dimitrovski), Dragi.Kocev@ijs.si (Dragi Kocev), suze@feit.ukim.edu.mk (Suzana Loskovska), Saso.Dzeroski@ijs.si (Sašo Džeroski) standard Picture Archiving and Communication System (PACS) [1], [2]. PACS integrates imaging modalities and interfaces with hospital and departmental information systems to manage storage and distribution of images to medical personnel, researchers, clinics, and imaging centers. An important requirement of PACS is the provision of an efficient search function to access the required images.

An universal format for PACS image storage and retrieval is the Digital Imaging and Communications in Medicine (DICOM) standard [3]. DICOM is a well known standard for handling, storing, printing, and transmitting information in medical imaging. The DICOM header contains tags to decode the body part examined, the patient position and the acquisition modality. Some of the tags are automatically set by the digital system according to the imaging protocol used to capture the pixel data. Other part of the tags are set manually by the physicians or radiologists during the routine documentation. This procedure cannot always be considered very reliable, since frequently happens that some entries are either missing, false, or do not describe the anatomic region precisely [4]. Furthermore, manual annotation of images is an expensive and time-consuming procedure, especially given the large and constantly growing databases of medical images. Thus, completely automated categorization in terms of DICOM tags is currently not possible, but is highly desirable.

Automatic image annotation or image classification is an important step in image retrieval. In the medical domain, using information directly extracted from images to annotate/categorize them will improve the quality of image annotation in particular, and more generally the quality of patient care. Properly classified medical image data can help medical professionals in fast and effective access to data in their teaching, research, training, and diagnostic problems. The results of the classification step can also be used for multilingual image annotation as well as for DICOM header correction [5].

Automatic image annotation can be used for retrospective annotation (pre DICOM). It can also be used as help for human annotators (i.e., radiologists), where the annotations that are suggested by the system are corrected/verified/confirmed by the human annotator. The limits of performance of an automated annotation system that learns from example images annotated by humans, is the rate/probability of operator error/agreement of annotators.

Automatic image annotation uses a computer system which automatically assigns metadata in the form of captions or keywords to a digital image. Typically, image analysis first extracts feature vectors. Then, together with the training annotations, they are used by a machine learning algorithm to learn to automatically assign annotations. The performance of the computer system largely depends on the availability of strongly representative visual features, able to characterize different visual properties of the images, and the use of effective algorithms for training classifiers for automatic image annotation.

A single image may contain different meanings organized in hierarchical semantics: hence, hierarchical multi-label classification (HMC) is strongly recommended for obtaining multi-label annotations. The task of multi-label classification is to assign multiple labels to each image. The assigned labels are a subset of a previously defined set or hierarchy of labels. HMC is used in various domains [6], such as text classification, scene and video classification, medical imaging and biological applications. One of the main issues involved in multi-label classification is the importance of detecting and incorporating the connections between the labels into the process of assigning multiple labels. A second and related issue is the additional complexity involved in learning multi-label classifiers, as compared to learning single-label classifiers.

In this paper, we present a HMC system for medical image annotation. This system consists of the two standard parts of image annotation systems, i.e., processing (feature extraction) and classification of images. The image processing part uses state-of-the-art approaches to convert an image to a set of numerical features extracted directly from the pixel values. The image classification part, which labels and groups the images, contains the main novelty of our approach: The labels can be organized in a hierarchy and an image can be labeled with more than one label (an image can belong to more than one group).

First, we generate four different types of descriptors suitable for X-Ray medical images: raw pixel representation (RPR) [7], local binary patterns (LBP) [8], edge histogram descriptors (EHD) [9], and scale-invariant feature transform (SIFT) [10]. The features are generated using the medical X-ray images from the ImageCLEF2009 medical image annotation task [5]. Next, we use these features together with the annotations to train the classifiers. In particular, we use ensembles (bags and random forests) of PCTs for HMC and SVMs for single-label classification, the most widely used classifier in the area of image annotation. At the end, we assess the predictive performance of the classifiers using the hierarchical error measure (HEM) from ImageCLEF [5] and overall recognition rate (RR), commonly used for assessing the predictive performance over the database we use.

The main question that we address in our research is whether exploiting the semantic knowledge about the inter-class relationships among the image labels (organized in a hierarchical structure) can improve the predictive performance of a system for automatic image annotation. To this end, we compare the predictive performance of the ensembles of PCTs for HMC (that predict all labels simultaneously) to that of SVMs (each of them predicting a single label). We do this across all feature extraction techniques, thus evaluating the different feature extraction techniques and their use in HMC of medical X-ray images. Moreover, we investigate whether (and which type of) combination of feature extraction techniques yields better predictive performance. We consider low level (LL) and high level (HL) feature fusion/combination schemes [7].

To emphasize the generality of our approach, we have also tested it on the database of general images from the ImageCLEF@ICPR 2010 photo annotation task [11]. The images in this database are annotated with 53 visual concepts organized in a classification scheme with hierarchical structure, which we used to build ensembles of PCTs for HMC as classifiers. The 53 concepts include abstract categories (like partylife), the time of day (like day or night), persons (like no person visible, small or big group) and quality (like blurred or underexposed). A complete overview of the task is given by Nowak [11].

The remainder of the paper is organized as follows. In Section 2, we give an overview of related work. Section 3 introduces predictive clustering trees and their use for HMC. Section 4 describes the techniques for feature extraction from images. In Section 5, we explain the experimental setup for annotating medical images. The obtained results and a discussion thereof are given in Section 6. Section 7 describes the experiments in annotation of general images, as well as their results. Section 8 concludes the paper and points out some directions for further work.

2. Related work

In this section, we present some classification methods that are or can be used for image annotation. We begin by presenting the methods that are most widely used by the image annotation community. We then present some recent machine learning methods that can be used for hierarchical image annotation and discuss their relation to the method we propose.

Regardless of the number of visual concepts that have to be learned and their mutual connections, most of the present systems for annotation of general images (and medical images in particular) learn a separate model for each visual concept (label), i.e., they treat the classes as completely separate and independent (both visually and semantically). This means that multi-label classification problems are transformed into several binary classification problems. For example, the methods with high predictive performance at recent challenges/competitions in detection and annotation tasks (such as the PASCAL Visual Object Classes challenge [12], the ImageCLEF medical image annotation task [13], [5] and the ImageCLEF visual concept detection and annotation task [14]) perform multi-label classification by building binary classifiers for each label. The instances associated with particular label are in one class and the rest are in another class. For solving the binary classification problems, is common to use a SVM with a χ^2 kernel [15]. This means that the increase of the number of labels used for annotation will linearly increase the complexity of such an approach.

To deal with a large number of labels/classes, many approaches combine binary classifiers using class hierarchies [16], [17]. This results in a logarithmic increase of complexity as the number of labels increases. The class hierarchies can be automatically constructed through analysis of visual similarities: this can proceed top-down by recursive partitioning of the set of classes [18] or bottom-up by agglomerative clustering [19]. The hierarchies could also be found by exhaustive search or random sampling followed by cross-validation [20].

An alternative method for automatic construction of hierarchies is to query an external semantic network with class labels [17]. Since semantic networks model concepts and relations between them, a subgraph in the form of a hierarchy can be easily extracted. Such an approach allows to incorporate prior knowledge about object identity into the visual recognition system. Our approach to automatic image annotation is based on this idea. We exploit the semantic knowledge about the inter-class relationships among the image labels organized in a hierarchical structure. We build one classifier that can simultaneously predict all annotations of an image, instead of building one binary classifier for each node in the hierarchy.

Another popular approach to image annotation is TagProp [21]. TagProp is a discriminatively trained nearest neighbor model. Tags of test images are predicted using a weighted nearest-neighbor model to exploit labeled training images. Neighbor weights are based on neighbor rank or distance. TagProp allows the integration of metric learning by directly maximizing the log-likelihood of the tag predictions in the training set. However, in a recent study, Mensink et al.[22] showed that per-label-trained linear SVM classifiers outperform TagProp.

So far, we presented the most widely used methods for image annotation and concluded that SVMs with a χ^2 kernel trained per label are the preferred method by the image annotation community. In the remainder of this section, we discuss recent machine learning methods that can be used in the context of hierarchical image annotation: SVMs for structured prediction, PCTs, ensembles of PCTs and ensembles of SVMs. To begin with, SVMs for predicting structured outputs can be considered as classifiers for hierarchical image annotation. Unfortunately, the most well-known system for predicting structured outputs based on SVMs, SVMstruct [23], does not offer facilities for HMC. Those that do are very recent [24], have high computational complexity and are not used by the image annotation community.

We can also apply PCTs for HMC to the task of hierarchical image annotation. Vens et al. [25] describe in detail PCTs that are able to perform hierarchical multi-label classification and perform extensive experimental evaluation on functional genomics datasets. They show that PCTs for HMC achieve very good predictive performance and are very efficient.

Ensemble methods are a popular approach that generates a set of classifiers (called base classifiers) and combine their predictions into a single prediction [26]. Many practical and theoretical studies show that ensembles achieve high predictive performance and lift the predictive performance of a single classifier [27, 28]. This is especially true for base classifiers that are unstable, i.e.,can change drastically due to small changes in the training data: Decision trees are typical example of unstable classifiers. Having this in mind, we extend the PCT framework in the context of ensemble learning, i.e., we construct ensembles of PCTs for HMC. We apply this approach to hierarchical image annotation: The ensembles of PCTs for HMC achieve better predictive performance than a single PCT and can be constructed efficiently.

Given that SVMs are the most widely used machine learning approach to image annotation, and that ensembles improve the performance of individual classifiers, one might also consider ensembles of SVMs. However, SVMs are relatively stable classifiers and less likely to benefit from an ensemble extension. Consequently, there is much less community consensus on whether and how ensembles of SVMs should be constructed (as compared to ensembles of decision trees). Evgeniou et al. [29] performed theoretical and empirical evaluation of ensembles from SVMs. The main finding in their study is that a single SVM classifier with tunned parameters performs similar to an ensemble of SVM classifiers. On the other hand, Valentini and Dietterich [30] and Wang et al. [31] show that ensembles of SVMs do lift the predictive performance of a single SVM. They also discuss practical issues in constructing such an ensemble and obtaining a prediction for unseen instance. A recent study by Ting and Zhu [32] proposed a boosting algorithm that uses a hybrid between decision trees and SVMs as base classifier. Their findings reveal that such an ensemble has better predictive performance than a single SVM and it is efficient. However, all these studies were performed in the context of binary or multi-class classification and their extension for the task of HMC is not straightforward. The base classifiers would need to be either SVMs for HMC or collections of SVMs as discussed above. In addition, ensembles of SVMs are not in widespread use in the image annotation community.

3. Ensembles of PCTs for HMC

This section presents our approach for building ensembles of PCTs. We first present the task of HMC. Next, we describe the predictive clustering trees and their instantiation for the task of HMC. Finally, we present ensembles of PCTs for HMC and methods for building them.

The development of this approach is motivated by the fact that ensembles lift the predictive performance of a single predictive model. This is well known in the case when the single predictive model is a classification or a regression tree. However, it is not obvious that the lift carries over to PCTs for HMC. When the base classifiers are decision trees, Bauer and Kohavi [33] conclude that the increase in performance is related to the trees being unpruned, i.e., overfitting. On the other hand, Blockeel et al. [34] state that PCTs for HMC overfit less as compared to individual trees for each class in the hierarchy. Having in mind these two conflicting influences, it is not obvious whether an ensemble of PCTs will significantly increase the predictive performance of a single PCT. Hence, this is an interesting issue to investigate. A further motivation for our study is provided by the fact that PCTs for HMC (and potentially ensembles thereof) are efficient to construct and perform well, yet their use in the context of hierarchical image annotation has so far not been investigated.

3.1. The task of HMC

Hierarchical multi-label classification is a variant of classification where (1) a single example may belong to multiple classes at the same time and (2) the possible classes are organized in a hierarchy. An example that belongs to some class c automatically belongs to all super-classes of *c*: This is called the hierarchical constraint. Problems of this kind can be found in many domains including text classification, functional genomics, and object/scene classification. For a more detailed overview of the possible application areas we refer the reader to Silla and Freitas[6].

image	features/descriptors						annotations/labels
	|	—	/	\	Ş		
	48	24	59	66	37	…	cervical spine@ musculoskeletal system
	36	25	53	45	15	…	middle abdomen@renal pelvis
	35	25	56	52	19		lumbar spine@ musculoskeletal system
…	…	…	…	…	…	…	…

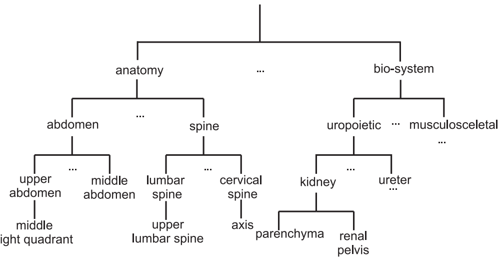

Figure 1: An example task of HMC in a medical domain. The table (on the left-hand side) contains a set of images with their visual descriptors and annotations. The annotations are part of the IRMA [35] hierarchical classification scheme (of which a small part is shown on the right hand side).

In medical image classification, the application domain on which we focus, an important problem is the development of an automatic image annotation system, which can specify the image modality, body orientation, body region, or the biological system examined. In this domain, the predefined set of labels might be organized in a semantic hierarchy, such as the one shown in Fig. 1. Each image is represented with: (1) a set of descriptors (in this example, the descriptors are histograms of five types of edges encountered in the image) and (2) a set of labels/annotations. A single image can be annotated with multiple labels at different levels of the predefined hierarchy.

For example, the image in the second row of the table in Fig. 1 has two labels, middle abdomen and renal pelvis, listed explicitly. Note that this image is also implicitly labeled with the labels: anatomy, abdomen, kidney, uropoietic and bio-system. These labels are all ancestors of the explicitly listed labels in the given hierarchy.

The data, as presented in the table in the left-hand side of Fig. 1, constitute a data set for HMC. This set can be used by a machine learning algorithm to train a classifier for HMC. For images in the testing set only the descriptors are given and no *a priori* annotations.

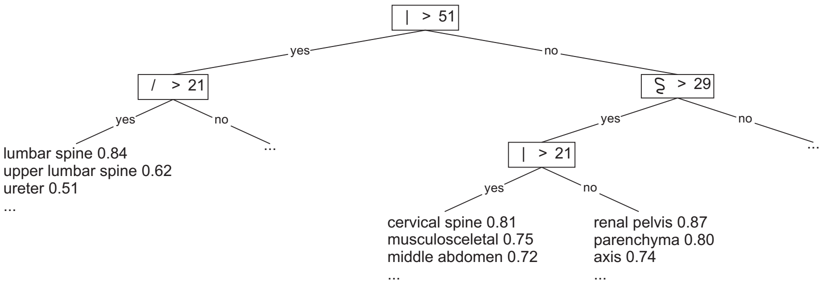

Figure 2: An example of a predictive clustering tree constructed using the descriptors from Fig. 1. The internal nodes contain tests on the descriptors, while the leafs store the probabilities that an image is annotated with a given label from the hierarchy.

3.2. Predictive clustering trees

Predictive Clustering Trees (PCTs) [36] [1] generalize decision trees [37] and can be used for a variety of learning tasks including different types of prediction and clustering. The PCT framework views a decision tree as a hierarchy of clusters: the top-node of a PCT corresponds to one cluster containing all data, which is recursively partitioned into smaller clusters while moving down the tree. The leaves represent the clusters at the lowest level of the hierarchy and each leaf is labeled with its cluster's prototype (prediction). Note that the hierarchical structure of the PCT (Fig. 2) does not necessary reflect the hierarchical structure of the annotations (Fig. 1).

PCTs are built with a greedy recursive top-down induction (TDI) algorithm, similar to that of C4.5 [38] or CART [37]. The learning algorithm starts by selecting a test for the root node. Based on this test, the training set is partitioned into subsets according to the test outcome. This is recursively repeated to construct the subtrees. The partitioning process stops when a stopping criterion is satisfied (e.g., the number of records in the induced subsets is smaller than some predefined value; the length of the path from the root to the current subset exceeds some predefined value etc.). In that case, the prototype is calculated and stored in a leaf.

One of the most important steps in the TDI algorithm is the test selection procedure. For each node, a test is selected by using a heuristic function computed on the training examples. The goal of the heuristic is to guide the algorithm towards small trees with good predictive performance. The heuristic used in this algorithm for selecting the attribute tests in the internal nodes is the reduction in variance caused by partitioning the instances, where the variance $Var(S)$ is defined by (Equation 1). Maximizing the variance reduction maximizes cluster homogeneity and improves predictive performance.

The main difference between the algorithm for learning PCTs and an algorithm for learning decision trees (such as C4.5 [38] and CART [37]) 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 PCTs have been instantiated for the following tasks: multiple targets prediction [39], [40], prediction of time-series [41] and hierarchical-multi label classification [25]. In this article, we focus on the last of these tasks.

3.3. PCTs for hierarchical multi-label classification

To apply PCTs to the task of HMC, the example labels are represented as vectors with Boolean components. Components in the vector correspond to labels in the hierarchy traversed in a depth-first manner. The i-th component of the vector is 1 if the example belongs to class c_i and 0 otherwise. If $v_i = 1$, then $v_j = 1$ for all v_j's on the path from the root to v_i.

The variance of a set of examples (S) is defined as the average squared distance between each example's label v_i and the mean label $\bar{v}$ of the set, i.e.,

$$Var(S) = \frac{\sum\limits_i d(v_i, \bar{v})^2}{|S|} \tag{1}$$

[1] The PCT framework is implemented in the CLUS system, which is available at http://www.cs.kuleuven.be/~dtai/clus.

The higher levels of the hierarchy are more important: an error at the upper levels costs more than an error at the lower levels. Considering this, a weighted Euclidean distance is used:

$$d(v_1, v_2) = \sqrt{\sum_i w(c_i)(v_{1,i} - v_{2,i})^2} \tag{2}$$

where $v_{k,i}$ is the i'th component of the class vector v_k of an instance x_k, and $w(c_i)$ are the class weights. The class weights decrease with the depth of the class in the hierarchy, $w(c_i) = w_0 \cdot w(c_j)$, where c_j is the parent of c_i. Each leaf in the tree stores the mean $\bar{v}$ of the vectors of the examples that are sorted into that leaf (Fig. 2). Each component of $\bar{v}$ is the proportion of examples $\bar{v}_i$ in the leaf that belong to class c_i. An example arriving in the leaf can be predicted to belong to class c_i if $\bar{v}_i$ is above some threshold t_i. The threshold can be chosen by a domain expert.

The PCTs are also extended for predicting hierarchies organized as directed acyclic graphs (DAGs). In this case, the depth of a class is not unique as classes do not have single path from the hierarchy's root. To resolve this issue, Vens et al. [25] suggest four aggregation schemes of the possible paths from the top-node to a given class: average, maximum, minimum and sum. After an extensive experimental evaluation, they recommend to use the average as aggregation function. For a detailed description of PCTs for HMC we refer the reader to Vens et al. [25]. Next, we explain how PCTs are used in the context of an ensemble classifier, in order to further improve the performance of PCTs.

3.4. Ensemble methods

An ensemble is a set of (base) classifiers. A new example is classified by the ensemble by combining the predictions of the member classifiers. The predictions can be combined by taking the average (for regression tasks), the majority vote (for classification tasks) [42],[43], or more complex combinations.

We use PCTs for HMC as base classifiers. Averaging is applied to combine the predictions of the different trees: the leaf's prototype is the proportion of examples of different classes that belong to it. Just like for the base classifiers, a threshold should be specified to make a prediction.

We consider two ensemble learning techniques that have primarily been used in the context of decision trees: bagging and random forests. Bagging [42] constructs the different classifiers by making bootstrap replicates of the training set and using each of these replicates to construct one classifier. Each bootstrap sample is obtained by randomly sampling training instances, with replacement, from the original training set, until a number of instances is obtained equal to the size of the training set. Bagging is applicable to any type of learning algorithm.

A random forest [43] is an ensemble of trees, obtained both by bootstrap sampling, and by 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., $f(x) = x$, $f(x) = \sqrt{x}$, $f(x) = \lfloor \log_2 x \rfloor + 1$, ...). By setting $f(x) = x$, we obtain the bagging procedure.

4. Feature extraction from images

Collections of medical images can contain various images obtained using different imaging techniques. Different feature extraction techniques are able to capture different aspects of an image (e.g., texture, shapes, color distribution...). In this study, we focus on X-ray images, hence, we use texture (LBP and EHD) and local (SIFT) features as most promising for describing X-ray images [5],[44].

Texture is especially important, because it is difficult to classify medical images using shape or gray level information. Effective representation of texture is needed to distinguish between images with equal modality and layout. Local image characteristics are fundamental for image interpretation: while global features retain information on the whole image, the local features capture the details. They are thus more discriminative concerning the problem of inter and intra-class variability, an open challenge in automatic annotation of medical images [7].

4.1. Raw pixel representation

The most straightforward approach to image classification is the direct use of the image pixel values as features. The images are scaled to a common size and represented by a feature vector that contains image pixel values. It has been shown that for classification and retrieval of medical radiographs, this method serves as a reasonable baseline [45]. We used a 32x32 down-sampled representation of the images as recommended by Tommasi et al. [7]. The obtained 1024 pixel values were then used as input features. Fig. 3 shows how we built the raw pixel representation for each image.

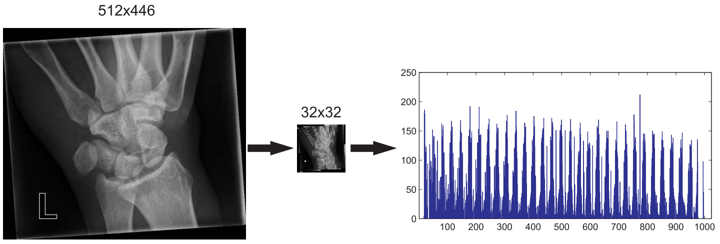

Figure 3: Down-sampling for raw pixel representation

4.2. Local binary patterns

Local binary patterns (LBP) are one of the best representations of texture content in images [8]. They are invariant to monotonic changes in gray-scale images and fast to compute. Furthermore, they are able to detect different micro patterns, such as edges, points and constant areas.

The basic idea behind the LBP approach is to use the information about the texture from a local neighborhood. First, we define the radius R of the local neighborhood under consideration. The LBP operator then builds a binary code that describes the local texture pattern in the neighborhood set of P pixels. The binary code is obtained by applying the gray value of the neighborhood center as a threshold. The binary code is then converted to a decimal number which represents the LBP code. Formally, given a pixel at position (x_c, y_c) the resulting LBP code can be expressed as follows:

$$LPB_{(P,R)}(x_c, y_c) = \sum_{n=0}^{P-1} S(i_n - i_c)2^n \tag{3}$$

where n ranges over the P neighbors of the central pixel (x_c, y_c), i_c and i_n are the gray-level values of the central pixel and the neighbor pixel, and $S(x)$ is defined as:

$$S(x) = \begin{cases} 1, & \text{if } x \geq 0 \quad (4a) \\ 0, & \text{otherwise} \quad (4b) \end{cases}$$

The image is traversed with the LBP operator pixel by pixel and the outputs are accumulated into a discrete histogram. However, not all LBP codes are informative. Certain LBP codes capture fundamental properties of the texture and are called uniform patterns because they constitute the vast majority, sometimes over 90 percent, of all patterns present in the observed textures [8]. These patterns have one thing in common, namely, a uniform circular structure that contains very few spatial transitions. They function as templates for micro-structures such as bright spot, flat area or dark spot.

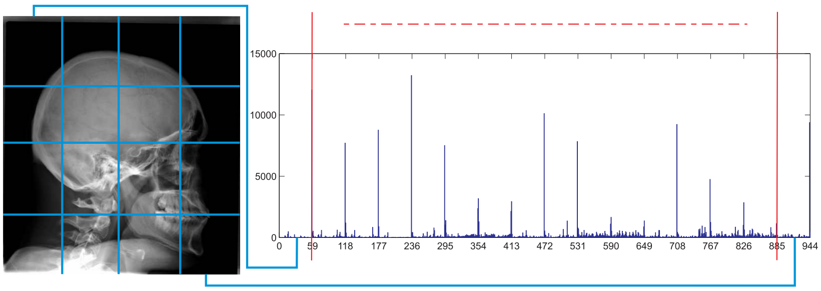

Figure 4: The image is divided into 4x4 non-overlapping sub-images from which LBP histograms are extracted and concatenated into a single, spatially enhanced histogram

In our experiments, we used the patterns $LBP^{u2}_{8,1}$, where the superscript $u2$ reflects the use of uniform patterns that have a U value of at most 2 on a neighborhood of size 8 and radius 1. The U value is the number of spatial transitions (bitwise 0/1 changes) in the pattern. The non-uniform patterns (patterns that have U value larger than 2) are grouped under one bin in the resulting histogram. With the $LBP^{u2}_{8,1}$ operator, the number of bins in the histogram is reduced from 256 to 59 (58 bins for uniform patterns and one bin for non-uniform/noisy patterns).

To spatially enhance the descriptors and improve the performance, it has been suggested to repeatedly sample predefined sub-regions of an image (e.g., 1x1, 2x2, 4x4 or 1x3) [46]. The different resolutions are then aggregated into a spatial pyramid which allows for region-specific weighting. Following these approaches, we divide the images into 4x4 non-overlapping sub-images (blocks) and concatenate the LBP histograms extracted for each sub-image into a single, spatially enhanced feature histogram. This approach aims at obtaining a more local description of the images. Fig. 4 shows how we build the LBP histogram with 944 bins in total for each image (16 blocks with 59 bins each).

4.3. Edge histogram descriptors

Edge detection is a fundamental problem of computer vision and has been widely investigated [47]. The goal of edge detection is to mark the points in a digital image at which the luminous intensity changes sharply. An edge representation of an image drastically reduces the amount of

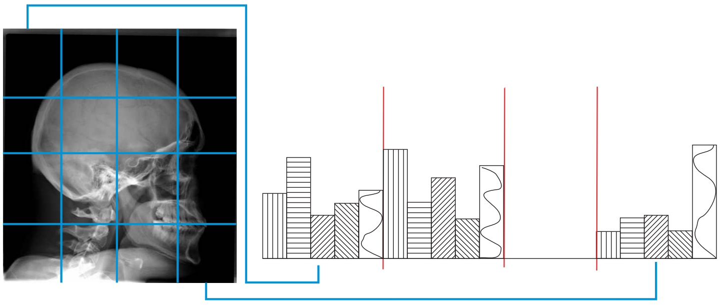

Figure 5: The image is divided into 4x4 non-overlapping sub-images. For each sub-image, five types of edge bins are calculated and concatenated into a single, spatially enhanced histogram

data to be processed, yet it retains important information about the shapes of objects in the scene. Edges in images constitute important features to represent their content.

The edge histogram in the image space represents the frequency and the directionality of the brightness changes in the image. To represent it, the MPEG-7 standard defines the edge histogram descriptor (EHD) [9]. The edge histogram descriptor basically represents the distribution of five types of edges (vertical, horizontal, two types of diagonal and non-directional edges; see Fig. 2). We divide the image space into 4x4 non-overlapping blocks, yielding 16 equal-sized sub-images and count the edges on each one of them (as shown in Fig. 5).

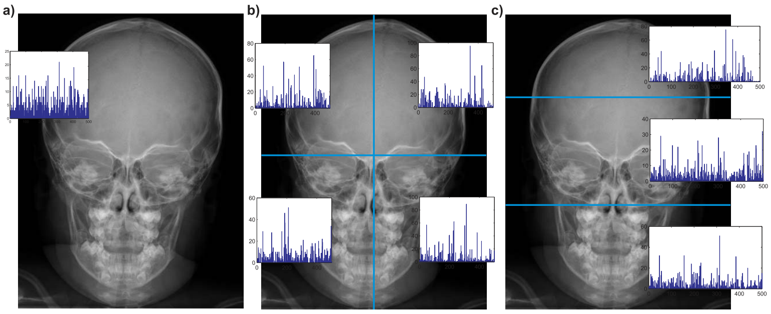

Figure 6: Three different spatial pyramids used in our experiments, a) 1x1, b) 2x2 and c) 1x3. The spatial pyramid constructs feature vectors for each of the specific part of the image.

To characterize the sub-images, a histogram of edge distribution for each sub-image is generated. Edges in the sub-images are categorized into five types: vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges, as presented in Fig. 5. The histogram for each sub-image represents the relative frequency of occurrence of the five types of edges in the corresponding sub-image and thus contains five bins.

Since there are 16 sub-images in the image and 5 types of edges, a total of 80 histogram bins are required. Note that each of the 80-histogram bins has its own semantics in terms of location and edge type. In our experiments, the edge detection is performed using the Canny edge detection algorithm [48].

4.4. SIFT descriptors

We employ the bag of features approach commonly used in many state of the art approaches in image classification [49]. The basic idea of this approach is to sample a set of local image patches using some method (densely, randomly or using a key-point detector) and calculate a visual descriptor on each patch (SIFT descriptor, normalized pixel values). The resulting distribution of descriptors is then quantified against a pre-specified visual codebook which converts it to a histogram. The main issues that need to be considered when applying this approach are: sampling of the patches, selection of the visual patch descriptor and building the visual codebook.

We use dense sampling of the patches, which samples an image grid in a uniform fashion using a fixed pixel interval between patches. We use an interval distance of 6 pixels and sample at multiple scales ($\sigma = 1.2$ and $\sigma = 2.0$). Due to the low contrast of the radiographs, it would be difficult to use any detector for points of interest. Also, it has been pointed by Zhang et al. [49], that a dense sampling is always superior to any strategy based on detectors for points of interest. We calculate a SIFT descriptor [10] for each image patch.

The crucial aspects of a codebook representation are the codebook construction and assignment. An extensive comparison of codebook representation variables is given by van Gemert et al. [50]. We employ k-means clustering (as implemented in the *R* environment) [51] on 400000 randomly chosen descriptors from the set of images available for training. k-means partitions the visual feature space by minimizing the variance between a predefined number of k clusters. Here, we set k to 500 and thus define a codebook with 500 codewords [7].

Dense sampling gives an equal weight to all key-points, irrespective of their spatial location in the image. To overcome this limitation, we follow the spatial pyramid approach which we applied for the LBP descriptor. For this descriptor, we used a spatial pyramid of 1x1, 2x2, and 1x3 regions. Since every region is an image in itself, the spatial pyramid can easily be used in combination with dense sampling. The resulting vector with 4000 bins ((1x1 + 2x2 + 1x3)x500) was obtained by concatenation of the eight histograms. Fig. 6 shows an example of the histograms extracted from an image for the spatial pyramids of 1x1, 2x2 and 1x3.

4.5. Feature fusion schemes

Different visual features bringing different information about the visual content of the images clearly outperform single feature approaches [5], [7]. Following these findings, we combine the different visual features described above. We investigate two different feature fusion schemes: low level (LL) and high level (HL). These fusion schemes are depicted in Fig. 7.

For the low level feature fusion scheme, the descriptors are concatenated in a single feature vector and a classifier is trained on the joint feature vector. The high level fusion scheme averages the predictions from the individual classifiers trained on the separate descriptors.

5. Experimental setup

In this section, we present the experimental setup we used to evaluate the proposed system and compare it to other approaches. First, we present the databases of images that we use. Next, we describe the evaluation metrics we use to assess the predictive performance of the classifiers. We then state the experimental questions that we investigate in this study. We specify the parameter instantiations for the algorithms and the design of the experiments.

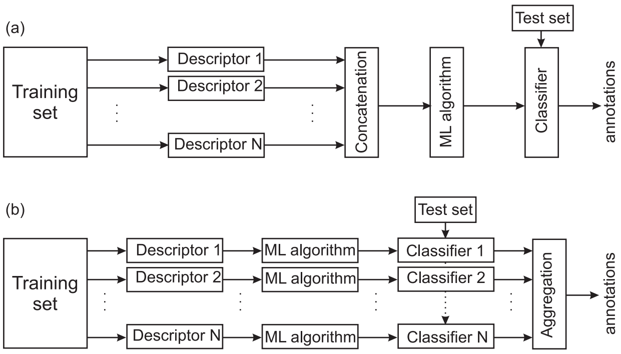

Figure 7: Fusion schemes for the different descriptors. a) Low level fusion, b) High level fusion.

5.1. IRMA database

We evaluated our system by applying it to the database for the ImageCLEF2009 medical image annotations task [5]. This database is provided by the IRMA group from the University Hospital of Aachen, Germany [35]. The database contains 12677 fully annotated radiographs, taken randomly from medical routine, which should be used to train a classifier. The dataset contains two parts: ImageCLEF2007 (12339 training and 1353 testing images) and ImageCLEF2008 (12667 training and 1733 testing images). These datasets present a difficult classification problem. First, the classes in the training set are extremely imbalanced (e.g. there are classes with less than 10 images and classes with more than 2000 images). Second, the distribution of the classes in the training set is different from the one on the testing set.

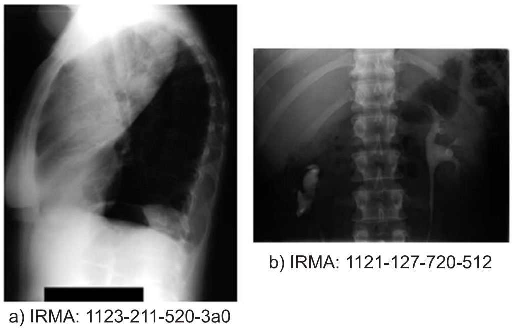

Figure 8: IRMA-coded chest and abdomen radiograph. For instance, the code for the biological axis (512) on the subfigure b) is translated as follows: 5 is for uropoietic system, 51 is for uropoietic system, kidney and 512 is uropoietic system, kidney, renal pelvis. The renal pelvis is an element of the kidney, which in turn is an element of the uropoietic system

The images are labeled according to the four annotation label sets [5]. We used the ImageCLEF2007 label set with 116 IRMA codes and the ImageCLEF2008 label set with 193 IRMA codes, both with a hierarchical nature of the coding scheme [35]. The goal is to correctly annotate 1353 (for 2007) and 1733 (for 2008) images that are provided without labels, using the different respective annotation label sets in turn.

The IRMA coding scheme consists of four axes with three to four positions, each position taking a value from the set 0,..., 9, a,..., z, where '0' denotes 'unspecified' and determines the end of a path along an axis. The four axes are: technical axis (T, image modality), directional axis (D, body orientation), anatomical axis (A, body region examined) and biological axis (B, biological system examined). This allows a short and unambiguous notation (IRMA: TTTT-DDD-AAA-BBB), where T, D, A, and B denotes a coding or sub-coding digit of the respective axis. A small part of the IRMA coding hierarchy is presented in Fig. 1. Fig. 8 gives two examples of unambiguous image classification using the IRMA code.

The IRMA code is hierarchical in its nature and it allows us to exploit the hierarchy of the code. This means that we can construct an automatic image annotation system based on predictive clustering trees for HMC.

5.2. Evaluation metrics

In this study, we use two evaluation metrics: the ImageCLEF hierarchical evaluation measure [5] and overall recognition rate. The ImageCLEF hierarchical evaluation measure takes into account the depth and the difficulty of the predictive problem ('branching factor') at which an error has occurred (Equation 5). It can be calculated using the following formula:

$$\sum_{i=1}^{I} \frac{1}{b_i} \frac{1}{i} \delta(v_i, \bar{v_i}), \tag{5}$$

$$\delta(v_i, \bar{v_i}) = \begin{cases} 0, & \text{if } v_j = \bar{v_j} \forall j \leq i & \text{(6a)} \\ 0.5, & \text{if } v_j = * \exists j \leq i & \text{(6b)} \\ 1, & \text{if } v_j \neq \bar{v_j} \exists j \leq i & \text{(6c)} \end{cases}$$

where I is the depth of the hierarchy, b_i is the number of possible labels at the error ('branching factor') and i is the depth at which the error occurred. This measure allows the classifier not to predict the complete code/annotation, that is, the classifier can predict the first 2 nodes of the code (level of the hierarchy) and then say 'don't know' (encoded by *) for the next node/level. The ImageCLEF evaluation measure can range from 0 to the number of testing images. If this measure is closer to 0, then the classifier is more accurate.

The overall recognition rate is a very common and widely used evaluation measure. It is the fraction of the test images whose complete IRMA code was predicted correctly.

5.3. Experimental questions

The goal of this study is to answer the following questions:

1. Does the use of the hierarchy (in ensembles of PCTs) improve the predictive performance over flat classification (SVMs)?
2. How is the relative performance of the two techniques affected by the:
    (a) Use of PCT ensembles versus single PCTs in the domain of image annotation?
    (b) Different ensemble methods: bagging or random forests?
    (c) Different feature extraction techniques for medical X-Ray images?
    (d) Schemes for fusion of the descriptors from the feature extraction techniques?
3. Is the proposed system with ensembles of PCTs for HMC scalable and efficient?

For the first three questions (1, 2a and 2b), we evaluate the performance of PCTs for HMC and ensembles (bagging and random forest) of PCTs. After that, we compare the best method for HMC with SVMs. It has been shown [25] that exploiting the structure of the hierarchy in tree classifiers yields better predictive performance in the domain of functional genomics. Here, we compare the performance of the ensemble classifiers with SVMs for flat classification - the most widely used classifiers for medical image annotation [7].

To check which feature extraction technique is most suitable for medical X-Ray images (question 2c), we compare the performance of the classifiers on each type of visual descriptors. For this purpose, we discuss only the results from the separate runs of the descriptors (first four rows from Table 1 and Table 2).

The various feature extraction techniques capture different aspects of an image. We also investigate whether the combination of feature extraction techniques can increase the predictive performance (question 2d). The results from the fusion schemes are presented in the last 10 rows in Table 1 and Table 2.

We compare the execution times of the different classifiers to assess the efficiency and scalability of the system (question 3). We measure the time needed to train the classifiers; for SVMs this includes also the time needed to optimize the parameters.

5.4. Experimental design

In this section, we describe the experimental setup that we used. First, we describe an adaptation of the hierarchy of the IRMA code and then the parameter instantiations of the learning algorithms. Note that we stated the parameters for the feature extraction techniques while explaining them (see Section 4).

The IRMA coding scheme was proposed by Lehmann et al. in [35]: It consists of four axes which are strictly hierarchical (tree-shaped hierarchies). The literature [5],[35] suggests that these four axes are independent. We conducted a series of experiments predicting the four axes simultaneously (combined in a single hierarchy) and separately. The predictive performance when using all four axes simultaneously was higher as compared to using each axis separately. This leads us to believe that these axes are not-independent. In a separate study, Tommasi et al. [7] come to a similar conclusion. To address this issue, we adapted the IRMA coding hierarchy as follows.

We take the code of the first position for the biological axis and add it in front of the codes for the anatomical and directional axes. The inclusion of the biological code in the first level in the hierarchy helps us to initially filter the images resulting in large visual differences in the first level of the hierarchy. In the context of the axis A, the first level of axis B is necessary because the examined body region insufficiently describes the content and structure of the images. For example, fluoroscopy of the abdominal region may access the vascular or the gastrointestinal system depending on the way the contrast agent is administered, which results in different image textures. For the directional axis, this is even more obvious. For instance, an image of a chest and an image of a hand can have the same directional code, but are visually very different.

The hierarchy of the IRMA code was adapted in order to increase the inter-class variability and decrease the intra-class variability of the images. Fig. 9 shows the adapted hierarchy of classes that we use in the experiments. Note that this hierarchy was only used to train the classifier. The evaluation was performed by using the original IRMA hierarchy.

In the following, we state the parameter instantiations that we used to train the classifiers: PCTs, ensembles and SVMs. The algorithm for learning PCTs requires as input the weight of

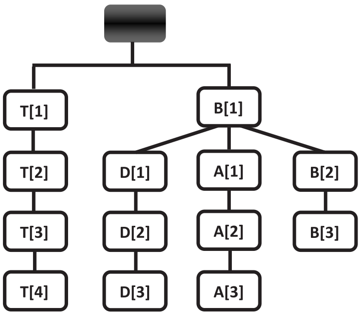

Figure 9: The adapted hierarchy of the classes in the IRMA code

the depth in the hierarchy. We set w_0 to 0.75 to force the algorithm to make better predictions on the upper levels of the hierarchy. Also, we performed F-test pruning to prevent over-fitting of the trees [25].

We trained ensembles of 100 un-pruned trees (PCTs). For the base PCTs, we used the same weight (0.75) used to train the single PCTs. The size of the feature subset that is retained at each node, when training a random forest, was set to 10% of the number of descriptive attributes. Remember that the output of the classifier is a probability that a given example is annotated with a given label. If the probability is higher than a given threshold (obtained during the training of the classifier), then the example is annotated with the given label. Since the hierarchical evaluation measure allows the classifier to predict a portion of the code, different thresholds for the different levels of the hierarchy were selected. If a probability for a given code was lower than the threshold, then for this code and its sub-codes the classifier predicts 'don't know'.

For training the SVMs, we used a custom developed application . This application uses the LibSVM library [52]. We apply the *One-against-All* (OvA) approach to solve the partial binary classification problems. Each of the SVMs was trained with a χ^2 kernel. We optimize the cost parameter C of the SVMs using an automated parameter search procedure. For the parameter optimization, we separate 20% of the training set and use it as validation set. After finding the optimal C value, the SVM was finally trained on the whole set of training images.

For the evaluation of the SVMs using the hierarchical error measure, we applied confidence based opinion fusion [7]. Let us assume that there are N classes. Then, using the OvA approach, N SVMs are trained – each separating a single class from all remaining ones. The decision is based on the distances of the test sample to the N hyperplanes. The prediction then corresponds to the hyperplane for which the distance is largest. The confidence based opinion fusion, however, takes into account the difference of the predictions with the two largest distances reported from the SVMs classifiers. This difference is computed only if their distances differ less than a threshold value (obtained during training using the validation data set). In that case, the final prediction will contain 'don't know' starting from the position where the two underlying predictions begin to differ. For example, if the two predictions for the anatomical axis are 411 and 421 then the final prediction will be 4**. This approach improves the hierarchical error measure for the SVMs classifier by 10 to 20 points depending on the used descriptors.

6. Results and discussion

Table 1 and Table 2 present the results obtained using the experimental setup described in Section 5 in terms of the hierarchical evaluation measure (HEM) and overall recognition rate (RR) respectively. In the discussion of the results, we first compare the performance of single PCTs and ensembles of PCTs. We then compare the performance of the best ensemble method (random forests) and SVMs. We focus on the first evaluation measure HEM (Table 1), since the two show similar behavior; the conclusions for HEM are also valid for RR.

Table 1: Predictive performance of the models learned from descriptors produced by different feature extraction algorithms and their combinations. The best results are shown in boldface. Performance is given in terms of the ImageCLEF hierarchical evaluation measure HEM, where smaller values mean better performance. The low-level fusion results are in rows that end with 'LL' and high-level fusion results are in rows that end with 'HH'.

	Hierarchical Error Measure							
	ImageCLEF2007				**ImageCLEF2008**			
	SVM	RF	Bag	PCTs	SVM	RF	Bag	PCTs
SIFT	75.00	**58.90**	59.78	180.00	179.88	**161.67**	161.47	320.90
LBP	124.44	95.71	95.71	210.40	257.92	209.47	208.97	360.00
EHD	127.41	105.12	105.12	222.39	265.95	249.44	249.74	380.12
32x32	202.94	195.78	200.12	310.90	376.93	361.21	361.31	530.11
LBP+EHD_LL	99.48	85.56	86.80	200.12	221.96	190.12	190.22	347.89
LBP+SIFT_LL	72.71	52.89	53.22	178.29	175.65	157.38	157.48	317.12
EHD+SIFT_LL	72.37	56.11	57.11	179.12	170.97	159.30	159.33	318.87
LBP+EHD+SIFT_LL	70.45	**51.90**	52.33	177.23	170.87	**153.21**	153.41	317.00
LBP+EHD+SIFT+32x32_LL	69.46	52.23	53.00	178.12	169.11	154.23	154.63	318.50
LBP+EHD_HL	100.37	87.90	89.21	201.30	223.73	195.96	196.06	347.90
LBP+SIFT_HL	73.72	54.21	54.56	178.90	177.12	159.73	160.03	318.00
EHD+SIFT_HL	72.70	59.12	61.71	179.50	174.44	161.85	162.05	318.80
LBP+EHD+SIFT_HL	71.58	52.54	53.00	177.90	174.18	156.21	156.31	317.90
LBP+EHD+SIFT+32x32_HL	70.46	53.90	54.50	178.58	173.28	156.50	156.70	318.30

The results clearly show that ensemble methods outperform single PCTs on all datasets: random forests are significantly better (according to the non-parametric Wilcoxon test for statistical significance) than single PCTs ($p < 4\cdot10^{-6}$) and bagging is better than single PCTs ($p < 4\cdot10^{-6}$). A comparison between the two ensemble methods shows that random forests outperforms bagging and that the difference is statistically significant ($p < 1 \cdot 10^{-4}$).

While extremely efficient, individual PCTs have the drawback of only using a small number of the available features, which results in low predictive performance. The PCTs trade predictive performance for interpretability. However, in the domains where interpretability of the model is a necessity, PCTs are the models that should be considered.

We next compare the performance of random forests to the performance of SVMs. On all datasets, random forests perform better than SVMs; the difference on average is $\sim$ 17 points for the ImageCLEF2007 and $\sim$ 20 points for ImageCLEF2008 datasets (note that a point in the HEM roughly corresponds to one completely misclassified image). The difference in performance is statistically significant (with $p < 4 \cdot 10^{-6}$). This shows that exploiting the structure of the hierarchy does help in improving the predictive performance.

We then analyze the results for the individual feature extraction algorithms (top 4 rows from Table 1 and Table 2). We can note the high predictive performance of the SIFT histogram: it is

Table 2: Predictive performance of the models learned from descriptors produced by different feature extraction algorithms and their combinations. The best results are shown in boldface. Performance is given in terms of the overall recognition rate evaluation measure, where larger values mean better performance. The low-level fusion results are in rows that end with 'LL' and high-level fusion results are in rows that end with 'HH'.

	Overall Recognition Rate							
	ImageCLEF2007				**ImageCLEF2008**			
	SVM	RF	Bag	PCTs	SVM	RF	Bag	PCTs
SIFT	77.31	**79.37**	79.08	63.04	62.44	**64.91**	64.80	52.04
LBP	70.36	75.24	75.24	56.02	56.26	60.99	60.70	47.02
EHD	68.37	72.28	72.21	55.06	54.53	54.99	54.81	45.00
32x32	57.35	58.01	57.64	45.97	45.47	45.52	45.47	36.98
LBP+EHD_LL	75.09	76.97	75.75	58.98	60.53	61.51	61.39	48.99
LBP+SIFT_LL	77.90	81.00	80.93	64.52	62.26	65.49	65.43	53.49
EHD+SIFT_LL	78.20	79.97	79.82	64.00	63.19	64.97	64.80	52.97
LBP+EHD+SIFT_LL	78.42	**81.96**	81.67	64.89	63.30	**65.95**	65.83	53.72
LBP+EHD+SIFT+32x32_LL	78.49	81.22	81.00	64.30	63.53	65.78	65.55	52.97
LBP+EHD_HL	74.87	76.01	76.64	58.38	60.13	61.45	61.39	48.87
LBP+SIFT_HL	77.46	79.97	79.97	64.22	62.26	65.32	65.14	53.49
EHD+SIFT_HL	77.90	79.00	78.86	63.93	62.44	64.80	64.62	52.79
LBP+EHD+SIFT_HL	78.05	81.00	80.93	64.59	62.78	65.78	65.72	53.66
LBP+EHD+SIFT+32x32_HL	78.42	80.70	80.56	64.37	63.13	65.60	65.49	52.97

most capable of capturing the hierarchical structure of the X-ray images. The other feature extraction algorithms follow after and are ordered by performance as follows: LBP, then EHD and the simplest descriptor RPR, which has the worst performance. The difference of performance to the LBP operator is very noticeable and larger for SVMs than for random forests: on the ImageCLEF2007 dataset, random forests are better by ∼ 30 points and on ImageCLEF2008 by ∼ 50 points and on the ImageCLEF2007 dataset, SVMs are better by ∼ 50 and on ImageCLEF2008 by ∼ 80 points. The LBP descriptors capture information that is more easily utilized by the random forests than by the SVMs.

The experimental results show that the features that describe the image content in a local manner (i.e., SIFT descriptors) outperform the ones that provide global descriptions. The local features capture the details in an image, while the global features are able to retain information on the whole image as a source of context. Furthermore, the SIFT descriptor is robust to noise, illumination, scale, translation and rotation changes. Hence, it can better resolve the inter and intra-class variability, thus it can offer better information to the classifier. We can conclude that the local features are generally more informative than global features for the medical image annotation task at hand.

We also compare the results of the experiments conducted with different feature fusion schemes. Inclusion of more than one type of features in the classification process contributes to better representation of the hierarchical nature of the images and helps to further improve the predictive performance. Low level fusion (concatenation) yields slightly better predictive performance than high level fusion. This is valid for all algorithms used in this study.

The classifiers on the fused feature sets use more information about the different aspects of an image that are captured by the different descriptors. Namely, they can consider combinations of features from different descriptors. This additional information is orthogonal and helps the classifiers to produce better annotations. Moreover, the ensembles of trees, such as random forests, can effectively exploit the information provided by the large number of features. Thus, low-level fusion yields better performance than high-level fusion.

The best results are achieved by using random forests on the concatenated SIFT, LBP and EHD descriptors (boldface in Table 1 and Table 2). This holds for both datasets, ImageCLEF2007 and ImageCLEF2008. The best results for overall recognition rate (81.96) are close to the error rate for the DICOM header. Namely, Guld et al [4] reported 15.5% disagreement between the data for the DICOM header and the radiologists' reference categorization. Moreover, our best results are better than the best results reported so far on this database [5]. Our score of 153.2 for ImageCLEF2008 is by 16.3 points better than the best result, and the score of 51.9 for ImageCLEF2007 is by 12.4 points better than the best result.

From the results, we can also notice the worse performance of all algorithms on the ImageCLEF2008 dataset, as compared to the ImageCLEF2007 dataset. This is mainly due to the larger hierarchy of the ImageCLEF2008 dataset (195 nodes as compared to 140 nodes for the ImageCLEF2007 dataset). In addition, the difference of the distribution of images in the training and the testing set is bigger for ImageCLEF2008 than for ImageCLEF2007.

Table 3: Running times of the algorithms: time needed to construct the classifier and time needed to produce an annotation for an unseen image. Note that this table only lists the results for the low-level fusion scheme (the results that end with 'LL'). The running times for the high-level fusion are the sum of running times for its constitutive runs. The experiments were executed on a Linux server with two Intel Quad-Core Processors@2.5GHz and 64GB of RAM.

		ImageCLEF 2007				**ImageCLEF 2008**			
		SVM	RF	Bag	PCTs	SVM	RF	Bag	PCTs
Training time [sec]	**EHD**	2820.873	92.668	566.880	4.667	3113.320	115.129	716.606	5.446
	LBP	4323.681	1909.510	21684.124	127.889	4406.340	2631.485	28612.105	158.955
	32x32	4745.630	1909.427	21458.823	110.436	5467.686	2614.089	28410.495	151.317
	SIFT	12451.760	2886.417	31611.480	227.709	13219.039	3717.713	40567.323	248.920
	LBP+EHD_LL	4824.592	2315.010	21629.071	231.516	4480.761	3012.840	28106.304	254.442
	LBP+SIFT_LL	14871.131	5095.170	55476.671	502.794	15788.345	6508.022	70057.262	487.347
	EHD+SIFT_LL	12656.792	3299.330	36001.937	337.784	13430.779	4165.986	45921.571	393.629
	LBP+EHD+SIFT_LL	15076.162	5094.305	55724.765	504.575	16006.638	6460.307	70462.933	500.873
	LBP+EHD+SIFT+32x32_LL	17700.564	6936.030	73786.231	591.772	18800.790	9128.094	95792.121	679.572
Testing time per image [sec]	**EHD**	0.016	0.002	0.003	0.001	0.019	0.004	0.003	0.001
	LBP	0.172	0.002	0.003	0.001	0.179	0.003	0.003	0.001
	32x32	0.189	0.002	0.003	0.001	0.192	0.002	0.002	0.001
	SIFT	0.551	0.002	0.003	0.001	0.591	0.003	0.004	0.001
	LBP+EHD_LL	0.175	0.003	0.002	0.001	0.176	0.002	0.003	0.001
	LBP+SIFT_LL	0.569	0.002	0.002	0.001	0.565	0.003	0.003	0.001
	EHD+SIFT_LL	0.552	0.002	0.003	0.001	0.552	0.003	0.003	0.001
	LBP+EHD+SIFT_LL	0.570	0.002	0.002	0.001	0.569	0.002	0.002	0.001
	LBP+EHD+SIFT+32x32_LL	0.600	0.002	0.002	0.002	0.590	0.003	0.003	0.002

Additionally, we assess the efficiency of the algorithms by measuring the time needed to learn the classifier and time needed to produce an annotation for an unseen image. The running times for the algorithms are presented in Table 3. The random forests are the fastest method; they are ∼ 10 times faster than bagging and ∼ 5.5 times than the SVMs (including the optimization of the SVM parameters). Recall that the random forests are ensembles of PCTs that predict the complete hierarchy (a single model), while the SVMs construct a classifier for each node of the hierarchy separately. Hence, the increase of the hierarchy will significantly increase the training time of SVMs (additional classifiers should be trained), while the training time for random forests will increase only slightly. The efficiency of the random forests of PCTs is even more prominent when producing annotations for unseen images. The random forests in this case are ∼ 165 times faster than the SVMs. In this respect, bagging performs comparably to random forests. This is due to the fact that passing through the tree has logarithmic complexity with respect to the number of leafs in the tree. Since random forests and bagging produce trees with similar sizes, these times will be similar. All in all, random forests of PCTs significantly outperform SVMs as compared by their training and testing times.

7. Experiments on photo annotation

To show the generality of the proposed system, we perform experiments on annotation of general images. In this section, we first present the experimental setup that we used (the data, evaluation metrics and the experimental design). We then present the results and compare them to those of state-of-the-art approaches used in image annotation.

7.1. Experimental setup

This set of experiments was performed using the database from the ImageCLEF@ICPR photo annotation task [53]. The database consists of 5000 train, 3000 validation, and 10000 test images annotated with 53 visual concepts organized in a small hierarchy with tree structure (see Fig. 10 for an example). The average number of annotations per image is 8.68 (including both leaf and internal nodes from the hierarchy). The visual concepts also contain abstract categories like Family/Friends, Partylife, Quality (blurred, underexposed, ...) and etc., thus making the annotation/classification task very challenging.

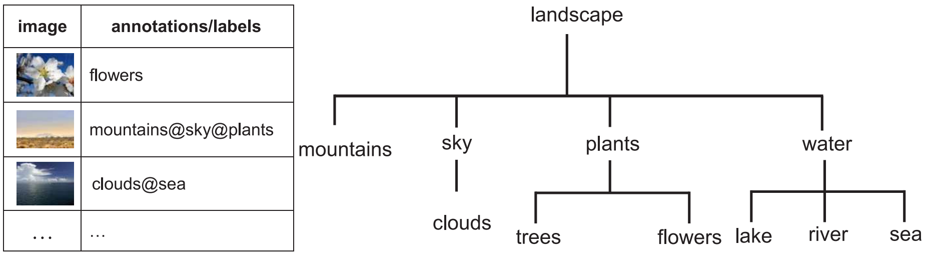

Figure 10: A fragment of the hierarchy for image annotation. The annotations are part of the hierarchical classification scheme for the ICPR 2010 photo annotation task (right). The table contains a set of images with their annotations (left).

The measures that we used to evaluate the performance of the algorithms on the medical X-ray images are specific for the problem of annotation of medical images using the IRMA coding scheme [2]. Here, we use the most widely used evaluation measure in the area of 'general photo annotation'/'visual concept detection': mean average precision (MAP) [12]. For a given target visual concept, the average precision can be calculated as the area under the precision-recall curve for that target. Hence, it combines both precision and recall into a single performance value. The average precision is calculated for each visual concept separately and the obtained values are then averaged to obtain the mean average precision. Because the true labels of the

[2] Note that the hierarchical error measure allows the algorithm to say 'don't know' for some classes, since the maximum number of labels per image with the IRMA coding scheme is known. In the case of general images, an image can be annotated with zero or $|C|$ classes. Also, for the Overall recognition rate, for the case of IRMA coding scheme, the number of possible combinations of labels is limited, while in the case of general images, this number is $2^{|C|}$. This makes overall recognition rate not suitable for measuring the predictive performance of algorithms in annotating general images.

test images from the ImageCLEF@ICPR 2010 database are not publicly available, we report the MAP value obtained on the validation dataset.

For the images from this database, we use SIFT features, which were the best performing features in previous experiments (also SIFT features are typically used in this type of problem [14]). The SIFT features for this set of experiments were constructed using a visual codebook with 4000 instead of 500 words (see Section 4.4). This modification was made because most of the state-of-the-art approaches for image classification of general photos use a visual codebook with 4000 words [14], [12]. In the previous experiments, random forests were the best performing method, so again we train random forests with 100 un-pruned PCTs for HMC. For the base PCTs, we used the same weight (0.75) and the size of the feature subset that is retained at each node was set to 10% of the number of descriptive attributes (same as in the experiments from the Section 5).

To train the SVMs, we use the LibSVM implementation with probabilistic outputs [54]. To solve the multiple classification problems, we employ again the *One-against-All* approach. For each visual concept, we build a binary classifier where instances associated with that visual concept are in one class (positive) and the rest are in another class (negative). To handle the imbalance in the number of positive versus negative training examples, we fix the weights of the positive and negative class. The weight of the positive class is set to $\frac{\#pos+\#neg}{\#pos}$ and the weight of the negative class is set to $\frac{\#pos+\#neg}{\#neg}$, with $\#pos$ the number of positive instances in the train set and $\#neg$ the number of negative instances [15]. As in the previous experiments, we optimize the value of the cost parameter C of the SVMs.

7.2. Results and discussion

The results from the photo annotation experiments are shown in Table 4. The table also contains the total training time and testing time per image for both SVMs and random forests of PCTs for HMC. From the presented results we can note that the random forests of PCTs for HMC outperform the SVMs both in terms of predictive performance and efficiency. The latter holds especially for the time needed to produce an annotation for a given test image: our approach is more than 500 times faster than the SVMs.

Table 4: Results of the photo annotation experiments evaluated using Mean Average Precision (larger values of MAP mean better performance).

	MAP	**Train time**	**Test time per image**
RF	0.450	9113.516	0.002
SVM	0.428	11821.227	1.078

Following the results from the study performed by Mensink et al. [22], this means that our system also outperforms the TagProp [21] approach for image annotation. The results show that our system offers better predictive performance and efficiency than systems that are most widely used for annotation of images. All in all, the proposed system has high predictive performance and efficiency, is general and is easily applicable to other domains.

8. Conclusions

Hierarchical multi-label classification (HMC) problems are encountered increasingly often in image annotation. However, flat classification machine learning approaches are predominantly applied in this area. In this paper, we propose to exploit the annotation hierarchy in image annotation by using ensembles of trees for HMC. Our approach to HMC exploits the annotation hierarchy by building a single classifier that simultaneously predicts all labels in the hierarchy. A substantial performance improvement is achieved by building ensembles of HMC trees, such as random forests.

We apply our approach to two benchmark tasks of hierarchical annotation of medical (X-ray) images and an additional task of photo annotation (i.e., visual concept detection). We compare it to a collection of SVMs (trained with a χ^2 kernel), each predicting one label at the lowest level of the hierarchy, the best-performing and most-frequently used approach to (hierarchical) image annotation. Our approach achieves better results than the competition on all of these: For the two medical image datasets, these are the best results reported in the literature so far. Our approach has superior performance, both in terms of accuracy/error and especially in terms of efficiency.

We explore the relative performance of ensembles of trees for HMC and collections of SVMs under a variety of conditions. Along one dimension, we consider three different datasets. Along another dimension, we consider two ensemble approaches, bagging and random forests. Furthermore, we consider several state-of-the-art feature extraction approaches and combinations thereof. Finally, we consider two types of feature fusion, i.e., low- and high-level fusion.

Ensembles of trees for HMC perform consistently better than SVMs over the whole range of conditions explored above. The two ensemble approaches perform better than SVM collections on all three tasks, with random forests being more efficient than bagging (and the most efficient overall). The relative performance holds for different image representations (we consider raw pixel representation, local binary patterns, edge histogram descriptors and SIFT histograms), as well as combinations thereof: The SIFT histograms are the best individual descriptors. Moreover, combinations of different descriptors yield better predictive performance than the individual descriptors. The relative performance also holds for both low-level and high-level fusion of the image descriptors, the former yielding slightly better performance. We can thus conclude that for the task of hierarchical image annotation, ensembles of trees for HMC are a superior alternative to using collections of SVMs, which are most-commonly applied in this context.

We expect it is possible to further improve the predictive performance of our system. We could try to adapt our tree-learning approach to tackle the shift in distribution of images between the training and the testing set. Better performance may also be obtained by including high level feature extraction algorithms able to give more understandable and compact representation of the visual content of the images (segmented objects with relations among them).

Let us conclude by emphasizing the scalability of our approach. Decision trees are one of the most efficient machine learning approaches and can handle large numbers of examples. The ensemble approach of random forests scales very well for large numbers of features. Finally, trees for HMC scale very well as the complexity of the annotation hierarchy increases, being able to handle very large hierarchies organized as trees or directed acyclic graphs. Combining these, our approach is scalable along all three dimensions.

Acknowledgment

The authors would like to thank Donato Malerba, Department of Computer Science, University of Bari for his valuable comments and suggestions.

References

[1] R. Choplin, J. Boehme, C. Maynard, Picture archiving and communication systems: an overview, Radiographics 12 (1) (1992) 127–129.

[2] S. Becker, R. Arenson, Costs and benefits of picture archiving and communication systems, Journal of the American Medical Informatics Association 1 (5) (1994) 361–371.

[3] N. E. M. Association, Digital imaging and communications in medicine - DICOM, http://dicom.nema.org/ (2009). URL `http://dicom.nema.org/`

[4] M. O. Guld, M. Kohnen, D. Keysers, H. Schubert, B. B. Wein, J. Bredno, T. M. Lehmann, Quality of DICOM header information for image categorization, in: SPIE vol. 4685 - Medical Imaging 2002: PACS and Integrated Medical Information Systems: Design and Evaluation, 2002, pp. 280–287.

[5] T. Tommasi, B. Caputo, P. Welter, M. O. Guld, T. M. Deserno, Overview of the CLEF 2009 medical image annotation track, in: CLEF 2009 Workshop - LNCS 6242, 2010, pp. 85–93.

[6] C. Silla, A. Freitas, A survey of hierarchical classification across different application domains, Data Mining and Knowledge Discovery 22 (1-2) (2011) 31–72.

[7] T. Tommasi, F. Orabona, B. Caputo, Discriminative cue integration for medical image annotation, Pattern Recognition Letters 29 (15) (2008) 1996–2002.

[8] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[9] D. K. Park, Y. S. Jeon, C. S. Won, Efficient use of local edge histogram descriptor, in: ACM workshops on Multimedia, 2000, pp. 51–54.

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

[11] S. Nowak, ImageCLEF@ICPR contest: Challenges, methodologies and results of the photo annotation task, in: International Conference on Pattern Recognition, 2010, pp. 489–492.

[12] M. Everingham, L. V. Gool, C. Williams, A. Zisserman, The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results, `http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html` (2009).

[13] T. Deselaers, H. Muller, P. Clough, H. Ney, T. M. Lehmann, The CLEF 2005 automatic medical image annotation task, International Journal of Computer Vision 74 (1) (2005) 51–58.

[14] S. Nowak, P. Dunker, Overview of the clef 2009 large-scale visual concept detection and annotation task, in: CLEF 2009 Workshop - LNCS 6242, 2010, pp. 94–109.

[15] K. van de Sande, T. Gevers, C. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1582–1596.

[16] M. Marszałek, C. Schmid, Constructing category hierarchies for visual recognition, in: European Conference on Computer Vision - LNCS 5305, Springer, 2008, pp. 479–491.

[17] M. Marszałek, C. Schmid, Semantic hierarchies for visual object recognition, in: IEEE Conference on Computer Vision & Pattern Recognition, 2007, p. 7.

[18] S. Liu, H. Yi, L.-T. Chia, D. Rajan, Adaptive hierarchical multi-class SVM classifier for texture-based image classification, in: IEEE International Conference on Multimedia and Expo, 2005, p. 4.

[19] L. Zhigang, S. Wenzhong, Q. Qianqing, L. Xiaowen, X. Donghui, Hierarchical support vector machines, in: IEEE International Geoscience and Remote Sensing Symposium, 2005, pp. 186–189.

[20] X. Yuan, W. Lai, T. Mei, X.-S. Hua, X. qing Wu, S. Li, Automatic video genre categorization using hierarchical SVM, in: IEEE International Conference on Image Processing, 2006, pp. 2905–2908.

[21] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, in: 12th International Conference on Computer Vision, 2009, pp. 309–316.

[22] T. Mensink, G. Csurka, F. Perronnin, J. Sanchez, J. J. Verbeek, LEAR and XRCE's participation to visual concept detection task - ImageCLEF 2010, in: CLEF 2010 LABs and Workshops, Notebook Papers, 2010, p. 12.

[23] T. Joachims, SVMstruct – support vector machine for complex outputs (2011).
URL `http://www.cs.cornell.edu/People/tj/svm_light/svm_struct.html`

[24] T. Gärtner, S. Vembu, On structured output training: hard cases and an efficient alternative, Machine Learning 76 (2009) 227–242.

[25] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, H. Blockeel, Decision trees for hierarchical multi-label classification, Machine Learning 73 (2) (2008) 185–214.

[26] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.

[27] S. Džeroski, P. Panov, B. Ženko, Ensemble methods in machine learning, in: Encyclopedia of complexity and systems science, Springer New York, 2009, pp. 5317–5325.

[28] G. Seni, J. F. Elder, Ensemble methods in data mining: Improving accuracy through combining predictions, Morgan & Claypool Publishers, 2010.

[29] T. Evgeniou, L. Perez-Breva, M. Pontil, T. Poggio, Bounds on the generalization performance of kernel machine ensembles, in: 17th International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., 2000, pp. 271–278.

[30] G. Valentini, T. Dietterich, Biasvariance analysis and ensembles of svm, in: Multiple Classifier Systems - LNCS 2364, Springer Berlin / Heidelberg, 2002, pp. 27–38.

[31] C. Wang, H. Yuan, J. Liu, T. Zhou, H. Lu, A novel support vector machine ensemble based on subtractive clustering analysis, in: Advances in Knowledge Discovery and Data Mining - LNCS 4426, Springer Berlin / Heidelberg, 2007, pp. 849–856.

[32] K. M. Ting, L. Zhu, Boosting support vector machines successfully, in: MCS '09: Proceedings of the 8th International Workshop on Multiple Classifier Systems - LNCS 5519, Springer-Verlag, 2009, pp. 509–518.

[33] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, Machine Learning 36 (1) (1999) 105–139.

[34] H. Blockeel, L. Schietgat, J. Struyf, S. Džeroski, A. Clare, Decision trees for hierarchical multilabel classification: A case study in functional genomics, in: Knowledge Discovery in Databases: PKDD 2006 - LNCS 4213, Springer Berlin / Heidelberg, 2006, pp. 18–29.

[35] T. M. Lehmann, H. Schubert, D. Keysers, M. Kohnen, B. B. Wein, The IRMA code for unique classification of medical images, in: SPIE vol. 5033 - Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation, 2003, pp. 440–451.

[36] H. Blockeel, L. D. Raedt, J. Ramong, Top-down induction of clustering trees, in: International Conference on Machine Learning, Morgan Kaufmann, 1998, pp. 55–63.

[37] L. Breiman, J. Friedman, R. Olshen, C. J. Stone, Classification and Regression Trees, Chapman & Hall/CRC, 1984.

[38] R. J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[39] D. Kocev, C. Vens, J. Struyf, S. Džeroski, Ensembles of multi-objective decision trees, in: European conference on Machine Learning - LNCS 4701, Springer, 2007, pp. 624–631.

[40] J. Struyf, S. Džeroski, Constraint based induction of multi-objective regression trees, in: 4th Workshop on Knowledge Discovery in Inductive Databases KDID - LNCS 3933, Springer, 2006, pp. 222–233.

[41] I. Slavkov, V. Gjorgjioski, J. Struyf, S. Džeroski, Finding explained groups of time-course gene expression profiles with predictive clustering trees, Molecular BioSystems 6 (4) (2010) 729–740.

[42] L. Breiman, Bagging predictors, Machine Learning 24 (2) (1996) 123–140.

[43] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[44] I. Dimitrovski, S. Loskovska, Content-based retrieval system for X-ray images, in: International Congress on Image and Signal Processing, 2009, pp. 2236–2240.

[45] D. Keysers, T. Deselaers, C. Gollan, H. Ney, Deformation models for image recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (8) (2007) 1422–1435.

[46] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: IEEE conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.

[47] D. Ziou, S. Tabbone, Edge detection techniques an overview, Pattern Recognition and Image Analysis 8 (24) (1998) 537–559.

[48] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.

[49] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, International Journal of Computer Vision 73 (2) (2007) 213–238.

[50] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, J. M. Geusebroek, Visual word ambiguity, IEEE Transactions on Pattern Analysis and Machine Intelligence 99 (1).

[51] R. D. C. Team, R: A language and environment for statistical computing (2009).
URL `http://www.R-project.org`

[52] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (2001).

[53] S. Nowak, Imageclef@icpr2010 – photo annotation task, http://www.imageclef.org/2010/ICPR/ (2010).
URL `http://www.imageclef.org/2010/ICPR/`

[54] H.-T. Lin, C.-J. Lin, R. C. Weng, A note on Platt's probabilistic outputs for support vector machines, Machine Learning 68 (2007) 267–276.