

Hierarchical Annotation of Medical Images

Ivica Dimitrovski, Member, IEEE, Dragi Koccev,
Suzana Loskovska, Senior Member, IEEE, Sašo Džeroski, Member, IEEE

Abstract—This paper presents a hierarchical multi-label classification system for medical image annotation. The system is composed of two parts: feature extraction and classification/annotation. The feature extraction provides global and/or local descriptions of the images in the form of numerical vectors. Using these numerical descriptions, we train a classifier, a predictive clustering tree (PCT), to produce annotations for unseen images. PCTs are able to handle target concepts that are organized in a hierarchy, i.e., perform hierarchical multi-label classification. To improve the classification performance, we construct ensembles (bags and random forests) of PCTs.

We evaluate our system on the IRMA database. The experiments show that our system outperforms SVMs. In addition, our approach is very general: it can be easily extended with new feature extraction methods, and it can thus be easily applied to other domains, types of images and other classification schemes. Furthermore, it can handle arbitrarily sized hierarchies organized as trees or directed acyclic graphs.

Index Terms— Automatic Image Annotation, Hierarchical Multi-Label Classification, Predictive Clustering Trees

I. INTRODUCTION

DIGITAL imaging in medicine is in constant growth due to the increasing availability of imaging equipment in hospitals (such as X-ray, computed tomography, magnetic resonance imaging, positron emission tomography, ultrasound, endoscopy and laparoscopy). Average-sized radiology departments now produce several tera-bytes of data annually. This prompts for efficient systems for image annotation, storage, retrieval and mining.

A straightforward way of using existing information retrieval tools for visual material is to manually annotate images by keywords and then apply text-based querying for retrieval. These annotations reflect the visual content of the images. For medical images, they can specify the image modality, body orientation, body region, or the biological system examined. However, manual annotation of images is an expensive and time-consuming procedure, especially given the large and constantly growing databases of medical images.

To tackle the problem of image retrieval, automatic image

annotation is proposed by which a computer system automatically assigns metadata in the form of captions or keywords to a digital image [1]. Typically, image analysis is performed by first extracting feature vectors; together with the training annotations, these are then used by a machine learning algorithm to learn to automatically assign annotations. The performance of the automatic image annotation 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 classifier training and automatic image annotation.

A single image may contain different meanings organized in hierarchical semantics: hence, hierarchical multi-label classification 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 always a subset of a previously defined set or hierarchy of labels. Multi-label classification is used in various domains, such as text classification [2], scene and video classification [3], medical imaging [4], and biological applications such as protein function classification and genomics [5]. One of the main issue involved in multi-label classification is the importance of detecting and incorporating correlations between labels into the multi-labeling process. A second and related issue is the additional complexity involved in multi-label learning, as compared to single-label learning.

Regardless of the number of visual properties that have to be learned and their mutual connections, most of the present systems for annotation of medical images learn a separate model for each visual property [4], [6], [7]. For this purpose, they adapt single-label classification algorithms for multi-label classification problems. Alternatively, they transform a multi-label classification problem into several single-label classification problems.

In this paper, we present a hierarchical multi-label classification system for medical image annotation. This system consists of two parts: processing (feature extraction) and classification of images. The image processing part converts an image to a set of numerical features extracted directly from the pixel values. The image classification part labels and groups the images. 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).

We investigate which type of feature extraction techniques is most suitable for X-Ray medical images. To this end, we generate four different types of descriptors for the images in the database: raw pixel representation (RPR) [8], local binary patterns (LBP) [9], edge histogram descriptors (EHD) [10] and

Manuscript received February 11, 2010.

I. Dimitrovski is with the Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia (corresponding author to provide phone: +38614773143; fax: +38614773315; e-mail: ivicad@feit.ukim.edu.mk).

D. Koccev and S. Džeroski are with Department of Knowledge Technologies, Jozef Stefan Institute, Ljubljana, Slovenia (e-mails: Dragi.Koccev@ijs.si, Saso.Dzeroski@ijs.si).

S. Loskovska is with the Department of Computer Science, Faculty of Electrical Engineering and Information Technologies, Skopje, Macedonia (e-mail: suze@feit.ukim.edu.mk).

scale-invariant feature transform (SIFT) [11]. We also investigate whether combinations of these techniques (obtained with simple concatenation of the feature vectors) can yield better predictive performance.

We evaluate the different techniques for feature extraction from images and their use in hierarchical multi-label classification on the medical X-ray images for the CLEF 2009 medical image annotations task [4]. The predictive performance of the ensembles (bags and random forest) of PCTs was compared to that of SVMs, the most widely used classifier in the area of image annotation. We use the hierarchical error measure from [4], commonly used for assessing the predictive performance over the database we use.

The remainder of the paper is organized as follows. Section 2 describes the techniques for feature extraction from images. Section 3 introduces predictive clustering trees and their use for HMLC. In Section 4, we explain the experimental setup. The obtained results and a discussion thereof are given in Section 5. Section 6 concludes the paper and points some directions for further work.

II. 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...) [12]. 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 [4], [13].

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 feature 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 [8].

A. 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 [14].

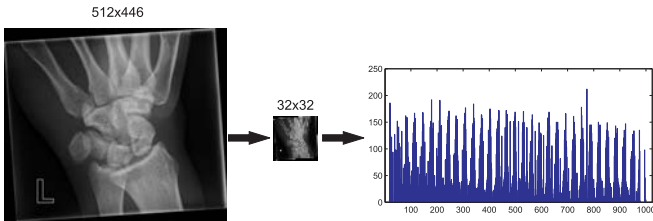


Fig. 1. Down-sampling for raw pixel representation.

We used a 32×32 down-sampled representation of the images as recommended in [8]. The obtained 1024 pixel

values were then used as input features. Fig. 1 shows how we built the raw pixel representation for each image.

B. Local binary patterns

Local binary patterns (LBP) are one of the best representations of texture content in images [9]. They are invariant to monotonic changes in gray-scale images and fast to compute. Furthermore, they are able to detect different micro patterns including edges, points, constant areas etc. LBP have already been used in many applications which require texture representation [15], [16], [17].

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:

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

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:

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

The image is traversed with the LBP operator pixel by pixel and the outputs are accumulated into a discrete histogram. The derived LBP histogram contains information about the distribution of local micro-patterns, such as edges, spots and flat areas, over the images. 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 [18]. 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.

In our experiments, we used the patterns $LBP_{8,1}^{u2}$, where the superscript $u2$ reflects the use of uniform patterns that have 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 large than 2) are grouped under one bin in the resulting histogram. With the $LBP_{8,1}^{u2}$ 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 [19], [20] to repeatedly sample predefined sub-regions of an image (e.g. 1×1 , 2×2 , 4×4 , 1×3 etc.). 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. 2 shows how we build the LBP histogram with 944 bins in total for each image (16 blocks with 59 bins each).

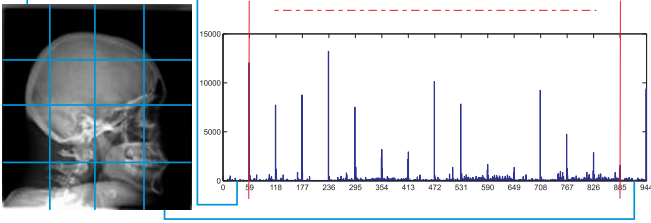


Fig. 2. The image is divided into 4x4 non-overlapping sub-images from which LBP histograms are extracted and concatenated into a single, spatially enhanced histogram.

C. Edge Histogram Descriptors

Edge detection is a fundamental problem of computer vision and has been widely investigated [21]. 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 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.

One way of representing important edge features is to use a histogram. An 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) [10]. The edge histogram descriptor basically represents the distribution of five types of edges in each local area/sub-image. The image space is divided into 4x4 non-overlapping blocks, yielding 16 equal-sized sub-images (Fig. 3).

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. 3. 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.

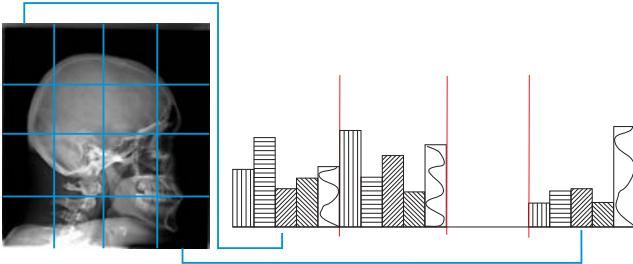


Fig. 3. The image is divided into 4x4 non-overlapping sub-images. For each sub-image five types of edge bins are calculated and concatenated into a single, spatially enhanced histogram.

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 [22].

D. SIFT descriptors

The scale-invariant feature transform (SIFT) is a method of extracting and describing key-points which are reasonably invariant to changes in illumination, image noise, rotation, scaling and small changes in viewpoint [11]. The SIFT algorithm has four major stages:

- 1) Scale-space extrema detection: searching over scale space using a difference of Gaussian functions to identify points of potential interest.
- 2) Key-point localization: the location and scale of each candidate point is determined and key-points are selected according to stability measures.
- 3) Orientation assignment: one or more orientations are assigned to each key-point based on local image gradients.
- 4) Key-point descriptor: a descriptor is generated for each key-point from local image gradients at the scale found in stage 2.

Regarding stages 1 and 2, in accordance with the conclusions from [8], we used random sampling. 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 in [23], that a dense random sampling is always superior to any strategy based on detectors for points of interest.

Regarding stage 3, the SIFT rotation-invariance is not relevant for the X-ray images that we used in our experiments because the various structures in the radiographs usually appear with the same orientation. Furthermore, the scale is not likely to change too much between images of the same class. These constraints are also in accordance with the conclusion presented in [8]. Therefore, we extracted the points only at the first octave and we removed the rotation-invariance.

To avoid using all visual features in an image we follow the well known codebook approach [23]. First, we assign visual features to discrete codewords predefined in a codebook. Then, we use the frequency distribution of the codewords as a compact feature vector. The crucial aspects of a codebook representation are the codebook construction and assignment. An extensive comparison of codebook representation variables is presented in [24].

We built the codebook by randomly sampling 50 points from each training image and extracting SIFT descriptors in each point. The total number of extracted key-points is 633850. We employ k-means clustering in the R environment for statistical computing and graphics [25] on this set of key-points to create the codewords. K-means partitions the visual feature space by minimizing the variance between a predefined number of k clusters. In our experiments, we set k to 500, so we defined a codebook with 500 codewords.

The random 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 previous descriptors (LBP

and EHD). For the SIFT descriptor, we used a spatial pyramid of 1x1, 2x2, and 1x3 regions. Since every region is an image in itself, the spatial pyramid can be easily used in combination with random sampling.

Finally, the feature vector for an image is defined by extracting a random collection of 1500 points from the entire image (spatial pyramid of 1x1), in each sub-image using a spatial pyramid of 2x2 and spatial pyramid of 1x3. The resulting distribution of descriptors in the feature space is then quantized in the codewords of the codebook and converted into a histogram of votes for each image/sub-image separately. The resulting vector with 4000 bins (8x500) was obtained by concatenation of these eight histograms. The total number of bins in each histogram is 500 because the codebook contains 500 codewords. Fig. 4 shows an example of the extracted histograms for spatial pyramids of 2x2 and 1x3.

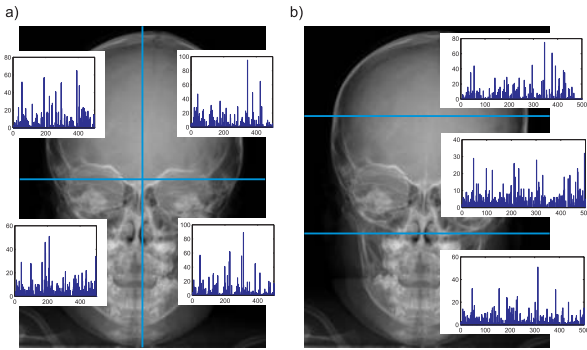


Fig. 4. Two different spatial pyramids used in our experiments, a) 2x2, b) 1x3. The spatial pyramid constructs feature vectors for each of the specific parts of the image.

III. ENSEMBLES OF PCTs FOR HMLC

A. PCTs for Hierarchical-Multi Label Classification

In the PCT framework [26], a tree is viewed 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 are constructed with a standard “top-down induction of decision trees” (TDIDT) algorithm. The heuristic for selecting the tests is the reduction in variance caused by partitioning the instances. Maximizing the variance reduction maximizes cluster homogeneity and improves predictive performance.

A leaf of a PCT is labeled with/predicts the prototype of the set of examples belonging to it. With appropriate variance and prototype functions, PCTs can handle different types of data, e.g., multiple targets [27] or time series [28]. A detailed description of the PCT framework can be found in [26].

To apply PCTs to the task of HMLC, 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 (see Fig. 5).

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.,

$$Var(S) = \frac{\sum_i d(v_i, \bar{v})^2}{|S|}$$

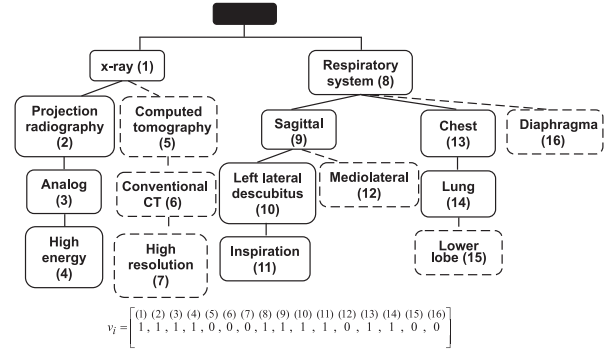


Fig. 5. Visualization of a part of the IRMA coding scheme. The codes are organized as a hierarchy, e.g., ‘Chest’ is a subclass of ‘Respiratory system’. The set of labels {x-ray, projection radiography, analog, high energy, respiratory system, sagittal, left lateral decubitus, inspiration, chest, lung} is indicated with solid round rectangles in the hierarchy and is represented as the vector v_i . These codes correspond to the image from Fig. 6a.

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:

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

where $v_{k,i}$ is the i 'th component of the class vector v_k of an instance x_k , and the class weights $w(c_i)$. The class weights decrease with the depth of the class in the hierarchy, $w(c_i) = w_0 w(c_j)$, 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 in that leaf. 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. A detailed description of PCTs for HMLC can be found in [29].

B. Ensemble Methods

An ensemble classifier 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) [30], [31], or more complex combinations.

We use PCTs for HMLC as base classifiers. Average 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 [30] 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 [31] 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., $f(x)=1$, $f(x)=\sqrt{x}$, $f(x)=\lfloor \log_2 x \rfloor + 1, \dots$). By setting $f(x)=x$, we obtain the bagging procedure.

IV. EXPERIMENTS

A. The IRMA Database

We evaluated our system by applying it to the database for the CLEF 2009 medical image annotations task [4]. This database is provided by the IRMA group from the University Hospital of Aachen, Germany [32]. 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).

The images are labeled according to the four annotation label sets [4]. We used the ImageCLEF2007 label set with 116 IRMA codes and the ImageCLEF2008 label set with 193 IRMA codes because of the hierarchical nature of the coding scheme [32]. The goal is to correctly annotate 1353/1733 images that are provided without labels, using the different respective annotation label sets in turn.

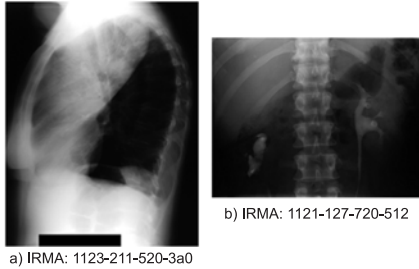


Fig. 6. IRMA-coded chest and abdomen radiograph. For instance, the code for the biological axis (512) on the sub-figure b) is translated as follows: 5 is for uropoietic system, 51 is for uropoietic system, kidney and 512 is uropoietic system, kidney, renal pelvis. The renal pelvis is an element of the kidney, which in turn is an element of the uropoietic system.

The IRMA coding scheme consists of four axes with three to four positions, each position taking a value from the set $\{0, \dots, 9, a, \dots, z\}$, 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. Fig. 6 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 hierarchical multi-label classification.

B. Evaluation Metrics

In this study, we use two evaluation metrics: precision-recall (PR) curves and the ImageCLEF hierarchical evaluation measure. Precision and recall are defined for binary classification problems, where the examples can be positive (belong to the class) or negative (do not belong to the class). Precision is the proportion of the positive predictions that are correct. Recall is the proportion of the positive examples that are correctly predicted as positive.

The PR curves are obtained by drawing the precision vs. the recall at different thresholds (note that a threshold is needed in order to make a prediction, from Section III.A and III.B). This measure offers the user a trade-off between the precision and the recall of the classifier. We are using this measure because we are more interested to correctly predict the presence of a label, rather than the absence of a label.

As performance measure, we use the area under the average PR curve. First, we construct the overall PR curve using average values for the precision and the recall over all classes. Then, we calculate the area under the average curve ($AUPRC$). This measure rewards a predictor that is able to exploit the information about the class frequencies of the different classes. The value for $AUPRC$ is in the range (0, 1) and if it is closer to 1 it means that the classifier is more accurate.

The ImageCLEF hierarchical evaluation measure is proposed in [4]. This measure takes into account the depth and the difficulty of the predictive problem (‘branching factor’) at which an error has occurred. It can be calculated using the following formula:

$$\sum_{i=1}^I \frac{1}{b_i} \frac{1}{i} \delta(v_i, \hat{v}_i) \quad ; \quad \delta(v_i, \hat{v}_i) = \begin{cases} 0 & \text{if } v_j = \hat{v}_j \quad \forall j \leq i \\ 0.5 & \text{if } v_j = * \quad \exists j \leq i \\ 1 & \text{if } v_j \neq \hat{v}_j \quad \exists j \leq i \end{cases}$$

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 to say ‘I don’t know’ 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.

C. Experimental Hypotheses

The goal of this study is to test the following hypotheses:

1. Does using ensembles of PCTs lift the predictive performance of a single PCT in the domain of image annotation? Which ensemble method performs better:

bagging or random forest? How do the ensembles of PCTs compare to the performance of SVMs?

2. Which feature extraction technique is most suitable for medical X-Ray images?
3. Can combinations of feature extraction techniques that capture different aspects of an image yield improvement in terms of annotation performance?

For the first hypothesis, we evaluate the performance of PCTs for HMLC and ensembles (bagging and random forest) of PCTs. We compare the HMLC approaches using $AUPRC$ and select the best performing method. After that, we compare the best method for HMLC with SVMs, using the ImageCLEF hierarchical error measure.

It has been shown [29] that exploitation of 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 SVM classifiers – the most widely used classifiers for medical image annotation [8].

To check which feature extraction technique is most suitable for medical X-Ray images (second hypothesis), 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 Tables I and II).

The various feature extraction techniques capture different aspects of an image. With the third hypothesis, we want to check whether combinations of feature extraction techniques can yield increase in the predictive performance. We concatenate the descriptors in a single feature vector and train a classifier on the joint feature vector. Here, we compare the last five rows from Tables I and II (with the first four).

D. Experimental design

In this section, we describe the experimental setup that we used. First, we describe how we re-engineered the hierarchy 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 II).

We modify the hierarchy of the IRMA code, in order to increase the inter-class variability and decrease the intra-class variability. Fig.7 shows the 're-engineered' hierarchy of the classes that we use. 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 visually more different images in the first level of the hierarchy. In the context of the axis A, the axis B is necessary because the body region examined insufficiently describes the content and structure of 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.

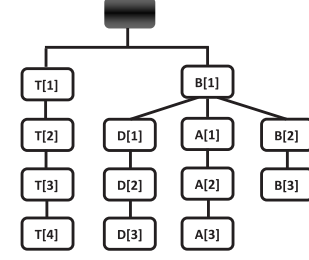


Fig. 7. The 're-engineered' hierarchy of the classes in the IRMA code.

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 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 [29].

We trained ensembles of 100 un-pruned trees (PCTs). For the base PCTs, we used the same weight (0.75) as when training 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.

We used a custom developed application for training and testing the SVMs. This application uses the SVMtorch library [33]. To solve the partial binary classification problems, we apply the One-against-All (OvA) approach. Each of the SVMs was trained with a Gaussian kernel. We used the default values for all parameters of the SVMs, except for the standard deviation (σ). This parameter presents a trade-off between over-fitting in the dense areas and under-fitting in the sparse areas of the dataset.

In order to select the best value for the standard deviation, we performed greedy search as follows. We separate 20% of the training set and use it as validation set. Then, we start training a SVM with σ set to 1 and record the performance of the SVM on the validation set. Afterwards, we increase the value of σ 1.5 times and train a new SVM classifier. We repeat this procedure until we reach a local optimum. Next, we add and extract some small value from the local optimum to fine tune the parameter. At the end, we train an SVM on the whole dataset with the selected value for standard deviation.

V. RESULTS AND DISCUSSION

In this section, we present the results obtained using the experimental setup described in the previous section. First, we compare the performance of PCTs and ensembles of PCTs (Table I). Then, we compare the performance of ensembles of PCTs and SVMs (Table II).

Table I summarizes the $AUPRC$ values for PCTs and ensembles of PCTs using the different feature extraction algorithms over the ImageCLEF 2007 and ImageCLEF 2008 datasets. In [25], Vens et al. show that PCTs for HMLC outperform significantly trees for single label classification, hierarchical single-label classification and the hierarchical extension of C4.5 [34]. Following their findings, we compare here the performance of PCTs for HMLC and ensembles of PCTs for HMLC as base classifiers.

TABLE I

PREDICTIVE PERFORMANCE ($AUPRC$) OF THE FEATURE EXTRACTION ALGORITHMS AND THEIR COMBINATIONS USING PCTs FOR HMLC, AS WELL AS RANDOM FOREST AND BAGGING OF PCTs FOR HMLC. BIGGER VALUE MEANS BETTER PERFORMANCE

		PCT	RForest	Bagging
ImageCLEF 2007	SIFT	0.789	0.930	0.927
	LBP	0.823	0.935	0.935
	EHD	0.790	0.901	0.906
	32x32	0.735	0.879	0.876
	LBP+EHD	0.817	0.936	0.935
	LBP+SIFT	0.830	0.943	0.941
	EHD+SIFT	0.793	0.932	0.931
	LBP+EHD+SIFT	0.822	0.945	0.942
	LBP+EHD+SIFT+32x32	0.824	0.94	0.939
ImageCLEF 2008	SIFT	0.749	0.888	0.882
	LBP	0.764	0.890	0.888
	EHD	0.746	0.856	0.859
	32x32	0.705	0.829	0.825
	LBP+EHD	0.756	0.891	0.889
	LBP+SIFT	0.755	0.897	0.896
	EHD+SIFT	0.754	0.889	0.887
	LBP+EHD+SIFT	0.757	0.896	0.894
	LBP+EHD+SIFT+32x32	0.751	0.893	0.894

The results clearly show that ensemble methods outperform single PCTs on all of the datasets: random forests are significantly better (according to the non-parametric Wilcoxon test for statistical significance) than single PCTs ($p < 2 \cdot 10^{-4}$) and bagging is better than single PCTs ($p < 2 \cdot 10^{-4}$). A comparison between the two ensemble methods shows that random forest outperforms bagging and that the difference is statistically significant ($p < 2.7 \cdot 10^{-2}$). Furthermore, the random forest method is (~ 10 times) faster than bagging.

From the results in Table I, we can also notice the worse performance of all algorithms on the ImageCLEF 2008 dataset, as compared to the ImageCLEF 2007 dataset. This is mainly because of the larger hierarchy of the ImageCLEF2008 dataset. The ImageCLEF 2008 dataset contains 195 nodes in the hierarchy, while the ImageCLEF 2007 dataset contains 140 nodes in the hierarchy.

The predictive performance of the individual feature extraction algorithms is shown in Fig. 8. From the PR curves we can note the high predictive performance of the LBP operator and the SIFT histogram. The LBP operator and the SIFT histogram are most capable of capturing the hierarchical structure of the X-ray images. The EHD feature performs slightly worse, and the simplest descriptor obtained from the raw pixel representation has worst performance. Similar conclusions can be made by observing the values of the $AUPRC$ from Table I and the ImageCLEF error in Table II.

Tables I and II present the results of the experiments conducted with different combinations of features. Inclusion of more than one type of features in the classification process contributes to the better representation of the hierarchical nature of the images and helps to further improve the predictive performance. For the ImageCLEF 2007 dataset, the best results were obtained using LBP, EHD and SIFT in combination, and for the ImageCLEF 2008 dataset by using

LBP and SIFT. This confirms our conclusion that LBP and SIFT are most capable of capturing the hierarchical nature of the image content.

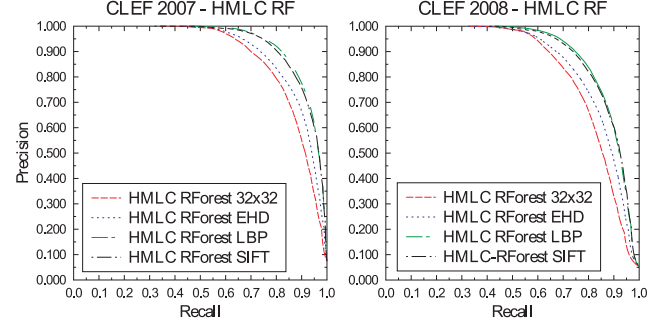


Fig. 8. Precision-Recall curves for the ImageCLEF 2007 and ImageCLEF2008 dataset, respectively, for the four different feature extraction algorithms. The curves were obtained using random forests of PCTs for HMLC.

TABLE II

PREDICTIVE PERFORMANCE (IMAGECLEF ERROR MEASURE) OF THE FEATURE EXTRACTION ALGORITHMS AND THEIR COMBINATIONS USING SVMs AND RANDOM FORESTS OF PCTs FOR HMLC. SMALLER VALUES MEAN BETTER PERFORMANCE.

		SVM	HMLC-RF
ImageCLEF 2007	SIFT	88.75	81.9
	LBP	125.19	72.71
	EHD	128.16	105.12
	32x32	203.69	190.78
	LBP+EHD	101.12	70.56
	LBP+SIFT	88.47	65.89
	EHD+SIFT	80.33	79.11
	LBP+EHD+SIFT	80.21	67.23
	LBP+EHD+SIFT+32x32	80.45	69.45
ImageCLEF 2008	SIFT	196.13	182.67
	LBP	259.17	179.47
	EHD	267.2	249.44
	32x32	378.18	321.21
	LBP+EHD	224.98	180.12
	LBP+SIFT	203.37	168.38
	EHD+SIFT	190.69	180.3
	LBP+EHD+SIFT	192.43	165.23
	LBP+EHD+SIFT+32x32	194.53	169.01

We have also compared our approach with the flat classification approach using SVMs (see Table II) by the ImageCLEF error measure. From the presented experimental results, we can see that the random forests of PCTs are superior to the SVM approach for all feature extraction algorithms and their combinations (the difference in performance is significant at $p < 2 \cdot 10^{-4}$). This shows that exploiting the structure of the hierarchy does help in improving the predictive performance. The error score of 165.23 for the ImageCLEF2008 dataset is better than the best error score reported at the ImageCLEF 2009 competition [4]. The error score of 65.89 is only by 1.59 points worse than the best error score reported for the ImageCLEF 2007 dataset also at the ImageCLEF 2009 competition [4].

VI. CONCLUSION

This paper presents a hierarchical multi-label classification approach to medical image annotation. For efficient image

representation, we used several feature extraction algorithms: raw pixel representation, local binary patterns, edge histogram descriptors and SIFT histograms. The predictive modeling problem that we consider is to hierarchically annotate medical X-ray images using the IRMA coding system.

The presented experimental results show that the LBP operator and the SIFT histogram most successfully describe and capture the hierarchical structure of the X-ray images. Moreover, the most favorable combination of representation and learning approaches, in terms of predictive performance and time efficiency, is when a LBP operator is used jointly with random forests of PCTs for HMLC.

The PCTs for HMLC make use of the information contained in the hierarchy to improve the classification performance (see the results from Table II). Furthermore, the PCTs approach (when a single tree is learned) has the additional advantage of identifying features that are relevant for all code labels in the hierarchy together (instead of separate features for each code label). The extensive experiments conducted on the benchmark database show that ensembles of PCTs outperform the best of the existing methods for medical image annotation.

There are several ways to further improve the predictive performance of the proposed approach. First, one could try to tackle the shift in distribution of images between the training and the testing set. A possible solution to this problem is to develop extensions of the PCT approach that can handle such differences. Second, better performance may 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).

In summary, we presented a general approach to hierarchical image annotation. The approach can be easily extended with new feature extraction methods, and can thus be applied to other domains. It can be also easily applied to arbitrary domains, because it can handle hierarchies with arbitrary sizes and shapes, including very large hierarchies and hierarchies that are organized as trees or directed acyclic graphs.

REFERENCES

- [1] J. Li and J. Z. Wang, Real-Time Computerized Annotation of Pictures. *IEEE Trans. Patt. Anal. Mach. Int.*, 30(6):985-1002, 2008.
- [2] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, Kernel-based learning of hierarchical multilabel classification models. *J. Machine Learning Research*, 7:1601–1626, 2006.
- [3] S. Nowak and P. Dunker, Overview of the CLEF 2009 Large-Scale Visual Concept Detection and Annotation Task. Working notes for the CLEF 2009 Workshop, Corfu, Greece, 2009.
- [4] T. Tommasi, B. Caputo, P. Welter, M. O. Guld, T. M. Deserno, Overview of the CLEF 2009 medical image annotation track, Working notes for the CLEF 2009 Workshop, Corfu, Greece, 2009.
- [5] Z. Barutcuoglu, R.E. Schapire, and O.G. Troyanskaya, Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.
- [6] T. Deselaers, H. Muller, P. Clough, H. Ney, T. M. Lehmann: The CLEF 2005 Automatic Medical Image Annotation Task. *Computer Vision* 74(1): 51-58, 2007.
- [7] H. Muller, P. Clough, W. R. Hersch, T. Deselaers, T. M. Lehmann, A. Geissbuhler, Evaluation axes for medical image retrieval systems: the imageCLEF experience. *Proceedings of 13th ACM Conference on Multimedia*, pp. 1014-1022, 2005.
- [8] T. Tommasi, F. Orabona, B. Caputo, Discriminative cue integration for medical image annotation. *Pattern Recognition Letters* 29(15): 1996-2002, 2008.
- [9] T. Ojala, M. Pietikainen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, *Pattern Recognition*, 29(1): 51-59, 1996.
- [10] D.K. Park, Y.S. Jeon, C.S. Won, Efficient use of local edge histogram descriptor. In *International Multimedia Conference, ACM workshops on Multimedia*, pp. 51–54, 2000.
- [11] D.G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *Int. Journal of Computer Vision*, 60(2):91–110, 2004.
- [12] T. Deselaers, D. Keysers, and H. Ney, Features for image retrieval: an experimental comparison, *Information Retrieval*, 11(2):77-107, 2008.
- [13] I. Dimitrovski, S. Loskovska, Content-based Retrieval System for X-ray images, 2nd International Congress on Image and Signal Processing (CISP'09), pp. 2236-2240, 2009.
- [14] D. Keysers, T. Deselaers, C. Gollan, H. Ney, Deformation Models for Image Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8), 1422–1435, 2007.
- [15] T. Ahonen, A. Hadid, M. Pietikainen, *Face Recognition with Local Binary Patterns LNCS*, Vol. 3021, pp.469-481 Springer, 2004.
- [16] T. Maenpaa, M. Turtinen, M. Pietikainen, Real-time surface inspection by texture, *Real-Time Imaging*, 9(5): 289-296, 2003.
- [17] M. Pietikainen, T. Nurmela, T. Maenpaa, M. Turtinen, View-based recognition of real-world textures, *Pattern Recognition*, 37(2):313-323, 2004.
- [18] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7): 971-987, 2002.
- [19] V. Takala, T. Ahonen, M. Pietikainen, Block-Based Methods for Image Retrieval Using Local Binary Patterns. *SCIA 2005*: 882-891.
- [20] S. Lazebnik, C. Schmid, and J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, *IEEE Conf. CVPR*, vol. 2, pp. 2169-2178, 2006.
- [21] D. Ziou, S. Tabbone, Edge Detection Techniques an Overview, *Pattern Recognition and Image Analysis*, 8(4):537-559, 1998.
- [22] J.F. Canny, A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 8(6): 679-698, 1986.
- [23] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study. *Computer Vision*, 73(2):213-238, 2007.
- [24] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, Visual word ambiguity, *IEEE Trans. on Patt. Anal. and Mach. Intel.*, In press, doi:10.1109/TPAMI.2009.132, 2010.
- [25] R Development Core Team. R: A language and environment for statistical computing. ISBN 3-900051-07-0, URL <http://www.R-project.org>, 2009.
- [26] H. Blockeel, L. De Raedt, J. Ramon, Top-down induction of clustering trees. In *Proc. of the 15th ICML*, p.55-63, 1998.
- [27] D. Koccev, C. Vens, J. Struyf, S. Dzeroski, Ensembles of Multi-Objective Decision Trees, In *Proc. of the ECML 2007, LNAI vol. 4701*, p. 624-631, 2007.
- [28] S. Dzeroski, V. Gjorgjioski, I. Slavkov, J. Struyf, Analysis of Time Series Data with Predictive Clustering Trees, In *KDID06, LNCS vol. 4747*, p. 63-80, 2007.
- [29] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, H. Blockeel, Decision trees for hierarchical multi-label classification, *Machine Learning*, 73(2):185-214, 2008.
- [30] L. Breiman, Bagging predictors, *Machine Learning*, 24(2):123-140, 1996.
- [31] L. Breiman, Random forests, *Machine Learning*, 45 (1):5-32, 2001.
- [32] T. M. Lehmann, H. Schuberta, D. Keysers, M. Kohnena, and B. B. Weina, The irma code for unique classification of medical images, *Proceedings of SPIE*, pp. 109-117, 2003.
- [33] R. Collobert, S. Bengio, J. Mariethoz, Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, 2002.
- [34] A. Clare, Machine learning and data mining for yeast functional genomics. PhD thesis, University of Wales, Aberystwyth, 2003.