Chapter 24
Idiap on Medical Image Classification

Tatiana Tommasi and Francesco Orabona

Abstract The team from the Idiap Research Institute in Martigny, Switzerland, participated in three editions of the CLEF medical image annotation task always reaching among the highest positions in the rankings. Here, we present in detailed form the successful strategies we used in the different editions of the challenge to face the inter– vs. intra–class image variability, to exploit the hierarchical labeling, and to cope with the unbalanced distribution of the classes.

24.1 Introduction

This chapter presents the algorithms and results of the Idiap participation in the ImageCLEFmed annotation task in 2007, 2008 and 2009. The goal of the challenge was to develop an automatic image annotation system able to distinguish x–ray images on the basis of the body region, the biological system examined, the body orientation and the imaging modality. The idea is to exploit content–based image analysis without making use of the textual information generally associated with medical images. A system performing this task reliably can avoid the cost of manually annotating several terabytes of image data collected annually in radiology departments and also help in image retrieval.

There are two main issues when working on large databases of medical images: intra–class variability vs. inter–class similarity and data imbalance. The first problem is due to the fact that images belonging to the same visual class might look very different, while images that belong to different visual classes might look very

Tatiana Tommasi
Idiap Research Institute, Centre du Parc, Rue Marconi 19, PO Box 592, 1920 Martigny, Switzerland, e-mail: ttommasi@idiap.ch

Francesco Orabona (work done while at Idiap Research Institute)
Dipartimento di Scienze dell'Informazione, Universita' degli Studi di Milano, via Comelico 39, 20135 Milano, Italy, e-mail: francesco@orabona.com

H. Müller et al. (eds.), *ImageCLEF*, The Information Retrieval Series 32,
DOI 10.1007/978-3-642-15181-1_24, © Springer-Verlag Berlin Heidelberg 2010

similar. Data imbalance is related to the natural statistics of the onset of diseases in the different parts of the body, thus it reflects the a priori probabilities of the routine diagnosis in a radiological clinic. To overcome both these problems, an automatic annotation system needs to use the most discriminative information from the available data; it also needs to be able to weigh properly the information coming from differently populated classes in the learning process.

For the CLEF challenge, the images were identified on the basis of the Image Retrieval in Medical Applications (IRMA) code (Lehmann et al, 2003). This is a multi–axial hierarchical scheme, which adds a further difficulty in the annotation process.

In our experience as participants of the ImageCLEFmed challenge, we tackled these problems and proposed different discriminative solutions based on Support Vector Machines (SVM) (Cristianini and Shawe-Taylor, 2000). In 2007 and 2008 our best run ranked first, while in 2009 the run reproducing the winning strategy of 2008 ranked second.

In the rest of the chapter we will focus on a number of issues as follows: Section 24.2 gives details about how we combined multiple cues to face the inter–class vs. intra–class variability; Section 24.3 introduces our confidence–based approach to exploit the hierarchical structure of the data; and Section 24.4 describes our strategy to overcome the data imbalance by creating virtual examples. Finally in Section 24.5 we describe our experimental set–up and summarize our results. Conclusions are drawn in Section 24.6.

24.2 Multiple Cues for Image Annotation

Several authors tried to address the inter-class vs. intra–class variability problem using local and global features, and more generally different types of descriptors, separately or combined together in a multiple cues approach (Müller et al, 2006; Güld et al, 2006; Florea et al, 2006). For some of these examples the performance was not very good. However, years of research on visual recognition in other domains have shown clearly that multiple cue methods outperform single–feature approaches (Matas et al, 1995; Mel, 1997; Sun, 2003). To have the maximum advantage from cue integration, each feature should represent a different aspect of the data allowing for a more informed decision. Heterogeneous and complementary visual cues, bringing different information content, were successfully used in the past (Slater and Healey, 1995; Mel, 1997; Nilsback and Caputo, 2004; Gehler and Nowozin, 2009). Regarding the integration techniques, they can all be reduced to one of these three approaches: *high–level*, *mid–level* and *low–level* integration (Sanderson and Paliwal, 2004; Polikar, 2006). Figure 24.1 illustrates schematically the basic ideas behind these methods.

Participating in the ImageCLEF challenge we proposed a discriminative approach for integration of cues by defining three strategies, one for each of the pos-

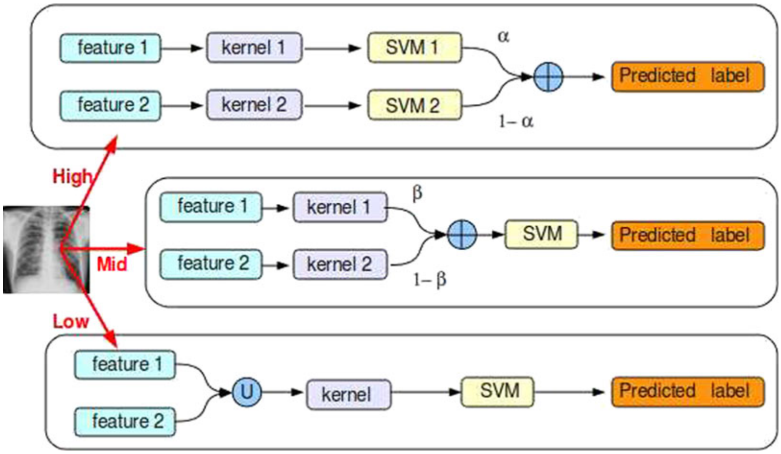


Fig. 24.1: A schematic illustration of the high–level, mid–level and low–level cue integration approaches.

sible levels of cue integration. The methods used are described in detail in the following sections.

24.2.1 High–Level Integration

High–level cue integration methods start from the output of two or more classifiers dealing with complementary information. Each of them produces an individual hypothesis about the object to be classified. All these hypotheses are then combined together to achieve a consensus decision. We applied this integration strategy using the Discriminative Accumulation Scheme (DAS) proposed first in (Nilsback and Caputo, 2004). It is based on a weak coupling method called accumulation, which does not neglect any cue contribution. Its main idea is that information from different cues can be summed together.

Suppose we are given M object classes and for each class a set of N_j training images $\{I_i^j\}_{i=1}^{N_j}$, $j = 1, \ldots M$. For each image we extract a set of P different features $T_p(I_i^j), p = 1 \ldots P$ so that for an object j we have P new training sets. For each feature we train an SVM. Kernel functions may differ from cue to cue and model parameters can be estimated via cross validation. Given a test image $\hat{I}$ and assuming $M \geq 2$, for each single–cue SVM we compute the distance from the separating hyperplane $D_j(p)$, $p = 1 \ldots P$. After collecting all the distances $\{D_j(p)\}_{p=1}^{P}$ for all the M objects and the P cues, we classify the image $\hat{I}$ using the linear combination:

$$j^* = \arg \max_{j=1\ldots M} \left\{ \sum_{p=1}^{P} a_p D_{j(p)} \right\} . \tag{24.1}$$

The coefficients $\{a_p\}_{p=1}^{P} \in \Re^+$ are determined via cross validation during the training phase.

24.2.2 Mid–Level Integration

Combining cues at the mid–level means that the different feature descriptors are kept separated, but they are integrated in a single classifier generating the final hypothesis. To implement this approach we developed a scheme based on multi–class SVMs with a Multi Cue Kernel, K_{MC}. This new kernel combines different features ($T_p(I)$) extracted from the images (I):

$$K_{MC}(\{T_p(I_i)\}_p, \{T_p(I)\}_p) = \sum_{p=1}^{P} a_p K_p(T_p(I_i), T_p(I)), \quad \sum_{p=1}^{P} a_p = 1 \; . \tag{24.2}$$

The Multi Cue Kernel is a Mercer kernel, as positively weighted linear combinations of Mercer kernels are Mercer kernels themselves (Cristianini and Shawe-Taylor, 2000). In this way it is possible to perform only one classification step, identifying the best weighting factors $a_p \in \Re^+$ through cross validation while determining the optimal separating hyperplane. This means that the coefficients a_p are guaranteed to be optimal.

24.2.3 Low–Level Integration

To combine cues it is also possible to use a low–level fusion strategy, starting from the descriptors and combining them in a new representation. In this way the cue integration does not directly involve the classification step. Here we use feature concatenation: two feature vectors f_i and c_i are combined into a single feature vector $v_i = (f_i, c_i)$ that is normalized to have its sum equal to one and is then used for classification. In this way the information related to each cue is mixed without a weighting factor that allows it to control the influence of each information channel on the final recognition result. A general drawback of this method is that the dimensionality of the feature vector increases as the number of cues grows, implying longer learning and recognition times, higher memory requirements and possibly risks curse of dimensionality effects.

24.3 Exploiting the Hierarchical Structure of Data: Confidence Based Opinion Fusion

The evaluation scheme for the medical image annotation task addresses the hierarchical structure of the IRMA code considering the number of possible choices at each node and the position of each node in the hierarchy. So, wrong decisions in easy nodes were penalized more than wrong decisions in difficult nodes, and mistakes at an early stage in the code were more costly than at a later stage. Moreover, the error evaluation method allowed the classifier to decide a 'don't know' at any level of the code, independently for each of the four axes: image modality, body orientation, body region and biological system (Lehmann et al, 2003).

In 2007 and 2008 many groups participating in the ImageCLEF medical annotation task tried to exploit the hierarchical labeling of the images classifying separately on the four axes of the IRMA code. However, analyzing the results, it was observed that the top–performing runs used each individual code as a class, using a flat classification approach which did not use the hierarchy (Deselaers et al, 2008). The only way to take advantage of the hierarchy seemed to be by exploiting the use of wildcard characters. Thus models which estimate the classifier's confidence in its decisions could be useful.

Discriminative classifiers usually do not provide any out–of–the–box solutions for estimating the confidence in the decision, but in some cases they can be transformed into opinion makers on the basis of the value of the discriminative function. In the case of SVM, it can be done by considering the distances between the test samples and the classification hyperplane. This approach turns out to be very efficient due to the use of kernel functions and does not require additional processing in the training phase.

In the one–vs.–all multi–class extension of SVM, if M is the number of classes, M SVMs are trained, each separating a single class from all remaining ones. The decision is then based on the distances of the test sample, $\boldsymbol{x}$, to the M hyperplanes, $D_j(\boldsymbol{x})$, $j = 1 \dots M$. The final output is the class corresponding to the hyperplane for which the distance is largest:

$$j^* = \arg \max_{j=1 \dots M} D_j(\boldsymbol{x}) \; . \tag{24.3}$$

If now we think of the confidence as a measure of unambiguity of the decision, we can define it as the difference between the maximal and the next largest distance:

$$C(\boldsymbol{x}) = D_{j^*}(\boldsymbol{x}) - \max_{j=1 \dots M, j \neq j^*} D_j(\boldsymbol{x}) \; . \tag{24.4}$$

The value $C(\boldsymbol{x})$ can be thresholded to obtain a binary confidence measure. Hence a confident prediction is assumed if $C(\boldsymbol{x}) > \tau$, for a given threshold τ. In the cases in which the decision is not confident, we decided to compare the labels corresponding to the first two margins and to put a 'don't know' term in the points of the code in which they differ.

24.4 Facing the Class Imbalance Problem: Virtual Examples

Unbalanced data sets define a challenging problem in machine learning. Classifiers generally perform poorly on unevenly distributed data sets because they are designed to generalize from sample data and output the simplest hypothesis that best fits them. In a binary problem with negative instances which heavily outnumber the positive ones, this means classifying almost all instances as negative. On the other hand, making the classifier too specific may make it sensitive to noise and prone to overfitting. Although SVMs have shown remarkable success in many applications, their capabilities are very limited when applied to the problem of learning from multi–class databases in which some of classes are sparsely populated. There are two known approaches to solve this problem. One is to bias the classifier so that it pays more attention to samples from poorly populated classes. This can be done, for instance, by increasing the penalty associated with misclassifying the class with few data with respect to the others. The second approach is to pre–process the data by resampling methods (Akbani et al, 2004). A possible alternative to resampling consists in exploiting the known invariances of the data to generate new synthetic minority instances and rebalance the data set. We adopted this solution.

Keysers et al (2003), the creators of the IRMA corpus used for the ImageCLEF challenge, explain that small transformations of the images do not alter their class membership. Therefore to improve the classification reliability, we enriched the poorly populated classes producing virtual examples as slightly modified copies of the training images. We increased and decreased each image side (100, 50 pixels); rotated them right and left (20, 40 degrees); shifted right, left, up, down and in the four diagonal directions (50 pixels); increased and decreased the brightness (add and subtract 20 to the original gray level). Thus the number of images in the poorly populated classes (with less than ten images) was increased by a factor of 17.

24.5 Experiments

All the techniques described above were optimized on the training set released and applied on the unlabeled test set of the last three editions of the CLEF challenge. In the following subsections we summarize the specific choices made in running the experiments and the results obtained.

24.5.1 Features

To extract different and complementary information from the images, we chose two types of features that were then combined with the high–, mid– and low–level integration strategies. In 2007 we combined a local (modSIFT) and a global feature

(Raw Pixels), while in 2008 and 2009 we considered two different local cues (modSIFT and LBP).

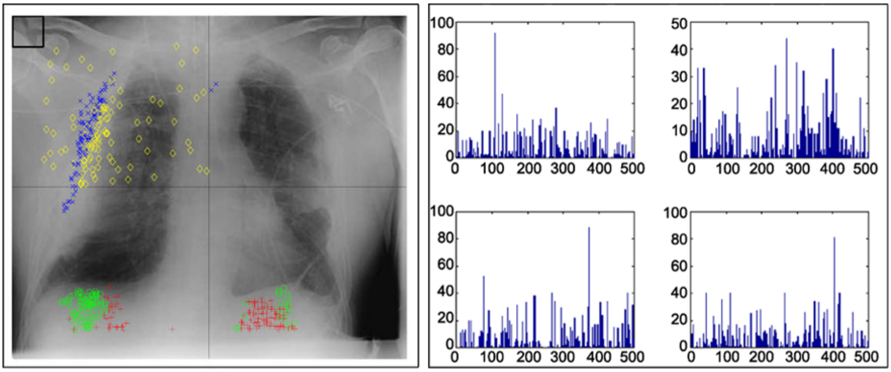
ModSIFT. Scale Invariant Feature Transform (SIFT) (Lowe, 1999) is a well known algorithm in computer vision used to detect and describe local features in images. We decided to use it adopting a bag–of–words approach: analogous to text classification, the basic idea is to sample image patches and to match them to a set of pre–specified 'visual words'. Note that the ordering of the visual words is not important and only the frequency of appearance of each word is used to form the feature vectors. The main implementation choices are thus: (1) how to sample patches, (2) what visual patch descriptor to use, and (3) how to build the vocabulary.

Regarding point (1), we used random sampling. Due to the low contrast of the radiographs it would be difficult to use any interest point detector. Moreover it has been pointed out by different papers and systematically verified by Nowak et al (2006) that a dense random sampling is always superior to any strategy based on interest point detectors for image classification tasks.

Regarding point (2), we decided to use a modified version of the SIFT descriptor. SIFTs are designed to describe an area of an image so as to be robust to noise, illumination, scale, translation and rotation changes. Given the specific constraints of our classification task, we slightly modified the classical version of this descriptor. The SIFT rotation invariance is not relevant for the ImageCLEFmed classification task, as the various structures in the radiographs are likely to always appear with the same orientation. Moreover, the scale is not likely to change too much between images of the same class. Hence a rotation– and scale–invariant descriptor could discard useful information for the classification. Thus we extracted the points at only one octave, the one that gave us the best classification performance on a validation set, and we removed the rotation–invariance. We call the modified SIFT descriptor modSIFT.

Regarding point (3), we built the vocabulary randomly sampling 30 points of each input image and extracting a modSIFT feature at each point. The visual words are created using an unsupervised K–means clustering algorithm. Note that in this phase both training and test images could be used, because the process does not need the labels. We chose K template modSIFTs with K equal to 500 and thus defined a vocabulary with 500 words. Various sizes of vocabulary were tested ($K = 500, 1,000, 2,000$). Preliminary results on a validation set showed no significant differences in performance between these three vocabulary sizes. We chose therefore $K = 500$, the smallest, for computational reasons.

Finally, the feature vector for an image is defined by extracting a random collection of points from the images. The resulting distribution of descriptors in the feature space is then quantized in the visual words of the vocabulary and converted into a frequency histogram. To add some spatial information, we decided to divide the images into four parts, collecting the histograms separately. In this way the dimension of the input space is multiplied by four (feature vector with $500 \times 4 = 2,000$ elements) but in our tests we gained about 3% in classification performance. We extracted 1,500 modSIFTs in each sub–image: such dense sampling adds robustness to the process. Figure 24.2 shows an example of the extracted local features.



(*a*) (*b*)

Fig. 24.2: (*a*) The four most present visual words in the image are drawn, each with a different color (better viewed in color). The square in the upper left corner represents the size of the patch used for computing the modSIFT descriptor. (*b*) Total counts of the visual words in the four sub–images.

In 2008 and 2009 we slightly modified the modSIFT feature inspired by the approach in (Lazebnik et al, 2006). We added to the original vector the histogram obtained extracting the feature from the whole image producing a final vector of 2,500 elements.

LBP. Local Binary Patterns (LBP) (Ojala et al, 2002) have been used extensively in face recognition, object classification (Ahonen et al, 2006; Zhang et al, 2007) and also in the medical area (Unay et al, 2007; Oliver et al, 2007). The basic idea of LBP is to build a binary code that describes the local texture pattern in a circular region thresholding each neighborhood on the circle by the gray value of its center. After choosing the dimension of the radius R and the number of points P to be considered on each circle, the images are scanned with the LBP operator pixel by pixel and the outputs are accumulated into a discrete histogram (Ojala et al, 2002). The operator is gray–scale invariant, moreover we used the *riu*2 rotational invariant LBP version which considers the uniform patterns with two spatial transitions ($\mathrm{LBP}^{riu2}_{P,R}$; (Ojala et al, 2002)).

Our preliminary results on a validation set showed that the best way to use LBP on the medical image database at hand was by combining a two dimensional histogram $\mathrm{LBP}^{riu2}_{8,8}$ with $\mathrm{LBP}^{riu2}_{16,12}$ and concatenating it with the two dimensional histogram made by $\mathrm{LBP}^{riu2}_{16,18}$ together with $\mathrm{LBP}^{riu2}_{24,22}$. In this way a feature vector of 648 elements is obtained. Each image is divided into four parts, one vector is extracted from each sub–image and from the central area and then they are concatenated producing a vector of 3,240 elements (see Figure 24.3).

Raw Pixels. We used the raw pixels as simplest possible global descriptor. Preliminary results on a validation set showed that downscaling images to 32 x 32 pixels did not produce any significant difference compared to downscaling to 48 x 48 but

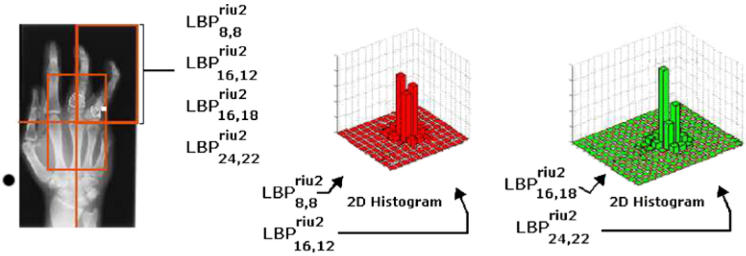


Fig. 24.3: A schematic drawing which shows how we built the texture feature vector combining the 1–dimensional histograms produced by the LBP operators in 2–dimensional histograms.

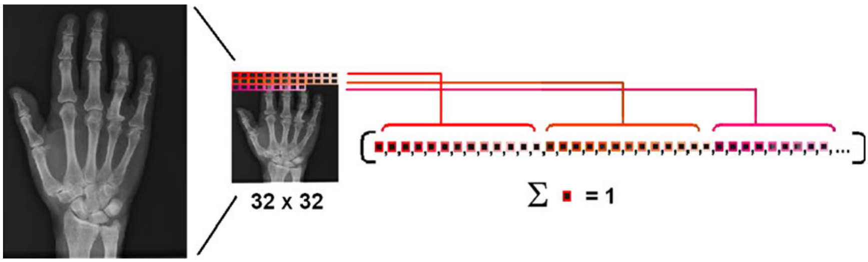


Fig. 24.4: An example showing the raw pixel representation.

the classification performance was better than that obtained on 16 x 16 images. So the images were resized to 32 x 32, regardless of the original dimension. The obtained $1,024$ pixel intensity values were then normalized to have sum equal to 1 and used as input features. Figure 24.4 shows how we built the raw pixel representation for each image.

24.5.2 Classifier

SVMs are a class of learning algorithms based on Statistical Learning Theory (Cristianini and Shawe-Taylor, 2000). Born as a linear classifier, SVM can be easily extended to nonlinear domains through the use of kernel functions. The kernels implicitly map the input space to a higher dimensional space, even with infinite dimensions. At the same time the generalization power of the classifier is kept under control by a regularization term that avoids overfitting in such high dimensional spaces (Cristianini and Shawe-Taylor, 2000).

The choice of the kernel heavily affects the performance of the SVM. We used an exponential χ^2 kernel for all the feature types and integration approaches, which is a valid kernel as proved in (Fowlkes et al, 2004):

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\gamma \sum_{i=1}^{N} \frac{(x_i - y_i)^2}{|x_i + y_i|}\right) . \tag{24.5}$$

In our experiments we also tested the linear kernel and the Radial Basis Function (RBF) kernel, but all of them gave worse results than the χ^2. The parameter γ was tuned through cross-validation together with the SVM cost parameter C.

Even if the labels are hierarchical, we used the standard one–vs.–all and one–vs.–one multi–class approaches. We verified experimentally that with our features, the recognition rate was lower using an axis–wise classification. This could be due to the fact that each super–class has a variability so high that our features are not able to model it, while they can model the small sub–classes very well.

24.5.3 Experimental Set–up and Results

To obtain reliable results in the training phase we used all the images released from the CLEF organizers, not considering the distinction between training and validation when it was suggested. Our strategy was to create five disjoint train/test splits on which to optimize the learning parameters. The performance was evaluated on the basis of the error score, the same as used in a second stage to rank the runs submitted to the challenge.

In 2008 and 2009, to take care of the class imbalance, the released database was divided into:

- rich_set: images belonging to classes with more than ten elements. From this group we built five disjoint sets, rich_train$_i$/rich_test$_i$, where the test sets were created by randomly extracting five images for each of the classes. Note that in this way we automatically considered a normalization on the classes.
- poor_set: images belonging to classes with less than ten elements. We used the whole poor_set as a second test set.

We trained the classifier on the rich_train$_i$ set and tested both on the rich_test$_i$ and on the poor_set, for each of the five splits. In this way, although the classes with few images were not considered in the training phase, we could evaluate the performance of the classifier to assign to those images the corresponding nearest class in the hierarchy. The error score was evaluated using the program released by the ImageCLEF organizers. The score values were normalized by the number of images in the corresponding test set, producing two average error scores. They were then multiplied by 500 and summed together supposing an ideal test set of 1,000 samples constituted half by images from the rich_set and half by images from the poor_set. The average of the scores obtained on the five splits is an estimator of the expected

Table 24.1: Ranking of our runs, name, score, and gain with respect to the best run of other participants (RWTHi6-4RUN-MV3) in 2007. The Low level cues integration was used only after the challenge. 'oa' and 'oo' indicate respectively the one–vs.–all and one–vs.–one SVM multi–class extensions.

Rank	**Name**	**Score**	**Gain**
1	Mid_oa	26.85	4.08
	Low_oa	26.96	3.96
	Low_oo	26.99	3.93
2	Mid_oo	27.54	3.38
3	modSIFT_oo	28.73	2.20
4	modSIFT_oa	29.46	1.47
5	High	29.90	1.03
6	RWTHi6-4RUN-MV3	30.93	0
28	PIXEL_oa	68.21	−37.28
29	PIXEL_oo	72.41	−41.48

value of the score. Each parameter in our methods was found by optimizing this expected score.

To evaluate the effect of introducing virtual examples in the poor_set we extracted from it only images belonging to classes with more than one element. We called this set poor_more. From this set we created six poor_more_train$_j$/poor_more_test$_j$ splits, where the train sets were defined extracting one image from each of the classes. Each poor_more_train set was enriched with the virtual examples as described in Section 24.4. Then we combined these sets joining rich_train$_i$ and poor_more_train$_j$ to build the training set and testing separately on rich_test$_i$ and poor_more_test$_j$.

Tables 24.1, 24.2, and 24.3 summarize all the results obtained by the Idiap team runs in 2007, 2008 and 2009 with the relative gain with respect to the best result from the other participating groups. In 2009 we participated in the ImageCLEFmed challenge organization and we decided to simply reuse the best approaches proposed in 2008, submitting these as baseline runs.

24.6 Conclusions

The Idiap team participated in the CLEF medical image annotation task from 2007 to 2009 proposing discriminative approaches coming from the image classification and recognition domain. The methods used are based on a combination of different local and global features and SVM as the classifier, together with specific solutions to face the class imbalance problem and to exploit the hierarchical labeling structure of data. On the basis of the results obtained we can state that the strategies adopted are suited to solve the challenging issue of annotating a large medical image database.

Table 24.2: Ranking of our submitted runs, name, score and gain with respect to the best run of the other participants (TAU-BIOMED-svm_full) in 2008. The extension 'virtual' stands for poor class enrichment by the use of virtual examples; 'confidence' stands for the combination of the first two SVM margins for the confidence based opinion fusion. For all the runs we used the one–vs.–all SVM multi–class extension.

Rank	**Name**	**Score**	**Gain**
1	Low_virtual_confidence	74.92	30.83
2	Low_virtual	83.45	22.30
3	Low_confidence	83.79	21.96
4	Mid_virtual_confidence	85.91	19.84
5	Low	93.20	12.55
6	modSIFT	100.27	5.48
7	TAU-BIOMED-svm_full	105.75	0
11	LBP	128.58	−22.83

Table 24.3: Ranking of our submitted runs, name, score and gain with respect to the best run of the other participants (TAUbiomed) in 2009. The extension 'virtual' stands for poor class enrichment by the use of virtual examples; 'confidence' stands for the combination of the first two SVM margins for the confidence based opinion fusion. For all the runs we used the one–vs.–all SVM multi–class extension.

Rank	**Name**	**Score**	**Gain**
1	TAUbiomed	852.8	0
2	Low_virtual_confidence	899.16	−46.36
3	Low_confidence:	899.4	−46.6
4	Low	1039.63	−186.83
5	Low_virtual	1042	−189.2

Acknowledgements The authors were supported by the EMMA project thanks to the Hasler foundation (http://www.haslerstiftung.ch/).

References

Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12):2037–2041

Akbani R, Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. In: European Conference on Machine Learning Lecture Notes in Computer Science (LNCS), vol 3201. Springer, pp 39–50

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines: and other kernel–based learning methods. Cambridge University Press

Deselaers T, Deserno TM, Müller H (2008) Automatic medical image annotation in ImageCLEF 2007: Overview, results, and discussion. Pattern Recognition Letters 29(15):1988–1995

Florea F, Rogozan A, Cornea V, Bensrhair A, Darmoni S (2006) MedIC/CISMeF at ImageCLEF 2006: image annotation and retrieval tasks. In: Working Notes of CLEF 2006

Fowlkes C, Belongie S, Chung F, Malik J (2004) Spectral grouping using the Nyström method. IEEE Transactions on Pattern Analysis and Machine Intelligence 26:214–225

Gehler P, Nowozin S (2009) On feature combination for multiclass object classification. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE Computer Society

Güld M, Thies C, Fischer B, Lehmann T (2006) Baseline results for the ImageCLEF 2006 medical automatic annotation task. In: CLEF 2006 Proceedings. Lecture Notes in Computer Science (LNCS), vol 4730. Springer, pp 686–689

Keysers D, Dahmen J, Ney H, Wein BB, Lehmann TM (2003) Statistical framework for model–based image retrieval in medical applications. Journal of Electronic Imaging 12(1):59–68

Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. Proceedings of the Conference on Computer Vision and Pattern Recognition 2:2169–2178

Lehmann TM, Schubert H, Keysers D, Kohnen M, Wein BB (2003) The IRMA code for unique classification of medical images. In: Proceedings SPIE, vol 5033, pp 109–117

Lowe DG (1999) Object recognition from local scale–invariant features. In: Proceedings of the IEEE International Conference on Computer Vision, vol 2. IEEE Computer Society, p 1150

Matas J, Marik R, Kittler J (1995) On representation and matching of multi–coloured objects. Proceedings of the IEEE International Conference on Computer Vision 726

Mel BW (1997) SEEMORE: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. Neural computation 9:777–804

Müller H, Gass T, Geissbuhler A (2006) Performing image classification with a frequency–based information retrieval schema for ImageCLEF 2006. In: Working Notes of CLEF 2006

Nilsback M, Caputo B (2004) Cue integration through discriminative accumulation. Proceedings of the Conference on Computer Vision and Pattern Recognition 2:578–585

Nowak E, Jurie F, Triggs B (2006) Sampling strategies for bag–of–features image classification. In: Proceedings of the European Conference of computer vision. Lecture Notes in Computer Science (LNCS). Springer, pp 490–503

Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray–scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7):971–987

Oliver A, Lladó X, Freixenet J, Martí J (2007) False positive reduction in mammographic mass detection using local binary patterns. In: Medical Image Computing and Computer–Assisted Intervention — MICCAI 2007 Lecture Notes in Computer Science (LNCS), vol 4791. Springer, pp 286–293

Polikar R (2006) Ensemble based system in decision making. IEEE Circuits and Systems Magazine 6(3):21–45

Sanderson C, Paliwal KK (2004) Identity verification using speech and face information. In: Digital Signal Processing, pp 449–480

Slater D, Healey G (1995) Combining color and geometric information for the illumination invariant recognition of 3–D objects. Proceedings of the International Conference on Computer Vision 563

Sun Z (2003) Adaptation for multiple cue integration. Proceedings of the Conference on Computer Vision and Pattern Recognition 440

Unay D, Ekin A, Cetin M, Jasinschi R, Ercil A (2007) Robustness of local binary patterns in brain MR image analysis. Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2098–2101

Zhang L, Li S, Yuan X, Xiang S (2007) Real–time object classification in video surveillance based on appearance learning. In: Proceedings of the Conference on Computer Vision and Pattern Recognition. IEEE Computer Society