Overview of the ImageCLEF 2007 Object Retrieval Task

Thomas Deselaers[1], Allan Hanbury[2], Ville Viitaniemi[3], András Benczúr[4], Mátyás Brendel[4], Bálint Daróczy[4], Hugo Jair Escalante Balderas[5], Theo Gevers[6], Carlos Arturo Hernández Gracidas[5], Steven C. H. Hoi[7], Jorma Laaksonen[3], Mingjing Li[7], Heidy Marisol Marin Castro[5], Hermann Ney[1], Xiaoguang Rui[7], Nicu Sebe[6], Julian Stöttinger[2], and Lei Wu[7]

[1] Computer Science Department, RWTH Aachen University, Germany
deselaers@cs.rwth-aachen.de
[2] Pattern Recognition and Image Processing Group (PRIP), Institute of Computer-Aided Automation, Vienna University of Technology, Austria
[3] Adaptive Informatics Research Centre, Helsinki University of Technology, Finland
[4] Data Mining and Web search Research Group, Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, Hungary
[5] TIA Research Group, Computer Science Department, National Institute of Astrophysics, Optics and Electronics, Tonantzintla, Mexico
[6] Intelligent Systems Lab Amsterdam, University of Amsterdam, The Netherlands
[7] School of Computer Engineering, Nanyang Technological University, Singapore
[8] Microsoft Research Asia, Beijing, China

Abstract. We describe the object retrieval task of ImageCLEF 2007, give an overview of the methods of the participating groups, and present and discuss the results.
The task was based on the widely used *PASCAL object recognition data* to train object recognition methods and on the *IAPR TC-12 benchmark dataset* from which images of objects of the ten different classes bicycles, buses, cars, motorbikes, cats, cows, dogs, horses, sheep, and persons had to be retrieved.
Seven international groups participated using a wide variety of methods. The results of the evaluation show that the task was very challenging and that different methods for relevance assessment can have a strong influence on the results of an evaluation.

1 Introduction

Object class recognition, automatic image annotation, and object retrieval are strongly related tasks. In object class recognition, the aim is to identify whether a certain object is contained in an image; in automatic image annotation, the aim is to create a textual description of a given image; and in object retrieval, images containing certain objects or object classes have to be retrieved out of a large set of images. Each of these techniques is important to allow for semantic retrieval from image collections.

Over the last year, research in these areas has strongly grown, and it is becoming clear that performance evaluation is a very important component for fostering progress in research. Several initiatives create benchmark suites and databases to quantitatively compare different methods tackling the same problem.

In the last years, evaluation campaigns for object detection [1, 2], content-based image retrieval [3] and image classification [4] have developed. There is however, no task aiming at finding images showing a particular object from a larger database. Although this task is extremely similar to the PASCAL visual object classes challenge [1, 2], it is not the same. In the PASCAL object recognition challenge, the probability for an object to be contained in an image is relatively high and the images to train and test the methods are from the same data collection. In realistic scenarios, this might not be a suitable assumption. Therefore, in the object retrieval task described here, we use the training data that was carefully assembled by the PASCAL NoE with much manual work, and the IAPR TC-12 database which has been created under completely different circumstances as the database from which relevant images are to be retrieved.

In this paper, we present the results of the object retrieval task that was arranged as part of the CLEF/ImageCLEF 2007 image retrieval evaluation. This task was conceived as a purely visual task, making it inherently cross-lingual. Once one has a model for the visual appearance of a specific object, such as a bicycle, it can be used to find images of bicycles independently of the language or quality of the annotation of an image.

ImageCLEF[9] [3] started within CLEF[10] (Cross Language Evaluation Forum) in 2003. A medical image retrieval task was added in 2004 to explore domain–specific multilingual information retrieval and also multi-modal retrieval by combining visual and textual features for retrieval. Since 2005, a medical retrieval and a medical image annotation task are both part of ImageCLEF. In 2006, a general object recognition task was presented to see whether interest in this area existed. Although only a few groups participated, many groups expressed their interest and encouraged us to create an object retrieval task. In ImageCLEF 2007, aside from the object retrieval task described here, a photographic retrieval task also using the IAPR TC-12 database [5], a medical image retrieval task [6], and a medical automatic annotation task [6] were organised.

2 Task Description

The task was defined as a visual object retrieval task. Training data was in the form of annotated example images of ten object classes (PASCAL VOC 2006 data). The task was to learn from the provided annotated images and then to find all images in the IAPR-TC12 database containing the learned objects. The particularity of the task is that the training and test images are not from the same set of images. This makes the task more realistic, but also more challenging.

[9] `http://www.imageclef.org`
[10] `http://www.clef-campaign.org/`

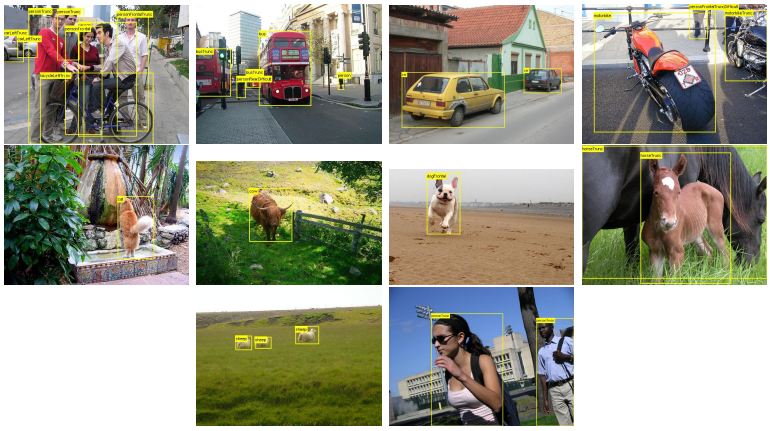

Fig. 1. Example images from the PASCAL VOC 2006 training dataset.

2.1 Datasets

For this task, the two datasets described below were used:

PASCAL VOC 2006 training data: As training data, the organisers of the PASCAL Network of Excellence visual object classes (VOC) challenge kindly agreed that we use the training data they assembled for their 2006 challenge. This data is freely available on the PASCAL web-page[11] and consists of approximately 2600 images, where for each image a detailed description of which of the ten object classes is visible in which area of the image is available (indicated by bounding boxes). Example images from this database are shown in Figure 1 with the corresponding annotation.

IAPR TC-12 dataset: The IAPR TC-12 Benchmark database [7] consists of 20,000 still images taken from locations around the world and comprising an assorted cross-section of still images which might for example be found in a personal photo collection. It includes pictures of different sports and actions, photographs of people, animals, cities, landscapes and many other aspects of contemporary life. Some example images are shown in Figure 2. This data is also strongly annotated using textual descriptions of the images and various meta-data. We use only the image data for this task.

2.2 Object Retrieval Task

The ten queries correspond to the ten classes of the PASCAL VOC 2006 data: bicycles, buses, cars, motorbikes, cats, cows, dogs, horses, sheep, and persons. For training, only the "train" and "val" sections of the PASCAL VOC database

[11] `http://www.pascal-network.org/challenges/VOC/`

Fig. 2. Example images from the IAPR TC-12 benchmark dataset

were to be used. For each query, participants were asked to submit a list of 1000 images obtained by their method from the IAPR-TC12 database, ranked in the order of best to worst satisfaction of the query.

2.3 Evaluation Measure

To evaluate the retrieval performance we use the same measure used by most retrieval evaluations such as the other tasks in CLEF/ImageCLEF [5, 6], TREC[12] and TRECVid[13]. The *average precision (AP)* gives an indication of the retrieval quality for one topic and the *mean average precision (MAP)* provides a single-figure measure of quality across recall levels averaged over all queries. To calculate these measures, it of course necessary to judge which images are relevant for a given query and which are not. To calculate the evaluation measures we use `trec_eval` [14], the standard program from TREC.

2.4 Relevance Assessments

To find relevant images, we created pools per topic [8] keeping the top 100 results from all submitted runs resulting in 1,507 images to be judged per topic on average. This resulted in a total of 15,007 images to be assessed. The normal relevance judgement process in information retrieval tasks envisages that several users judge each document in question for relevance and that for each image relevance for the particular query is judged. Given that judging the presence or absence of a given object in an image is a straightforward task, we postulate that every two persons among the judges would come to the same conclusion, and therefore each image was judged by only one judge. The whole judgement

[12] `http://trec.nist.gov/`
[13] `http://www-nlpir.nist.gov/projects/t01v/`
[14] `http://trec.nist.gov/trec_eval/`

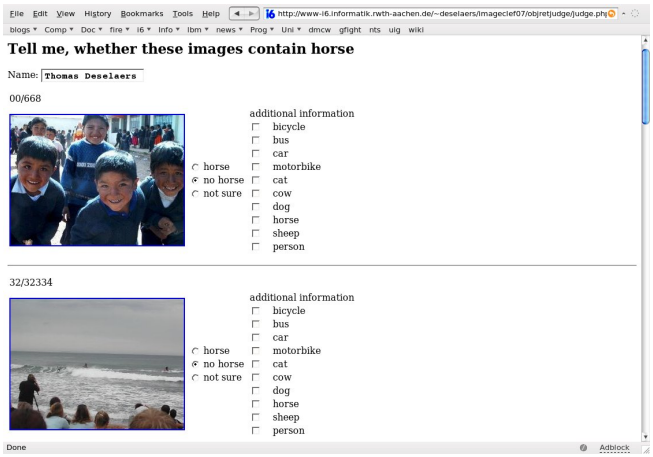

Fig. 3. The relevance judgement web-interface.

process was performed over a web interface which was quickly created and everybody from the RWTH Aachen University Human Language Technology and from the Vienna University of Technology Pattern Recognition and Image Processing (PRIP) group was invited to judge images. Thus, most of the judges are computer science students and researchers with a human language technology or pattern Recognition and image analysis background. Note that in the pooling process all images that are not judged are automatically considered to be not relevant.

The web-interface is shown in Figure 3 to give an impression of the process. On each page, 10 images are shown, and the judge has to decide whether a particular object is present in these images or not. To reduce boredom for the judges, they are allowed (and recommended) to specify whether other object classes are present in the images. This facility was added after the first 3,000 images had already been judged due to complaints by the judges that the task was too simple. The judges were told to be rather positive about the relevance of an image, e.g. to consider sheep-like animals such as llamas to be sheep and to consider tigers and other non-domestic cats to be cats. In the analysis of the results published in [9] and [10] it turned out that these judging guidelines were rather to imprecise and led to an inconsistent judging of the images.

Furthermore, Ville Viitaniemi from the HUTCIS group, judged all 20,000 images with respect to relevance for all of the topics with a stricter definition of relevances.

Results from the Relevance Judgements Table 1 gives an overview how many images were found to be relevant for each of the given topics using simulated pooling. For the initial evaluation [9], the pooling was done using some incorrect submissions and without sufficiently strict judging guidelines. Here, the pooling was simulated after all runs were checked to strictly follow the submission guide-

Table 1. Results from the relevance judgement process. Column 3 shows the number of relevant images when standard (simulated) pooling is used, column 4 when the (simulated) additional class information is taken into account. Column 5 shows the results of the relevance judgement of all 20,000 images.

query	query name	relev. in pool	additional relev.	relev. in database
1	bicycle	81/1422 (5.7%)	350/10060 (3.5%)	655/20000 (3.3%)
2	bus	29/1481 (2.0%)	106/10060 (1.1%)	218/20000 (1.1%)
3	car	219/1665 (13%)	644/10060 (6.4%)	1268/20000 (6.3%)
4	motorbike	17/1481 (1.1%)	48/10060 (0.48%)	86/20000 (0.43%)
5	cat	2/1566 (0.13%)	4/10060 (0.04%)	7/20000 (0.04%)
6	cow	10/1559 (0.64%)	30/10060 (0.30%)	49/20000 (0.25%)
7	dog	4/1554 (0.26%)	32/10060 (0.32%)	72/20000 (0.36%)
8	horse	33/1547 (2.1%)	110/10060 (1.1%)	175/20000 (0.88%)
9	sheep	0/1427 (0.00%)	1/10060 (0.01%)	6/20000 (0.03%)
10	person	1095/1734 (63%)	5356/10060 (53%)	11248/20000 (56%)

lines using the annotation of the full database. It can be observed that there are far more relevant images for the person topic than for any other topic. From these numbers it can be seen that the task at hand is challenging for most of the classes. It can also be observed that the percentage of relevant images in the additional pooling observation is very similar to the full database annotation and thus we can assume that choosing a (sufficiently large) random partition of documents to be judged can lead to a good estimate of relevant documents in the database. However, since the assumption that objects occur uncorrelated in the images is certainly invalid, this additional relevance information, which favors images with at least two different objects shown, is not optimal.

If only the data from the conventional pooling process is considered, then for five of the ten classes less than a thousandth of all images in the database are relevant, and the fact that still a high number of images has to be judged makes the usefulness of the whole judging process for this task questionable.

Another problem with pooling is reusability: since only a small portion of the relevant images in the whole database is found by the pooling process, the evaluation of a new method with the found pools is questionable. The additional pools, given that more of the relevant images are found, might be better suited, but as described above introduce a different form of bias.

3 Methods

Seven international groups from academia participated in the task and submitted a total of 38 runs. The group with the highest number of submissions had 13 submissions. In the following sections, the methods of the groups are explained (in alphabetical order) and references to further work are given.

3.1 Budapest methods

Authors: Mátyás Brendel, Bálint Daróczy, and András Benczúr
Affiliation: Data Mining and Web search Research Group, Informatics Laboratory, Computer and Automation Research Institute of the Hungarian Academy of Sciences
Email: {mbrendel, daroczyb, benczur}@ilab.sztaki.hu

budapest-acad315 The task of object retrieval is to classify objects found in images. This means to find objects in an image that are similar to sample objects in the pre-classified images. There are two problems with this task: the first is, how do we model objects. The second is, how do we measure similarity of objects. Our first answer to the first question is to model objects with image segments. Segment, region or blob based image similarity is a common method in content based image retrieval, see for example [11–14].

Instead of the PASCAL VOC 2006 database we used the PASCAL VOC 2007 database, since that database contained samples with exact object-boundaries, which is important for our methods. It is possible that our method will also work almost with the same efficiency with the PASCAL VOC 2006 database, but we have no test for this at current time.

The basis of our first method is to find segments on the query image which are similar to the objects in the pre-classified images. The image is then classified to be in that class, to which we find the most similar segment in the query image.

Image segmentation in itself is a widely researched and open problem. We used an image segmenter developed by our group to extract segments from the query images. Our method is based on a graph-based algorithm developed by Felzenszwalb and Huttenlocher [15]. We implemented a pre-segmentation method to reduce the computational time and use a different smoothing technique. All images were sized to a fixed resolution. Gaussian-based smoothing helped us cut down high frequency noise. Because of the efficiency of the OpenCV[15], implementation we did not implement resizing and Gaussian-based smoothing algorithms. As pre-segmentation we built a three-level Gaussian-Laplacian pyramid to define initial pixel groups. The original pyramid-based method, which considers the connection between pixels on different levels too, was modified to eliminate the so-called blocking problem. We used brightness difference to measure distance between pixels:

$$diffY(P_1, P_2) = 0.3 * \mid R_{P_2} - R_{P_1} \mid + 0.59 * \mid G_{P_2} - G_{P_1} \mid + 0.11 * \mid B_{P_2} - B_{P_1} \mid \quad (1)$$

After pre-segmentation, we had segments of 16 pixels maximum. To detect complex segments, we modified the original graph-based method by Felzenszwalb and Huttenlocher [15] with an adaptive threshold system using Euclidean distance to prefer larger regions instead of small regions of the image. Felzenszwalb

[15] http://www.intel.com/technology/computing/opencv/

and Huttenlocher defined an undirected graph $G = (V, E)$ where $\forall v_i \in V$ corresponds to a pixel in the image, and the edges in E connect certain pairs of neighboring pixels. This graph-based representation of the image reduces the original proposition into a graph cutting challenge. They made a very efficient and linear algorithm that yields a result near to the optimal normalized cut which is one of the NP-complete graph problems [15, 16]. The algorithm is listed in Algorithm 1.

Algorithm 1 Segmentation algorithm.

Algorithm $Segmentation\,(I_{src}, \tau_1, \tau_2)$
τ_1 and τ_2 are threshold functions. Let I_2 be the source image, I_1 and I_0 are the down-scaled images. Let $P(x, y, i)$ be the pixel $P(x, y)$ in the image on level i (I_i). Let $G = (V, E)$ be an undirected weighted graph where $\forall v_i \in V$ corresponds to a pixel $P(x, y)$. Each edge (v_i, v_j) has a non-negative weight $w(v_i, v_j)$.

Gaussian-Laplacian Pyramid

1. For every P(x,y,1) $Join(P(x, y, 1), P(x/2, y/2, 0))$ if $\tau_1 < diffY(P(x, y, 1), P(x/2, y/2, 0))$
2. For every P(x,y,2) $Join(P(x, y, 2), P(x/2, y/2, 1))$ if $\tau_1 < diffY(P(x, y, 2), P(x/2, y/2, 1))$

Graph-based Segmentation

1. Compute $Max_{weight}(R) = \max_{e \in MST(R,E)} w(e)$ for every coherent group of points R where $MST(R, E)$ is the minimal spanning tree
2. Compute $Co(R) = \tau_2(R) + Max_{weight}(R)$ as the measure of coherence between points in R
3. $Join(R_1, R_2)$ if $e \in E$ exists so $w(e) < min(Co(R_1), Co(R_2))$ is true, where $R_1 \bigcap R_2 = \emptyset$ and $w(e)$ is the weight of the border edge e between R_1 and R_2
4. Repeat steps 1,2,3 for every neighboring group (R_1, R_2) until possible to join two groups

This algorithm sometimes does not find relevant parts with low initial thresholds. To find the relevant borders which would disappear with the graph-based method using high thresholds we calculated the Sobel-gradient image to separate important edges from other remainders.

Similarity of complex objects is usually measured on a feature base. This means that the similarity of the objects is defined by the similarity in a certain feature space.

$$dist(S_i, O_j) = d(F(S_i), F(O_j)) : S_i \in S, O_j \in O \tag{2}$$

where S is the set of segments and O is the set of objects, $dist$ is the distance function of the objects and segments, d is a distance function in the feature space (usually some of the conventional metrics in the n-dimensional real space), F is the function which assigns features to objects and segments. We extracted from the segments features, like mean color, size, shape information, and histogram information. As shape information a 4×4 sized low-resolution variant of the segment (framed in a rectangle with background) was used. Our histograms had

5 bins in each channel. Altogether a 35 dimensional, real valued feature-vector was extracted for each of the segments. The same features were extracted for the objects in the pre-classified images taking them as segments. The background and those classes which were not requested were ignored. The features of the objects were written to a file, with the class-identifiers, which were extracted from the color-coding. This way we obtained a data-base of class samples, containing features of objects belonging to the classes. After this, the comparison of the objects of the pre-classified sample images and the segments of the query image was possible. We used Euclidean distance to measure similarity. The distance of the query-image Q was computed as:

$$dist(Q) = \min_{i,j} dist(S_i, O_j) : S_i \in S, O_j \in O \tag{3}$$

where S is the set of segments of image Q, O is the set of the pre-classified sample objects. Q is classified to be in the class of the object that minimizes the distance. The score of an image was computed as:

$$score(Q) = 1000/dist(Q) \tag{4}$$

where Q is the query image.

budapest-acad314 In our first method (see budapest-acad315) we found that our segments are much smaller than the objects in the pre-segmented images. It would have been possible to get larger segments by adjusting the segmentation algorithm, however this way we would not get segments which were really similar to the objects. We found that our segmentation algorithm could not generate segments similar to the the objects in the pre-classified images with any settings of the parameters. Even if we tried our algorithm on the sample images, and the segments were approximately of the same size, the segments did not match the pre-classified objects. The reason for this is that pre-segmentation was made by humans and algorithmic segmentation is far from capable of the same result. For example, it is almost impossible to write an algorithm, which would segment a shape of a human being as one segment if his clothes are different. However, people were one of the classes defined, and the sample images contained people with the entire body as one object. Therefore we modified our method. Our second method is still segment-based. But we also do a segmentation on the sample-images. We took the segmented sample-images, and if a segment was 80% inside of an area of a pre-defined object, then we took this segment as a proper sample for that object. This way a set of sample segments was created. After this the method is similar to the previous, the difference is only that we have sample segments instead of sample objects, but we treat them the same way. The features of the segments were extracted and they were written to a file, with the identifier of the class, which was extracted from the color-codes. After this, the comparison of the segments of the pre-classified images and the query image was possible. We used Euclidean distance again to measure similarity. The closest segment of the image to a segment in any of the objects was searched using thhe distance

$$dist(Q) = \min_{i,j} dist(S_i, S_j) : S_i \in S, S_j \in O \quad (5)$$

where S is the segments of image Q, O is the set of segments belonging to the pre-classified objects. The image was classified according to the object, to which the closest segment belongs. As we expected, this modification made the algorithm better.

3.2 HUTCIS: Conventional Supervised Learning using Fusion of Image Features

Authors: Ville Viitaniemi, Jorma Laaksonen
Affiliation: Adaptive Informatics Research Centre/Laboratory of Computer and Information Science, Helsinki University of Technology, Finland
Email: `firstname.lastname@tkk.fi`

All our 13 runs identified with prefix HUTCIS implement a similar general system architecture with three system stages:

1. Extraction of a large number of global and semi-global image features. Here we interpret global histograms of local descriptors as one type of global image feature.
2. For each individual feature, conventional supervised classification of the test images using the VOC2006 trainval images as the training set.
3. Fusion of the feature-wise classifier outputs.

By using this architecture, we knowingly ignored the aspect of qualitatively different training and test data. The motivation was to provide a baseline performance level that could be achieved by just applying a well-working implementation of the conventional supervised learning approach. Table 2 with ROC AUC performances in the VOC 2006 test set reveals that the performance of our principal run HUTCIS_SVM_FULLIMG_ALL is relatively close to the best performances in last year's VOC evaluation [2]. The last row of the table indicates what the rank of the HUTCIS_SVM_FULLIMG_ALL run would have been among the 19 VOC 2006 participants.

The following briefly describes the components of the architecture. For a more detailed description, see e.g. [17].

Table 2. ROC AUC performance in VOC2006 test set

Run id.	bic.	bus	car	cat	cow	dog	horse	mbike	person	sheep
FULLIMG_ALL	0.921	0.978	0.974	0.930	0.937	0.866	0.932	0.958	0.874	0.941
FULLIMG_IP+SC	0.922	0.977	0.974	0.924	0.934	0.851	0.928	0.953	0.865	0.941
FULLIMG_IP	0.919	0.952	0.970	0.917	0.926	0.840	0.903	0.943	0.834	0.936
Best in VOC2006	0.948	0.984	0.977	0.937	0.940	0.876	0.927	0.969	0.863	0.956
Rank	7th	4th	3rd	4th	4th	3rd	1st	5th	1st	6th

Table 3. Some of the image features used in the HUTCIS runs

Colour layout	Dominant colour
Sobel edge histogram (4x4 tiling of the image)	HSV colour histogram
Average colour (5-part tiling)	Colour moments (5-part tiling)
16×16 FFT of edge image	Sobel edge histogram (5-part tiling)
Sobel edge co-occurrence matrix (5-part tiling)	

Features: For different runs, the features are chosen from a set of feature vectors, each with several components. Table 3 lists 10 of the features. Additionally, the available feature set includes interest point SIFT feature histograms with different histogram sizes, and concatenations of pairs, triples and quadruples of the tabulated basic feature vectors. The SIFT histogram bins have been selected by clustering part of the images with the self-organising map (SOM) algorithm.

Classification and fusion: The classification is performed either by a C-SVC implementation built around the LIBSVM support vector machine (SVM) library [18], or a SOM-based classifier [19]. The SVM classifiers (prefix HUTCIS_SVM) are fused together using an additional SVM layer. For the SOM classifiers (prefix HUTCIS_PICSOM), the fusion is based on the summation of the normalised classifier outputs.

The different runs: Our principal run HUTCIS_SVM_FULLIMG_ALL implements all the three system stages in the best way possible. Other runs use subsets of the image features, inferior algorithms or are otherwise predicted to be suboptimal.

The run HUTCIS_SVM_FULLIMG_ALL performs SVM-classification with all the tabulated features, SIFT histograms and twelve previously hand-picked concatenations of the tabulated features, selected on the basis of SOM classifier accuracy in the VOC2006 task. The runs HUTCIS_SVM_FULLIMG_IP+SC and HUTCIS_SVM_FULLIMG_IP are otherwise similar but use just subsets of the features: SIFT histograms and colour histogram, or just SIFT histograms, respectively.

The runs identified by prefix HUTCIS_SVM_BB are naive attempts to account for the different training and test image distributions. These runs are also based on SIFT histogram and colour histogram features. For the training images, the features are calculated from the bounding boxes specified in the VOC2006 annotations. For the test images, the features are calculated for whole images. The different runs with this prefix correspond to different ways of selecting the images as a basis for SIFT codebook formation.

The run HUTCIS_FULLIMG+BB is the rank based fusion of features extracted from full images and bounding boxes. The runs HUTCIS_PICSOM1 and HUTCIS_PICSOM2 are otherwise identical but use different settings of the SOM classifier parameters. HUTCIS_PICSOM2 smooths the feature spaces less, and

the detection is based on more local information. Both the runs are based on the full set of features mentioned above.

Results: As expected, the run HUTCIS_SVM_FULLIMG_ALL with the full set of visual features extracted from the whole image turned out to be the best of our runs on average. However, for several individual query topics other runs produced better results. It remains unclear how much of the difference is explained by statistical fluctuations and how much by genuine differences between the various techniques on one hand, and between query topics on the other. However, by comparison with purely random AP values [10] it is reasonable to believe that some of the differences reflect real phenomena.

The mechanism for fusing the visual features was generic and straightforward. Still, using all of the features in a rather large set usually provided better performance than subsets of the features (HUTCIS_SVM_FULLIMG_ALL vs. HUTCIS_SVM_FULLIMG_IP+SC and HUTCIS_SVM_FULLIMG_IP), with some notable exceptions, especially query "motorbike". This is in line with our general observation (and common knowledge) that without specific knowledge of the target objects, an acceptable solution can often be found by blindly fusing a large number of features.

In general, it was found better to train with features extracted from whole images instead of just bounding boxes (e.g. HUTCIS_SVM_FULLIMG_IP+SC and HUTCIS_SVM_BB_BB_IP+SC), with possible exception in the query "person". This is no surprise given the unsymmetry in our feature extraction and matching: the features extracted from bounding boxes of the training objects were compared with the features of all of the test images. The bounding box technique does not even seem to give much complementary information in addition to the full image information, as fusing these approaches (HUTCIS_SVM_FULLIMG+BB) usually results in worse performance than using the full images alone.

The results of the SOM classifier runs did not provide information that would be of general interest, besides confirming the previously known result of SOM classifiers being inferior to SVMs.

3.3 INAOE's Annotation-based object retrieval approaches

Authors: Heidy Marisol Marin Castro, Hugo Jair Escalante Balderas, and Carlos Arturo Hernández Gracidas

Affiliation: TIA Research Group, Computer Science Department, National Institute of Astrophysics, Optics and Electronics, Tonantzintla, Mexico

Email: {hmarinc,hugojair,carloshg}@ccc.inaoep.mx

The *TIA* research group at *INAOE*, Mexico proposed two methods based on image labeling. Automatic image annotation methods were used for labeling regions within segmented images, and then we performed object retrieval based on the generated annotations. Two approaches were proposed: a semi-supervised classifier based on unlabeled data and a supervised one, where the latter method was enhanced by a recently proposed method based on semantic cohesion [20]. Both approaches followed the following steps:

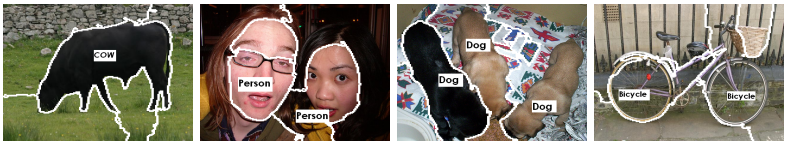

Fig. 4. Sample images from the generated training set. .

1. Image segmentation
2. Feature extraction
3. Manual labeling of a small subset of the training set
4. Training a classifier
5. Using the classifier for labeling the test-images
6. Using labels assigned to region images for object retrieval

For both approaches the full collection of images was segmented with the normalized cuts algorithm [21]. A set of 30 features were extracted from each region; we considered color, shape and texture attributes. We used our own tools for image segmentation, feature extraction and manual labeling [22]. The considered annotations were the labels of the 10 objects defined for this task. The features for each region together with the manual annotations for each region were used as the training set with the two approaches proposed. Each classifier was trained with this dataset and then all of the test images were annotated with such a classifier. Finally, the generated annotations were used for retrieving objects with queries. Queries were created using the labels of the objects defined for this task; and selected as relevant those images with the highest number of regions annotated with the object label. Sample segmented images with their corresponding manual annotations are shown in Figure 4. As we can see the segmentation algorithm works well for some images (isolated cows, close-up of people), however for other objects segmentation is poor (a bicycle, for example).

***KNN+MRFI*, A supervised approach:** For the supervised approach we used a simple *knn* classifier for automatically labeling regions. Euclidean distance was used as the similarity function. The label of the nearest neighbor (in the training set) for each test-region was assigned as annotation for this region. This was our baseline run (*INAOE-TIA-INAOE-RB-KNN*).

The next step consisted of improving the annotation performance of *knn* using an approach called *MRFI* [20] which we recently proposed for improving annotation systems. This approach consists of modeling each image (region-annotations pairs) with a Markov random field (*MRF*), introducing semantic knowledge, see Figure 5. The top$-k$ more likely annotations for each region are considered. Each of these annotations has a confidence weight related to the relevance of the label to being the correct annotation for that region, according to *knn*. The *MRFI* approach uses the relevance weights with semantic information for choosing a unique (the correct) label for each region. Semantic information

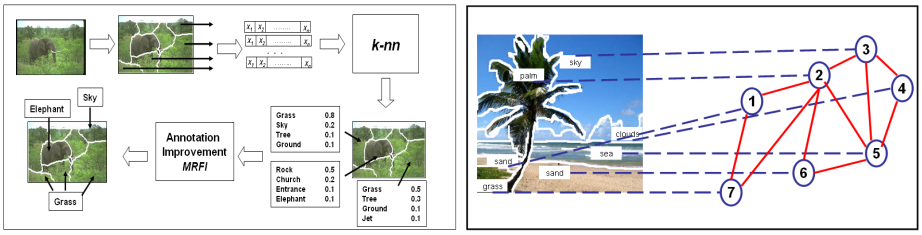

Fig. 5. Left: graphical description of the improvement process of *MRFI*. Right: interpretation of *MRFI* for a given configuration of labels and regions; (red) line-arcs consider semantic cohesion between labels, while (blue) dashed-arcs consider relevance weight of each label according to $k-nn$.

is considered in the *MRF* for keeping coherence among annotations assigned to regions within a common image; while the relevance weight is considered for taking into account the confidence of the annotation method ($k-nn$) on each of the labels, see Figure 5. The (pseudo) optimal configuration of region-annotations for each image is obtained by minimizing an energy function defined by potentials. For optimization we used standard simulated annealing.

The intuitive idea of the *MRFI* approach is to guarantee that the labels assigned to regions are coherent among themselves, taking into account semantic knowledge and the confidence of the annotation system. In previous work, semantic information was obtained from cooccurrences of labels on an external corpus. However for this work semantic association between a pair of labels is given by the normalized number of relevant documents returned by Google^R to queries generated using the pair of labels. This run is named *INAOE-TIA-INAOE-RB-KNN+MRFI*, see [20] for details.

SSAssemble: Semi-supervised Weighted AdaBoost: The semi-supervised approach consists of using a recently proposed ensemble of classifiers, called WSA [22]. Our WSA ensemble uses naive Bayes as its base classifier. A set of these is combined in a cascade based on the AdaBoost technique [23]. Ensemble methods work by combining a set of base classifiers in some way, such as a voting scheme, producing a combined classifier which usually outperforms a single classifier. When training the ensemble of Bayesian classifiers, WSA considers the unlabeled images at each stage. These are annotated based on the classifier from the previous stage, and then used to train the next classifier. The unlabeled instances are weighted according to a confidence measure based on their predicted probability value; while the labeled instances are weighted according to the classifier error, as in standard AdaBoost. Our method is based on the supervised multi-class AdaBoost ensemble, which has shown to be an efficient scheme to reduce the error rate of different classifiers.

Formally the WSA algorithm receives a set of labeled data (L) and a set of unlabeled data (U). An initial classifier NB_1 is build using L. The labels in L are used to evaluate the error of NB_1. As in AdaBoost the error is used to weight the

examples, increasing the weight of the misclassified examples and keeping the same weight of the correctly classified examples. The classifier is used to predict a class for U with certain probability. In the case of U, the weights are multiplied by the predicted probability of the majority class. Unlabeled examples with high probability of their predicted class will have more influence in the construction of the next classifier than examples with lower probabilities. The next classifier NB_2 is build using the weights and predicted class of $L \cup U$. NB_2 makes new predictions on U and the error of NB_2 on all the examples is used to reweight the examples. This process continues, as in AdaBoost, for a predefined number of cycles or when a classifier has a weighted error greater than or equal to 0.5. As in AdaBoost, new instances are classified using a weighted sum of the predicted class of all the constructed base classifiers. WSA is described in Algorithm 2.

We faced several problems when performing the annotation image task. The first one was that the training set and the test set were different, so this caused a classification with high error ratio. The second one was due the segmentation algorithm. The automatic segmentation algorithm did not perform well for all images leading to incorrect segmentation of the objects in the images. The last one concerns the different criteria for manual labeling of the training set. Due to all these facts we did not get good results. We hope to improve the annotation task by changing part of the labeling strategy.

3.4 MSRA: Object Retrieval

Authors: Mingjing Li, Xiaoguang Rui, and Lei Wu
Affiliation: Microsoft Research Asia
Email: mjli@microsoft.com

Two approaches were adopted by Microsoft Research Asia (MSRA) to perform the object retrieval task in ImageCLEF 2007. One is based on the visual topic model (VTM); the other is the visual language modelling (VLM) method [24]. VTM represents an image by a vector of probabilities that the image belongs to a set of visual topics, and categorizes images using SVM classifiers. VLM represents an image as a 2-D document consisting of visual words, trains a statistical language model for each image category, and classifies an image to the category that generates the image with the highest probability.

VTM: Visual Topic Model: Probabilistic Latent Semantic Analysis (pLSA) [25], which is a generative model from the text literature, is adopted to find out the visual topics from training images. Different from traditional pLSA, all training images of 10 categories are put together in the training process and about 100 visual topics are discovered finally.

The training process consists of five steps: local feature extraction, visual vocabulary construction, visual topic construction, histogram computation, and classifier training. At first, salient image regions are detected using scale invariant interest point detectors such as the Harris-Laplace and the Laplacian detectors. For each image, about 1,000 to 2,000 salient regions are extracted. Those regions

Algorithm 2 Semi-supervised Weighted AdaBoost (WSA) algorithm.

Require: L: labeled instances, U: unlabeled instances, P: training instances, T: Iterations

Ensure: Final Hypothesis and probabilities: $H_f = \text{argmax}\sum_{t=1}^{T} log\frac{1}{B_t}$, $P(x_i)$

$W(x_i)^0 = \frac{1}{NumInst(L)}, \forall x_i \in L$
for t from 1 to T **do**
  $W(x_i)^t = \frac{W(x_i)}{N} \forall x_i \in L$
  $\sum_{i=1} W(x_i)$
  $h_t = C(L, W(x_i)^t)$
  $e_t = \sum_{i=1}^{N} W(x_i)^t$ if $h_t(x_i) \neq y_i$
  if $e_t \geq 0.5$ **then**
    exit
  end if
  if $e_t = 0.0$ **then**
    $e_t = 0.01$
  end if
  $B_t = \frac{e_t}{(1-e_t)}$
  $W(x_i)^{(t+1)} = W(x_i)^t * B_t$ if $h_t(x_i) = y_i \ \forall x_i \in L$
  $P(x_i) = C(L, U, W(x_i)^t)$
  $W(x_i) = P(x_i) * B_t \ \forall x_i \in U$
end for

are described by the SIFT descriptor which computes a gradient orientation histogram within the support region. Next, 300 local descriptors are randomly selected from each category and combined together to build a global vocabulary of 3,000 visual words. Based on the vocabulary, images are represented by the frequency of visual words. Then, pLSA is performed to discover the visual topics in the training images. pLSA is also applied to estimate how likely an image belongs to each visual topic. The histogram of the estimated probabilities is taken as the feature representation of that image for classification. For multi-class classification problem, we adopt the one-against-one scheme, and train an SVM classifier with RBF kernel for each possible pair of categories.

VLM: Visual Language Modeling: The approach consists of three steps: image representation, visual language model training and object retrieval. Each image is transformed into a matrix of visual words. First, an image is simply segmented into 8×8 patches, and the texture histogram feature is extracted from each patch. Then all patches in the training set are grouped into 256 clusters based on their features. Next, each patch cluster is represented using an 8-bit hash code, which is defined as the visual word. Finally, an image is represented by a matrix of visual words, which is called a *visual document*.

Visual words in a visual document are not independent to each other, but correlated with other words. To simplify the model training, we assume that visual words are generated in the order from left to right, and top to bottom and each word is only conditionally dependent on its immediate top and left neighbors, and train a trigram language model for each image category. Given a test image, it is transformed into a matrix of visual words in the same way, and the probability that it is generated by each category is estimated respectively. Finally, the image categories are ranked in the descending order of these probabilities.

3.5 NTU: Solution for the Object Retrieval Task

Authors:	Steven C. H. Hoi
Affiliation:	School of Computer Engineering, Nanyang Technological University, Singapore
Email:	chhoi@ntu.edu.sg

Introduction: Object retrieval is an interdisciplinary research problem between object recognition and content-based image retrieval (CBIR). It is commonly expected that object retrieval can be solved more effectively with the joint maximization of CBIR and object recognition techniques. We study a typical CBIR solution with application to the object retrieval tasks [26, 27]. We expect that the empirical study in this work will serve as a baseline for future research when using CBIR techniques for object recognition.

Overview of Our Solution: We study a typical CBIR solution for the object retrieval problem. In our approach, we focus on two key tasks. One is the feature representation, the other is the supervised learning scheme with support vector machines.

Feature Representation: In our approach, three kinds of global features are extracted to represent an image, including color, shape, and texture.

For color, we study the Grid Color Moment feature (GCM). We split each image into a 3×3 grid and extract color moments to represent each of the 9 regions of the grid. Three color moments are then computed: color mean, color variance and color skewness in each color channel (H, S, and V), respectively. Thus, an 81-dimensional color moment is adopted as the color feature for each image.

For shape, we employ the edge direction histogram. First, an input color image is converted into a grayscale image. Then a Canny edge detector is applied to obtain its edge image. Based on the edge images, the edge direction histogram can be computed. Each edge direction histogram is quantized into 36 bins of 10 degrees each. In addition, we use a bin to count the number of pixels without edges. Hence, a 37-dimensional edge direction histogram is used for shape.

For texture, we investigate the Gabor feature. Each image is first scaled to the size of 64×64. Then, the Gabor wavelet transformation is applied to the scaled image at 5 scale levels and 8 orientations, which results in a total of 40 subimages for each input image. For each subimage, we calculate three statistical moments to represent the texture, including mean, variance, and skewness. Therefore, a 120-dimensional feature vector is used for texture.

In total, a 238-dimensional feature vector is used to represent each image. The set of visual features has been shown to be effective for content-based image retrieval in our previous experiments [26, 27].

Supervised Learning for Object Retrieval: The object retrieval task defined in ImageCLEF 2007 is similar to a relevance feedback task in CBIR, in which a number of positive and negative labeled examples are given for learning. This can be treated as a supervised classification task. To solve it, we employ the support vector machines (SVM) technique for training the classifiers on the given examples [26]. In our experiment, a standard SVM package is used to train the SVM classifier with RBF kernels. The parameters C and γ are best tuned on the VOC 2006 training set, in which the training precision is 84.2% for the classification tasks. Finally, we apply the trained classifiers to do the object retrieval by ranking the distances of the objects from the classifier's decision boundary.

Concluding Remarks: We found that the current solution, though it was trained with good performance in an object recognition test-bed, did not achieve promising results in the tough object retrieval tasks. In our future work, several directions can be explored to improve the performance, including local feature representation and better machine learning techniques.

3.6 PRIP: Color Interest Points and SIFT features

Authors:	Julian Stöttinger[1], Allan Hanbury[1], Nicu Sebe[2], Theo Gevers[2]
Affiliation:	[1] PRIP, Institute of Computer-Aided Automation, Vienna University of Technology, Vienna, Austria; [2] Intelligent Systems Lab Amsterdam, University of Amsterdam, The Netherlands
Email:	{julian,hanbury}@prip.tuwien.ac.at, {nicu,gevers}@science.uva.nl

In the field of retrieval, detection, recognition and classification of objects, many state of the art methods use interest point detection at an early stage. This initial step typically aims to find meaningful regions in which descriptors are calculated. Finding salient locations in image data is crucial for these tasks. Most current methods use only the luminance information of the images. This approach focuses on the use of color information in interest point detection and its gain in performance. Based on the Harris corner detector, multi-channel visual information transformed into different color spaces is the basis to extract the most salient interest points. To determine the characteristic scale of an interest

point, a global method of investigating the color information on a global scope is used. The two PRIP runs differ in the properties of these interest points only. The method consists of the following stages:

1. Extraction of multi-channel based interest points
2. Local descriptions of interest points
3. Estimating the signature of an image
4. Classification

Extraction of multi-channel based interest points: An extension of the intensity-based Harris detector [28] is proposed in [29]. Because of common photometric variations in imaging conditions such as shading, shadows, specularities and object reflectance, the components of the RGB color system are correlated and therefore sensitive to illumination changes. However, in natural images, high contrast changes may appear. Therefore, a color Harris detector in RGB space does not dramatically change the position of the corners compared to a luminance based approach. Normalized rgb overcomes the correlation of RGB and favors color changes. The main drawback, however, is its instability in dark regions. We can overcome this by using quasi invariant color spaces.

The approach PRIP-PRIP_HSI_ScIvHarris uses the HSI color space [30], which is quasi-invariant to shadowing and specular effects. Therefore, changes in lighting conditions in images should not affect the positions of the interest points, resulting in more stable locations. Additionally, the HSI color space discriminates between luminance and color. Therefore, much information can be discarded, and the locations get more sparse and distinct.

The PRIP_cbOCS_ScIvHarris approach follows a different idea. As proposed in [31], colors have different occurrence probabilities and therefore different information content. Therefore, rare colors are regarded as more salient than common ones. We use a boosting function so that color vectors having equal information content have equal impact on the saliency function. This transformation can be found by analyzing the occurrence probabilities of colors in large image databases. With this change of focus towards rare colors, we aim to discard many repetitive locations and get more stable results on rare features.

The characteristic scale of an interest point is chosen by applying a principal component analysis (PCA) on the image and thus finding a description for the correlation of the multi-channel information [32]. The characteristic scale is decided when the Laplacian of Gaussian function of this projection and the Harris energy is a maximum at the same location in the image. The final extraction of these interest points and corresponding scales is done by preferring locations with high Harris energy and large scales. A maximum number of 300 locations per image has been extracted, as over-description diminishes the overall recognition ability.

Local descriptions of interest points: The scale invariant feature transform (SIFT) [33] showed to give best results in a broad variety of applications [34].

We used the areas of the extracted interest points as a basis for the description phase. SIFT are basically sampled and normalized gradient histograms, which can lead to multiple descriptions per location. This occurs if there is more than one direction of the gradients regarded as predominant.

Estimating the signature of an image: In this bag of visual features approach [35], we cluster the descriptions of one image to a fixed number of 40 clusters using k-means. The centroids and the proportional sizes of the clusters build the signature of one image having a fixed dimensionality of 40 by 129.

Classification: The Earth Mover's Distance (EMD) [36] showed to be a suitable metric for comparing image signatures. It takes the proportional sizes of the clusters into account, which gains much discriminative power. The classification itself is done in the most straightforward way possible: for every object category, the smallest distances to another signature indicate the classification.

3.7 RWTHi6: Patch-Histograms and Log-Linear Models

Authors: Thomas Deselaers, Hermann Ney
Affiliation: Human Language Technology and Pattern Recognition, RWTH Aachen University, Aachen, Germany
Email: surname@cs.rwth-aachen.de

The approach used by the Human Language Technology and Pattern Recognition group of the RWTH Aachen University, Aachen, Germany, to participate in the PASCAL Visual Object Classes Challenge consists of four steps:

1. patch extraction
2. clustering
3. creation of histograms
4. training of a log-linear model

where the first three steps are feature extraction steps and the last is the actual classification step. This approach was first published in [37, 38].

The method follows the promising approach of considering objects to be constellations of parts which offers the immediate advantages that occlusions can be handled very well, that the geometrical relationship between parts can be modelled (or neglected), and that one can focus on the discriminative parts of an object. That is, one can focus on the image parts that distinguish a certain object from other objects.

The steps of the method are briefly outlined in the following paragraphs. To model the difference in the training and test data, the first three steps have been done for the training and test data individually, and then the corresponding histograms have been extracted for the respective other, so that the vocabulary was learnt once for the training data and once for the test data, and the histograms are created for each using both vocabularies. Results however show that this seems not to be a working approach to tackle divergence in training and testing data.

Patch Extraction: Given an image, we extract square image patches at up to 500 image points. Additionally, 300 points from a uniform grid of 15×20 cells that is projected onto the image are used. At each of these points a set of square image patches of varying sizes (in this case $7 \times 7, 11 \times 11, 21 \times 21,$ and 31×31 pixels) are extracted and scaled to a common size (in this case 15×15 pixels).

In contrast to the interest points from the detector, the grid-points can also fall onto very homogeneous areas of the image. This property is on the one hand important for capturing homogeneity in objects which is not found by the interest point detector and on the other hand it captures parts of the background which usually is a good indicator for an object, as in natural images objects are often found in a "natural" environment.

After the patches are extracted and scaled to a common size, a PCA dimensionality reduction is applied to reduce the large dimensionality of the data, keeping 39 coefficients corresponding to the 40 components of largest variance but discarding the first coefficient corresponding to the largest variance. The first coefficient is discarded to achieve a partial brightness invariance. This approach is suitable because the first PCA coefficient usually accounts for global brightness.

Clustering: The data are then clustered using a k-means style iterative splitting clustering algorithm to obtain a partition of all extracted patches. To do so, first one Gaussian density is estimated which is then iteratively split to obtain more densities. These densities are then re-estimated using k-means until convergence is reached and then the next split is done. It has been shown experimentally that results consistently improve up to 4096 clusters but for more than 4096 clusters the improvement is so small that it is not worth the higher computational demands.

Creation of Histograms: Once we have the cluster model, we discard all information for each patch except its closest corresponding cluster center identifier. For the test data, this identifier is determined by evaluating the Euclidean distance to all cluster centers for each patch. Thus, the clustering assigns a cluster $c(x) \in \{1, \dots C\}$ to each image patch x and allows us to create histograms of cluster frequencies by counting how many of the extracted patches belong to each of the clusters. The histogram representation $h(X)$ with C bins is then determined by counting and normalization such that $h_c(X) = \frac{1}{L_X} \sum_{l=1}^{L_X} \delta(c, c(x_l))$, where δ denotes the Kronecker delta function, $c(x_l)$ is the closest cluster center to x_l, and x_l is the l-th image patch extracted from image X, from which a total of L_X patches are extracted.

Training & Classification: Having obtained this representation by histograms of image patches, we define a decision rule for the classification of images. The approach based on maximum likelihood of the class-conditional distributions does not take into account the information of competing classes during training. We can use this information by maximizing the class posterior probability

$\prod_{k=1}^{K} \prod_{n=1}^{N_k} p(k|X_{kn})$ instead. Assuming a Gaussian density with pooled covariances for the class-conditional distribution, this maximization is equivalent to maximizing the parameters of a log-linear or maximum entropy model

$$p(k|h) = \frac{1}{Z(h)} \exp\left(\alpha_k + \sum_{c=1}^{C} \lambda_{kc} h_c\right),$$

where $Z(h) = \sum_{k=1}^{K} \exp\left(\alpha_k + \sum_{c=1}^{C} \lambda_{kc} h_c\right)$ is the renormalization factor. We use a modified version of generalized iterative scaling. Bayes' decision rule is used for classification.

4 Results

The results of this task published in [9] were shown to have several problems due to unclear relevance judgement guidelines and invalid submission files (e.g. wrong query order) [10].

Therefore a thorough analysis of all submitted runs was performed for this work and the results presented here differ in part significantly from those presented in [9]. In particular,

- all runs were carefully checked to fully comply with the latest version of `trec_eval` and to deliver a maximum of 1,000 results per class;
- based on the full annotation of the database by Ville Viitaniemi [10], the pooling was re-done and new relevance judgements were created as they would have been if judging guidelines would have been more clear and all runs would have had proper formatting.

The results presented here are all fully comparable except for the two runs from the Budapest group. They assigned one class-label per image instead of possibly several ones (e.g. there may be a bicycle and a person in an image). Furthermore they used different, more strongly labelled training data.

Table 4 gives results for all runs using the relevance judgements obtained from simulated pooling and Table 5 gives the same results but uses the relevance information for the whole database. The tables are ordered by MAP (last column). The ordering, however should not be interpreted as a general ranking of the methods since the methods perform very differently among the different topics.

5 Discussion

In this section, the results for the full database annotation are discussed in more detail. However most of the observations can also be found in the results obtained using the simulated pooling.

Considering the class-wise results, it can be observed that the best overall results were obtained for the *car* query (column 3), for which the best run has an

Table 4. Results from the ImageCLEF 2007 object retrieval task using the relevance judgements obtained from simulated pooling. All values have been multiplied by 100 to make the table more readable. The numbers in the top row refer to the class id's (see Table 1). The MAP over all classes is in the last column. The highest AP per class is shown in bold.

	query										
run id	1	2	3	4	5	6	7	8	9	10	MAP
HUTCIS_SVM_FULLIMG_ALL	**18.7**	**4.0**	**22.7**	1.7	0.0	2.4	0.8	**13.1**	0.0	18.5	**9.1**
HUTCIS_SVM_FULLIMG_IP+SC	9.1	2.9	21.7	4.3	0.0	4.1	1.4	11.6	0.0	18.8	8.2
HUTCIS_SVM_FULLIMG_IP	8.2	3.1	20.2	**8.6**	0.0	**4.6**	0.7	11.5	0.0	16.2	8.1
HUTCIS_SVM_FULLIMG+BB	12.1	3.7	9.5	2.7	0.0	2.4	**2.1**	8.7	0.0	20.7	6.9
HUTCIS_SVM_BB_ALL	6.0	3.3	1.7	1.4	0.0	2.2	0.8	6.0	0.0	**22.7**	4.9
HUTCIS_SVM_BB_BB_IP+SC	5.2	3.3	2.2	1.6	0.0	1.5	0.4	4.0	0.0	22.5	4.5
HUTCIS_SVM_BB_FULL_IP+SC	8.3	2.3	1.0	0.9	0.0	2.6	0.4	4.0	0.0	20.7	4.5
HUTCIS_SVM_BB_BAL_IP+SC	4.8	2.7	1.7	1.0	0.0	1.5	0.9	2.5	0.0	22.4	4.2
HUTCIS_SVM_BB_BB_IP	3.9	1.6	0.9	7.0	0.0	1.0	0.3	2.8	0.0	16.9	3.8
HUTCIS_PICSOM1	2.9	2.4	12.3	2.0	0.0	0.9	0.7	1.2	0.0	10.4	3.6
HUTCIS_SVM_BB_BAL_IP	3.8	2.2	0.7	1.5	0.0	0.9	0.8	2.5	0.0	17.7	3.3
HUTCIS_PICSOM2	1.6	2.3	12.1	1.6	0.0	0.6	1.0	0.8	0.0	9.1	3.2
MSRA-MSRA_RuiSp	2.7	1.4	7.5	2.4	2.3	0.1	0.9	0.4	0.0	10.6	3.1
HUTCIS_SVM_BB_FULL_IP	0.4	2.7	0.5	1.3	0.0	0.9	0.3	2.9	0.0	13.9	2.5
NTU_SCE_HOI-NTU_SCE_HOI_1	4.2	2.0	4.5	0.0	0.0	0.0	0.3	0.1	0.0	0.2	1.3
RWTHi6-HISTO-PASCAL	0.4	0.4	1.3	0.6	0.0	0.1	0.0	0.2	0.0	7.1	1.1
budapest-acad-budapest-acad314	0.4	0.1	1.2	0.0	0.0	0.6	0.5	0.6	0.0	5.7	1.0
budapest-acad-budapest-acad315	2.0	0.0	0.6	0.5	0.0	0.0	0.0	0.0	0.0	5.3	1.0
PRIP-PRIP_HSI_ScIvHarris	0.3	0.0	0.4	0.1	**4.6**	0.5	0.0	0.0	0.0	1.5	0.8
MSRA-MSRA-VLM_8_8_640_ful	0.7	0.6	0.8	0.3	0.0	0.0	0.2	0.8	0.0	2.6	0.7
MSRA-MSRA-VLM-8-8-800-HT	1.1	0.4	0.7	0.2	0.0	0.0	0.0	0.5	0.0	2.2	0.6
INAOE-TIA-INAOE-RB-KNN+MRFI	0.7	0.0	0.5	0.0	0.0	0.6	0.0	0.0	0.0	2.5	0.5
INAOE-TIA-INAOE-RB-KNN+MRFI_ok	0.7	0.0	0.5	0.0	0.0	0.6	0.0	0.0	0.0	2.5	0.5
INAOE-TIA-INAOE-RB-KNN	0.0	0.0	0.3	0.2	0.0	0.0	0.0	0.0	0.0	3.3	0.4
PRIP-PRIP_cbOCS_ScIvHarr2	0.1	0.0	0.1	1.5	0.2	0.0	0.0	0.0	0.0	0.6	0.3
INAOE-TIA-INAOE_SSAssemble	0.4	0.0	0.0	0.0	0.0	0.2	0.0	0.1	0.0	0.6	0.2

AP of about 11%. This can clearly be useful in a practical application. The best run for the *bicycle* class (column 1) is also able to find enough relevant images to be useful in a practical application.

It is also clear that the three classes for which the best APs are significantly high (car, bicycle and person) are also those classes with the highest number of relevant images (Table 1). This could be because having a high number of relevant images in the dataset means that they have a higher probability of being detected by chance. However, the bad performance on these classes by some of the methods also provides evidence against this conjecture.

The results for the classes having fewer relevant images in the dataset are less easy to interpret and generalise. For the *bus* class (column 2), although the better runs are able to find a few images showing buses, these images are not ranked very highly. By joining all runs, only 140 of the 218 images in the database that show buses are found. The best run for the *cat* class (column 5) obtains an average precision of 1.4% which can already be considered a promising result given the extremely low number of relevant images in the database. The best run for the *cow* query (column 6) finds 12 out of 49 relevant images. However, only one of the images is among the top 10 retrieved. For queries 7 (*dog*) and

Table 5. Results from the ImageCLEF 2007 object retrieval task with complete relevance information for the whole database. All values have been multiplied by 100 to make the table more readable. The numbers in the top row refer to the class id's (see Table 1). The MAP over all classes is in the last column. The highest AP per class is shown in bold.

	query										
run id	1	2	3	4	5	6	7	8	9	10	MAP
HUTCIS_SVM_FULLIMG_ALL	**4.1**	**1.2**	10.6	0.4	0.0	0.6	0.1	**3.8**	0.0	8.3	2.9
HUTCIS_SVM_FULLIMG_IP+SC	2.6	1.0	**11.1**	1.0	0.0	1.0	0.1	3.2	0.0	8.2	2.8
HUTCIS_SVM_FULLIMG_IP	2.4	1.1	10.3	**1.8**	0.0	**1.1**	0.1	3.0	0.0	8.1	2.8
HUTCIS_SVM_FULLIMG+BB	3.0	1.1	4.2	0.6	0.0	0.7	**0.1**	2.5	0.0	**8.6**	2.1
HUTCIS_SVM_BB_ALL	1.6	0.9	0.5	0.3	0.0	0.6	0.1	1.5	0.0	8.3	1.4
HUTCIS_SVM_BB_BB_IP+SC	1.4	1.0	0.7	0.3	0.0	0.5	0.1	1.1	0.0	8.4	1.4
HUTCIS_SVM_BB_FULL_IP+SC	2.0	0.8	0.4	0.2	0.0	0.8	0.1	1.1	0.0	8.2	1.3
HUTCIS_PICSOM1	0.9	0.7	4.5	0.6	0.0	0.3	0.1	0.7	0.0	5.6	1.3
MSRA-MSRA_RuiSp	0.9	0.5	3.6	0.6	0.7	0.1	0.1	0.4	0.0	6.0	1.3
HUTCIS_SVM_BB_BAL_IP+SC	1.3	0.8	0.5	0.2	0.0	0.5	0.1	0.8	0.0	8.4	1.3
HUTCIS_PICSOM2	0.8	0.6	4.2	0.5	0.0	0.3	0.1	0.4	0.0	5.4	1.2
HUTCIS_SVM_BB_BB_IP	1.1	0.7	0.4	1.4	0.0	0.3	0.0	1.0	0.0	7.2	1.2
HUTCIS_SVM_BB_BAL_IP	1.1	0.8	0.3	0.3	0.0	0.4	0.1	0.9	0.0	6.9	1.1
HUTCIS_SVM_BB_FULL_IP	0.3	0.9	0.3	0.3	0.0	0.3	0.0	1.1	0.0	6.6	1.0
RWTHi6-HISTO-PASCAL	0.4	0.2	1.4	0.2	0.0	0.1	0.0	0.2	0.0	5.5	0.8
budapest-acad-budapest-acad314	0.1	0.1	0.8	0.0	0.0	0.2	0.0	0.2	0.0	4.1	0.5
NTU_SCE_HOI-NTU_SCE_HOI_1	1.2	0.7	2.4	0.0	0.0	0.0	0.1	0.1	0.0	0.8	0.5
budapest-acad-budapest-acad315	0.4	0.0	0.4	0.1	0.0	0.0	0.1	0.1	0.0	3.9	0.5
MSRA-MSRA-VLM_8_8_640_ful	0.4	0.3	0.7	0.1	0.1	0.0	0.0	0.3	0.0	2.5	0.4
MSRA-MSRA-VLM-8-8-800-HT	0.3	0.2	0.5	0.0	0.0	0.1	0.0	0.2	**0.1**	2.5	0.4
INAOE-TIA-INAOE_SSAssemble	0.1	0.0	0.1	0.0	0.0	0.2	0.0	0.2	0.0	3.2	0.4
INAOE-TIA-INAOE-RB-KNN+MRFI	0.5	0.1	0.6	0.0	0.0	0.2	0.0	0.0	0.0	2.2	0.4
INAOE-TIA-INAOE-RB-KNN+MRFI_ok	0.5	0.1	0.6	0.0	0.0	0.2	0.0	0.0	0.0	2.2	0.4
PRIP-PRIP_HSI_ScIvHarris	0.1	0.0	0.3	0.1	**1.4**	0.1	0.0	0.0	0.0	1.5	0.4
INAOE-TIA-INAOE-RB-KNN	0.3	0.0	0.5	0.1	0.0	0.0	0.0	0.0	0.0	2.2	0.3
PRIP-PRIP_cbOCS_ScIvHarr2	0.1	0.0	0.1	0.5	0.1	0.0	0.1	0.1	0.0	0.8	0.2

9 (*sheep*), the best runs have an AP of 0.1%, which is not high enough for use in a practical application. The best result for the *horse* query (column 8) finds 49 of the 175 relevant images in the database.

The *person* query is certainly to be treated differently than all other queries since the number of relevant images for this query is higher than the allowed number of results returned. The best run returns 984 images showing persons which is 98.4% of the optimal result. Several other runs find more than 800 relevant images. However, all runs jointly only found 6029 relevant images which is an indicator that most runs found similar, and probably "easier" images. While these algorithms produce promising results, they are not suitable for finding all images showing a person.

One issue that should be taken into account when interpreting the results is that about 50% of the evaluated runs are from HUTCIS and thus this group had a significant impact on the pools for relevance assessment. In the initial evaluation, this effect was further boosted by the fact that the initial runs from HUTCIS (which were used for the pooling) had the queries 4–10 in wrong order. This problem, was fixed in the evaluation here by simulating proper pooling using the annotation of the complete database. However, it can still be observed that the

high number of HUTCIS runs makes them appear slightly better in the pooled results than in the results using the full database annotation. Additionally, we evaluated all runs with a different pooling strategy: the pooling was simulated with only one run per group, which removes the bias introduced by strongly differing numbers of submissions. Here we observed a ranking that is more similar to the ranking obtained when the annotation of the full database is used.

By comparing the results with and without pooling, it can be observed that pooling changes the results, however using the additional relevance information obtained during judging the pools, a more stable result can be obtained. The effect of pooling is particular strong for runs with only very few relevant images and for runs with very many relevant images.

The results clearly show that the task is a very difficult one and that it is very important to clearly define judging criteria and relevance assessment methods before running the evaluation. In particular it seems to be important to ensure an appropriate (not too few, not too many) number of relevant images per topic.

6 Conclusion

We presented the object retrieval task of ImageCLEF 2007, the methods of the participating groups, and the results. The main challenges inherent in the task were the difference in the nature of the images used for training and testing, as well as the large variation in the number of relevant images for each query, ranging from 6 to 11,248 of 20,000 images. The results show that none of the methods really solves the assigned task. Although large advances in object detection and recognition were achieved over the last years, still many improvements are necessary to solve difficult tasks with a high variability and only a restricted amount of training data. It can however be observed that some of the methods are able to obtain reasonable results for a limited set of classes. An interesting observation is that few participating groups attempted to compensate for the differences in training and testing data, while the few attempts made were in general not successful.

Furthermore, the analysis of the results showed that the use of pooling techniques for relevance assessment can be problematic if the pools are biased due to erroneous runs or due to many strongly correlated submissions as it was the case in this evaluation.

Acknowledgements

We would like to thank the CLEF campaign for supporting the ImageCLEF initiative. This work was partially funded by the European Commission MUSCLE NoE (FP6-507752) and the DFG (German research foundation) under contract Ne-572/6.

We would like to thank the PASCAL NoE for allowing us to use the training data of the PASCAL 2006 Visual object classes challenge as training data for this task, in particular, we would like to thank Mark Everingham for his cooperation.

We would like to thank Paul Clough from Sheffield University for support in creating the pools and Jan Hosang, Jens Forster, Pascal Steingrube, Christian Plahl, Tobias Gass, Daniel Stein, Morteza Zahedi, Richard Zens, Yuqi Zhang, Markus Nušbaum, Gregor Leusch, Michael Arens, Lech Szumilas, Jan Bungeroth, David Rybach, Peter Fritz, Arne Mauser, Saša Hasan, and Stefan Hahn for helping to create relevance assessments for the images.

The work of the Budapest group was supported by a Yahoo! Faculty Research Grant and by grants *MOLINGV* NKFP-2/0024/2005, NKFP-2004 project Language Miner.

References

1. Everingham, M., et al.: The 2005 PASCAL Visual Object Classes Challenge. In: Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment (PASCAL Workshop 05). Number 3944 in LNAI, Southampton, UK (2006) 117–176
2. Everingham, M., Zisserman, A., Williams, C., Gool, L.V.: The Pascal Visual Object Classes Challenge 2006 (VOC2006) results. Technical report (2006) Available online at `http://www.pascal-network.org/`.
3. Clough, P.D., Müller, H., Sanderson, M.: Overview of the CLEF cross–language image retrieval track (ImageCLEF) 2004. In Peters, C., Clough, P.D., Jones, G.J.F., Gonzalo, J., Kluck, M., Magnini, B., eds.: Multilingual Information Access for Text, Speech and Images: Result of the fifth CLEF evaluation campaign. Lecture Notes in Computer Science, Bath, England, Springer–Verlag (2005)
4. Moellic, P.A., Fluhr, C.: ImageEVAL 2006 official campaign. Technical report, ImagEVAL (2006)
5. Grubinger, M., Clough, P., Hanbury, A., Müller, H.: Overview of the ImageCLEF 2007 photographic retrieval task. In: Working Notes of the 2007 CLEF Workshop, Budapest, Hungary (2007)
6. Müller, H., Deselaers, T., Kim, E., Kalpathy-Cramer, J., Deserno, T.M., Clough, P., Hersh, W.: Overview of the ImageCLEFmed 2007 medical retrieval and annotation tasks. In: Working Notes of the 2007 CLEF Workshop, Budapest, Hungary (2007)
7. Grubinger, M., Clough, P., Müller, H., Deselaers, T.: The IAPR benchmark: A new evaluation resource for visual information systems. In: LREC 06 OntoImage 2006: Language Resources for Content-Based Image Retrieval, Genoa, Italy (2006) in press
8. Braschler, M., Peters, C.: CLEF methodology and metrics. In: C. Peters (Ed.), Cross-language information retrieval and evaluation: Proceedings of the CLEF2001 Workshop, Lecture Notes in Computer Science 2406, Springer Verlag. (2002) 394–404
9. Deselaers, T., Hanbury, A., Viitaniemi, V., Benczúr, A., Brendel, M., Daróczy, B., Escalante Balderas, H.J., Gevers, T., Hernández Gracidas, C.A., Hoi, S.C.H., Laaksonen, J., Li, M., Marin Castro, H.M., Ney, H., Rui, X., Sebe, N., Stöttinger, J., Wu, L.: Overview of the ImageCLEF 2007 object retrieval task. In: Working notes of the CLEF 2007 Workshop, Budapest, Hungary (2007)
10. Viitaniemi, V., Laaksonen, J.: Thoughts on evaluation of image retrieval inspired by ImageCLEF 2007 object retrieval task. In: MUSCLE / ImageCLEF Workshop on Image and Video Retrieval Evaluation, Budapest, Hungary (2007)

11. Chen, Y., Wang, J.Z.: Image categorization by learning and reasoning with regions. Journal of Machine Learning Research **5** (2004) 913–939
12. Prasad, B.G., Biswas, K.K., Gupta, S.K.: Region-based image retrieval using integrated color, shape, and location index. Computer Vision and Image Understanding **94**(1-3) (2004) 193–233
13. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. PAMI **24**(8) (2002) 1026–1038
14. Lv, Q., Charikar, M., Li, K.: Image similarity search with compact data structures. In: CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, New York, NY, USA, ACM Press (2004) 208–217
15. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. International Journal of Computer Vision **59** (2004)
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22** (2000) 888–905
17. Viitaniemi, V., Laaksonen, J.: Improving the accuracy of global feature fusion based image categorisation. In: Proceedings of the 2nd International Conference on Semantic and Digital Media Technologies (SAMT 2007). Lecture Notes in Computer Science, Genova, Italy, Springer (2007) 1–14
18. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
19. Laaksonen, J., Koskela, M., Oja, E.: PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing **13**(4) (2002) 841–853
20. Escalante, H.J., y Gómez, M.M., Sucar, L.E.: Word co-occurrence and MRFs for improving automatic image annotation. In: In Proceedings of the 18th British Machine Vision Conference (BMVC 2007), Warwick, UK (September, 2007)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI **22**(8) (2000) 888–905
22. Marin-Castro, H.M., Sucar, L.E., Morales, E.F.: Automatic image annotation using a semi-supervised ensemble of classifiers. to appear. In: 12th Iberoamerican Congress on Pattern Recognition CIARP 2007. Lecture Notes in Computer Science, Viña del Mar, Valparaiso, Chile, Springer (2007)
23. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning. (1996) 148–156
24. Wu, L., Li, M.J., Li, Z.W., Ma, W.Y., Yu, N.H.: Visual language modeling for image classification. In: 9th ACM SIGMM International Workshop on Multimedia Information Retrieval, (MIR'07), Augsburg, Germany (2007)
25. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: International Conference on Computer Vision, Beijing, China (2005)
26. Hoi, S.C.H., Lyu, M.R.: A novel log-based relevance feedback technique in content-based image retrieval. In: 12th ACM International Conference on Multimedia (MM 2004), New York, NY, USA (2004) 24–31
27. Hoi, S.C., Lyu, M.R., Jin, R.: A unified log-based relevance feedback scheme for image retrieval. IEEE Transactions on Knowledge and Data Engineering **18**(4) (2006) 509–524
28. Harris, C., Stephens, M.: A combined corner and edge detection. In: 4th Alvey Vision Conference. (1988) 147–151

29. Montesinos, P., Gouet, V., Deriche, R.: Differential invariants for color images. In: ICPR. (1998) 838
30. van de Weijer, J., Gevers, T.: Edge and corner detection by photometric quasi-invariants. PAMI **27**(4) (2005) 625–630
31. van de Weijer, J., Gevers, T., Bagdanov, A.: Boosting color saliency in image feature detection. PAMI **28**(1) (2006) 150–156
32. Stöttinger, J., Hanbury, A., Sebe, N., Gevers, T.: Do colour interest points improve image retrieval? ICIP (2007) to appear.
33. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2) (2004) 91–110
34. Mikolaczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI **27**(10) (2005) 1615–1630
35. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. CWPR **73**(2) (2006) 213–238
36. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. IJCV **40**(2) (2000) 99–121
37. Deselaers, T., Keysers, D., Ney, H.: Discriminative training for object recognition using image patches. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., San Diego, CA (2005) 157–162
38. Deselaers, T., Keysers, D., Ney, H.: Improving a discriminative approach to object recognition using image patches. In: DAGM 2005, Pattern Recognition, 26th DAGM Symposium. Number 3663 in LNCS, Vienna, Austria (2005) 326–333