The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results

Mark Everingham
University of Oxford
me@robots.ox.ac.uk

Andrew Zisserman
University of Oxford

Chris Williams
University of Edinburgh

Luc Van Gool
KU Leuven

September 11, 2006

Abstract

This report presents the results of the 2006 PASCAL Visual Object Classes Challenge (VOC2006). Details of the challenge, data, and evaluation are presented. Participants in the challenge submitted descriptions of their methods, and these have been included *verbatim*. This document should be considered preliminary, and subject to change.

Contents

1 **Challenge** 4

1.1 Classification Task 4

1.2 Detection Task 4

1.3 Timetable 4

1.4 Image Sets 4

1.4.1 Database Rights 5

1.4.2 Ground Truth Annotation 5

1.5 Competitions 6

1.6 Evaluation 6

1.6.1 Classification Task 6

1.6.2 Detection Task 7

2 **Participants** 7

2.1 AP06_Batra 7

2.2 AP06_Lee 9

2.3 Cambridge 9

2.4 ENSMP 10

2.5 INRIA_Douze 12

2.6 INRIA_Laptev 12

2.7 INRIA_Larlus 13

2.8 INRIA_Marszalek 15

2.9 INRIA_Moosmann 16

2.10 INRIA_Nowak 16

2.11 INSARouen 17

2.12 KUL 17

2.13 MIT_Fergus 19

2.14 MIT_Torralba 19

2.15 MUL 20

2.16 QMUL 21

2.17 RWTH_DiscHist 23

2.18 RWTH_GMM 24

2.19 RWTH_SparseHists 25

2.20 Siena 25

2.21 TKK 26

2.22 TUD 27

2.23 UVA_big5 29

2.24 UVA_weibull 29

2.25 XRCE 30

2.26 ROUND2_INRIA_Moosmann 30

3 **Results: Classification** 31

3.1 Competition 1 31

3.2 Competition 2 43

4 **Results: Detection** 46

4.1 Competition 3 46

4.2 Competition 4 53

5 Acknowledgements **56**

A Annotation Guidelines **56**

A.1 Guidelines on what and how to label 56

A.2 Guidelines on categorization 57

A.3 “Difficult” flag 57

1 Challenge

The goal of this challenge was to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects). Ten object classes are annotated in the data provided:

- bicycle, bus, car, motorbike
- cat, cow, dog, horse, sheep
- person

There were two tasks:

1.1 Classification Task

For each of the ten object classes predict the presence/absence of at least one object of that class in a test image. The output from your system should be a real-valued confidence of the object's presence so that an ROC curve can be drawn.

1.2 Detection Task

For each of the ten classes predict the bounding boxes of each object of that class in a test image (if any). Each bounding box should be output with an associated real-valued confidence of the detection so that a precision/recall curve can be drawn.

1.3 Timetable

The challenge was run according to the following timetable:

- 14 February 2006: Development kit (training, validation data and software) made available.
- 31 March 2006: Test set (without annotation) made available.
- 27 April 2006: Deadline for submission of results.

After completion of the main challenge, a "second round" was run for which participants were invited to submit additional results. This round was judged separately from the main challenge. The deadline for submission of second round results was 30 June 2006.

1.4 Image Sets

There were four sets of images provided, for use in both the classification and detection tasks.

- `train`: Training data
- `val`: Validation data (suggested). The validation data may be used as additional training data (see below).
- `trainval`: The union of `train` and `val`.

Table 1: Statistics of the image sets

	`train`		`val`		`trainval`		`test`	
	img	**obj**	**img**	**obj**	**img**	**obj**	**img**	**obj**
Bicycle	127	161	143	162	270	323	268	326
Bus	93	118	81	117	174	235	180	233
Car	271	427	282	427	553	854	544	854
Cat	192	214	194	215	386	429	388	429
Cow	102	156	104	157	206	313	197	315
Dog	189	211	176	211	365	422	370	423
Horse	129	164	118	162	247	326	254	324
Motorbike	118	138	117	137	235	275	234	274
Person	319	577	347	579	666	1156	675	1153
Sheep	119	211	132	210	251	421	238	422
Total	1277	2377	1341	2377	2618	4754	2686	4753

- `test`: Test data.

Table 1 summarizes the number of objects and images (containing at least one object of a given class) for each class and image set. The data is split into 50% for training/validation and 50% for testing. The distributions of images and objects by class are approximately equal across the training/validation and test sets. In total there are 5,304 images, containing 9,507 annotated objects.

1.4.1 Database Rights

The challenge database contains images provided by Microsoft Research Cambridge and collected from the photo-sharing web-site "flickr". Use of these images must respect the corresponding terms of use. These are available via the challenge web pages: `http://www.pascal-network.org/challenges/VOC/voc2006/`.

1.4.2 Ground Truth Annotation

Objects of the ten classes listed above are annotated in the ground truth. For each object, the following annotation is present:

- **class**: the object class e.g. 'car' or 'bicycle'
- **bounding box**: an axis-aligned rectangle specifying the extent of the object visible in the image.
- **view**: 'frontal', 'rear', 'left' or 'right'. The views are subjectively marked to indicate the view of the 'bulk' of the object. Some objects have no view specified.
- **'truncated'**: an object marked as 'truncated' indicates that the bounding box specified for the object does not correspond to the full extent of the object. Truncation may occur for two reasons: i) the object extends outside the image e.g. an image of a person from the waist up; ii) the boundary of the object is occluded e.g. a person standing behind a wall.

- **'difficult'**: an object marked as 'difficult' indicates that the object is considered difficult to recognize, for example an object which is clearly visible but unidentifiable without substantial use of context. Objects marked as dificult were *ignored* in the evaluation of the challenge.

In preparing the ground truth, annotators were given a detailed list of guidelines on how to complete the annotation. These are reproduced in Appendix A.

1.5 Competitions

Four competitions were defined according to the task and the choice of training data: (i) taken from the VOC `trainval` data provided, or (ii) from any source excluding the VOC `test` data provided:

No.	**Task**	**Training data**	**Test data**
1	Classification	`trainval`	`test`
2	Classification	**any but** VOC `test`	`test`
3	Detection	`trainval`	`test`
4	Detection	**any but** VOC `test`	`test`

Any annotation provided in the VOC `train` and `val` sets could be used for training, for example bounding boxes or particular views e.g. 'frontal' or 'side'. Participants were free to perform manual annotation on the training data if they wished. Manual annotation of the test data to optimize algorithm performance was *not* permitted.

In competitions 2 and 4, any source of training data could be used *except* the provided `test` images. Researchers who had pre-built systems trained on other data were particularly encouraged to participate. The test data includes images from the Microsoft Research Cambridge object recognition database, and "flickr" (`www.flickr.com`); these sources of images could *not* be used for training.

For each competition, participants could choose to tackle all, or any subset of object classes, for example "cars only" or "motorbikes and cars".

1.6 Evaluation

Participants were expected to submit a *single* set of results per method employed. Participants who investigated several algorithms were allowed to submit one result per method. Changes in algorithm parameters do *not* constitute a different method – all parameter tuning had to be conducted using the training and validation data alone.

1.6.1 Classification Task

The classification task was judged by the Receiver Operating Characteristic (ROC) curve. The principal quantitative measure used was the area under curve (AUC). Example code for computing the ROC and AUC measure is provided in the development kit.

Images which contain only objects marked as 'difficult' (section 1.4.2) were *ignored* by the evaluation.

1.6.2 Detection Task

The detection task was judged by the precision/recall curve. The principal quantitative measure used was the average precision (AP) as used by TREC. The average precision is defined thus: for 11 thresholds on recall $r \in \{0, 0.1, \ldots, 0.9, 1\}$ the *interpolated* precision $\tilde{p}(r)$ is computed and the arithmetic mean taken. The interpolated precision $\tilde{p}(r)$ is defined as the *maximum* precision for which the corresponding recall is greater than or equal to the threshold r. Example code for computing the precision/recall and AP measure is provided in the development kit.

Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, the area of overlap a_o between the predicted bounding box B_p and ground truth bounding box B_{gt} must exceed 50% by the formula:

$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{1}$$

Example code for computing this overlap measure is provided in the development kit. Multiple detections of the *same* object in an image are considered *false* detections e.g. 5 detections of a single object is counted as 1 correct detection and 4 false detections – it is the responsibility of the participant's system to filter multiple detections from its output.

Objects marked as 'difficult' (section 1.4.2) were *ignored* by the evaluation.

2 Participants

This section lists (in no significant order) the participants in the challenge who submitted final results. Each participant has been assigned an identifier based on the institution and the corresponding author, which is referred to in all results figures and tables. A description of the method used has been provided by each participant and is reproduced here.

2.1 AP06_Batra

Participants: Dhruv Batra, Gunhee Kin, Alexei Efros
Affiliation: Carnegie Mellon University,
Advanced Perception class 16-721
E-mail: `batradhruv@cmu.edu`

We draw a distinction between two kinds of classes in the VOC2006 database, first kind being the "structured" classes (car, bus, bicycle, motorbike), the second being "unstructured/deformable" classes (person, cat, cow, horse, dog, sheep). We make an observation that while the former possess strict geometry which is rarely deformed, the latter can be treated as a texture recognition problem with consistency in their background acting as context. We work on two different algorithms to harness these two consistencies.

Method 1 (for "structured" classes). We pose this problem as a local feature matching problem between the test images and the annotated training images (Hand segmented to exact boundaries by us). The feature detector used was Lowe's DOG [1]. The local features we experimented with were Yan Ke's PCA-SIFT [2] and actual patches cut out of images at proper scales in Gaussian pyramids. To boost the matching over a NN based scheme, and to incorporate spatial and geometric constraints in the matched local features, we used M. Leordeanu et al's spectral correspondence [3] based matching. Scores were generated which denoted consistency of matched features. This method, although robust to small intra-class shape variances, requires a certain class geometry to be preserved, and this is precisely why we cannot use this method for highly deformable classes.

Method 2 (for "unstructured" classes). We use color and texture histograms after generating patches by over-segmentation using Jianbo Shi's Normalized cuts [4] (75 patches are used an image). For each patch, we extract 64-D RGB histogram as a color descriptor, and 48-D outputs of Leung-Malik Filter banks as a texture descriptor. In the training step, we generate the color and texture histograms of all patches in the all images of the training set. We reject background by manually labeled masks and store only descriptors of patches on the target object. In the matching step, for each patch of a test image, we find the nearest neighbor descriptors from a training set, and then assign the nearest object class. And then, we make a histogram through voting of all patches. The scores are derived from these voting numbers. The underlying assumption of this voting based object classification is that most patches on the target object are successfully identified, whereas the background patches are randomly matched.

We would like to thank Antonio Torralba, Bryan Russell, and the organizers of PASCAL VOC 2006 for letting us use the LabelMe tool to segment and label the PASCAL VOC training set (as our algorithm required perfect segmentations).

References

[1] D. G. Lowe. Object recognition from local scale-invariant features, ICCV 1999.

[2] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors, CVPR2004.

[3] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints, ICCV2005.

[4] J. Shi and J. Malik. Normalized Cuts and Image Segmentation, PAMI 2000.

[5] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons, IJCV 2001.

2.2 AP06_Lee

Participants: David Changsoo Lee, Nik Melchior, Alexei Efros
Affiliation: Carnegie Mellon University,
Advanced Perception class 16-721
E-mail: `dlee1@andrew.cmu.edu`

We have used the Bag-Of-Words approach. An image is divided into small patches and classification is performed for each patch. The final classification is decided by the vote of the classification result of all patches.

Descriptor. For each 16x16 pixel patch, descriptors are computed as follows.

- Color: Top 2 colors in the RGB histogram
- Texture: Histogram of 32 textons. (textons adopted from Martin01)
- Histogram of Oriented Gradients: Proposed by Dalal05. 16x16 pixel patch is divided into four 8x8 regions. Each 8x8 region gives 18 dimensional vector, concatenating 4 regions gives a 72 dimensional HOG descriptor.

A total of 7 combination of these 3 descriptors are used. (color, texture, HOG, color+texture, color+HOG, texture+HOG, color+texture+HOG)

Training. We divide the bounding box of an object into dense overlapping multiscale patches. We collect all the patches extracted from the bounding box of the object of interest and quantize them into 300 clusters using K-means. The same process is performed on image with object of interest including the background to obtain additional 300 clusters. Finally, the same process is applied to images that do not contain the object of interest. This is done for all 7 descriptors.

Testing. A classification of a patch is done by determining whether the given patch is closer to a positive cluster or a negative cluster. There are two weak classifiers per each descriptor, one trained on the bounding box and one trained with background, and there are 7 descriptors, so a total of 14 weak classifiers are applied to a test patch. A weighted sum of these 14 weak classifiers gives the final confidence. Weights for weak classifiers are determined by the error on test set, similar to Adaboost. The average of all the confidence of patches in a query image is used for determining the final confidence of an image.

2.3 Cambridge

Participants: Jamie Shotton
Affiliation: University of Cambridge
E-mail: `jamieshotton@gmail.com`

The TextonBoost algorithm [1] was used with minor modifications reflecting the considerably different problem being posed in the VOC2006 as compared with the original work. These modifications were as follows:

1. Training images each had an automatic GrabCut [2] process applied to convert the bounding box to an approximate segmentation. These segmentations were used as the "ground truth" for training the classifier.

2. A "forest" of 10 TextonBoost classifiers was learned on subsets of the training data, training on all classes simultaneously, and also including a 'background' class. The classification results of each classifier were averaged together.

3. Only the shape-texture potentials in the model are used, and no graph-cuts are run, only the boosted classifier; hence a distribution over class labels is obtained at each pixel.

4. For the classification challenge, the confidence value is given as $\frac{1}{N}\sum_i p(c_i = c|I)$ where N is the number of pixels in image I, and $p(c_i = c|I)$ is the probability that the inferred class label at pixel i is the class in question c.

5. For the detection challenge, the maximum a-posteriori class labels are found, and contiguously segmented regions which contain at least 1000 pixels are found. The smallest rectangle that bounds all pixels of a given region forms a detection.

References

[1] J. Shotton, J. Winn, C. Rother and A. Criminisi. TextonBoost: Joint appearance, shape and context modelling for multi-class object recognition and segmentation. In Proc. ECCV, pages I:1–15, 2006.

[2] C. Rother, V. Kolmogorov and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics (SIGGRAPH'04), 2004.

2.4 ENSMP

Participants: Fabien Moutarde
Affiliation: Robotics Laboratory, Ecole des Mines de Paris
E-mail: `fabien.moutarde@ensmp.fr`

The detectors used are obtained as boosted assemblies of simple visual features, as described in the 3 papers included in this directory, respectively published or accepted in:

[1] "YEF real-time object detection", Y. Abramson, B. Steux and H. Ghorayeb, Proc. of Intl. Workshop on Automatic Learning and Real-Time (ALART05), Siegen, Germany (2005).

[2] "SEmi-automatic VIsuaL LEarning (SEVILLE): a tutorial on active learning for visual object recognition", Y. Abramson and Y. Freund, Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR05), San Diego (2005)

[3] “Combining AdaBoost with a Hill-Climbing evolutionary feature search for efficient training of performant visual object detectors”, Y. Abramson, F. Moutarde, B. Steux and B. Stanciulescu, accepted for presentation at FLINS2006 conference on Applied Computational Intelligence (Genova, Italy, 29-31 August 2006).

More precisely, the car detections were obtained as simple *union* of detections from 2 different detectors:

- a “lateral-view car detector”
- a “rear-front-view car detector”

The first one examines rectangles with width= 2.3∗height, while the second one examines square areas. In both cases, positive examples were generated by automatically extracting (thanks to annotations files) from the trainval set images about 10 small images per annotated object, all centered on the object but with slightly randomized margin around and center offset, and cropped to normalize to desired width/height ratio; then only “acceptable” images (i.e. with the good car orientation, not truncated, etc..) were selected. The result was:

- one set of ∼2500 positive rectangular images of “clean” lateral-views of cars,
- one set of ∼1500 positive square images of “clean” rear-or-front-views of cars.

Two small (typically 100-200) initial sets of negative examples of various sizes (but the 2 respective desired width/height ratio) were manually extracted from trainval images. Those initial negative sets were then semi-automatically enriched by the iterative procedure described in [2] (i.e. building first crude detectors to generate false positive detections to be added to negative examples, retrain with the bigger training set, etc...). The typical size of the negative set at the final stage (i.e. the one used for training the final detector evaluated) is 5000 to 6000. All examples (positive or negative) are subimages of trainval images (surrounded by a black border). Each final detector assembles 600 weighted “Control-Point” features (see [1] for details), and it scans the image at various resolutions in order to detect object of any size bigger than the smallest resolution (i.e. 46x20 pixels for the lateral car detector, 32x32 pixels for the rear-front car detector).

The cow detector is obtained in the same way, but uses only one detector trained to detect only lateral-views of standing-up cows (in subimages with width/height=1.4, and bigger than minimum size 56x40). The final training set used for training the final detector contains 2300 positive rectangular images of “clean” lateral-views of standing-up cows, and 5000 negative examples mostly collected semi-automatically. All examples (positive or negative) are subimages of trainval images (surrounded by a black border). The cow-detector performance is clearly lower than the car detector, as lateral-viewed-standing-up cows represent only a relatively small part of all annotated cows. Also, there is obviously some confusion with sheeps, horses, dogs (or even cats) viewed in the same “lateral” position, and it is probable that including *explicitly* in the negative examples a significant number of ROIs centered on those "confusing" objects would lead to a better performance, but it was too late to execute and evaluate one more training before challenge deadline... It is also very probable that some "union of detectors" strategy (i.e. using also a front-viewed-cows detector, and simply cumulate the detections of both cow detectors) would significantly improve the global detection rate, as in the car case, but again we did not have time to finish this before challenge deadline.

Note: the typical computation time for each detector on a standard 3Ghz desktop is around 0.5 to 7 seconds per image (depending on the image size, and on the minimum detection window size).

2.5 INRIA_Douze

Participants: Matthijs Douze, Navneet Dalal
Affiliation: INRIA Rhones-Alpes
E-mail: matthijs.douze@inrialpes.fr

We used the same method as that of the 2005 Pascal Challenge (described in "Histograms of Oriented Gradients for Human Detection", Navneet Dalal and Bill Triggs, CVPR05).

For the comp3_* results, the learning was done on the bounding boxes not marked as difficult or truncated. The ground truth orientation was not taken into account. The results on the classes cat, dog and horse were too bad to be significant.

comp4_det_test_person.txt was trained on our own person dataset. On the validation dataset it performed better than the corresponding comp3 result, presumably thanks to more appropriate annotations.

2.6 INRIA_Laptev

Participants: Ivan Laptev
Affiliation: IRISA / INRIA Rennes
E-mail: ivan.laptev@inria.fr

We detect objects using a window-scanning approach. Each rectangular window of the image is classified into an object or non-object using AdaBoost classifier [2]. Inspired by the success of histogram-based descriptors for recognition [1,5,7,8], we use histograms of gradient orientation as image features. Each histogram feature is computed for a particular rectangular region in the object window. A complete set of such features is used to train AdaBoost cascade classifier. For the detection, a strong classifier based on a subset of selected histogram features is evaluated rapidly using integral histograms [6]. The speed of the current implementation is approximately three frames per second on 640 × 480 images. While the method is conceptually similar to [4], we use several extensions to improve on the quality of detection. These extensions concern (i) AdaBoost weak learner in terms of Fisher linear discriminant, (ii) construction of histogram features and (iii) preparation of the training data. The details of the method are available in [3].

We consider task 3 of the PASCAL VOC'06 Challenge and apply the method to five object classes *'bicycle', 'cow', 'horse', 'motorbike'* and *'person'*. For motorbikes we combine results of the side-view and the frontal-view detector. For bicycles, cows and horses we train side-view detectors only. For people, only subjects in standing postures are considered for training. We apply detectors to image sub-windows densely sampled over positions and scales. Multiple responses of the same detector are clustered in the position-scale space. The size of resulting clusters is used as a confidence measure of the detection. Due to the similarity of classes 'cow' and 'horse' as well as 'motorbike' and 'bicycle' we systematically reduced the confidence of detections with high scores of similar classifiers.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Proc. CVPR, pages I:886-893, 2005.

[2] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119-139, 1997.

[3] I. Laptev. Improvements of object detection using boosted histograms. In Proc. BMVC, 2006.

[4] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In Proc. CVPR, pages II:5360, 2004.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, page to appear, 2004.

[6] F. M. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In Proc. CVPR, pages I:829836, 2005.

[7] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. IJCV, 36(1):3150, January 2000.

[8] M. J. Swain and D. H. Ballard. Color indexing. IJCV, 7(1):1132, November 1991.

2.7 INRIA_Larlus

Participants: Diane Larlus, Frederic Jurie
Affiliation: INRIA Rhone-Alpes
E-mail: `diane.larlus@inrialpes.fr`

The following method has already competed with success in competition 1 (classification task in test1) during last year challenge.

Our method is based on an SVM classifier trained on feature vectors built using local image descriptors. Our approach is purely appearance based, i.e.

it does not explicitly use the local structures of object classes. The learning consists of four steps. First, we extract local image features using a dense multi-scale representation. Our novel clustering method is then applied to build a codebook of visual words. This codebook is used to compute “bag of features” representation for each image, similar to [2], then an SVM classifier is trained to separate between object images and the background (the other classes of the database). In the following we describe in detail each step of our method.

Feature extraction Overlapping local features are extracted on each scale according to a regular grid defined to be sufficiently dense to represent the entire image. Our parameters are set to extract approximately 10000 regions per image. Each region is then represented by a 128 dimensional SIFT descriptor [5], i.e. a concatenated 8-bin orientation histograms on a 4x4 grid.

Codebook creation The extracted set of dense features has two important properties. First, it is very populated; the large amount features per image leads to a total number of few hundreds of thousands for the entire training set (train+val). Second, the dense feature set is extremely unbalanced as it was shown in [3]. Therefore, to obtain a discrete set of labels on the descriptors we have designed a new clustering algorithm [4] taking into account these properties. The method has two main advantages. It can discover low populated regions of the descriptor space, and it can easily cope with large amount of descriptors.

Our iterative approach discovers new clusters at each step by consecutively calling a sampling and a k-median algorithm until the required total number of clusters are found. In order to decrease the importance of highly populated regions we use biased sampling: new regions are discovered far enough from previously found centers. This is realized by introducing an *influence radius* to affect points close to already found centers. All affected descriptors are then excluded from any further sampling. The influence radius determines an affectation ball around each center. All descriptors within these balls are removed and the remaining portion is then random sampled. The influence radius ($r = 0.6$) and the total number of clusters ($k = 5000$) are parameters of our method.

The biased sampling is followed by the *online median* algorithm proposed by Mettu and Plaxton [6]. Their method is based on the *facility location* problem and chooses the centers one by one. At each iteration of our algorithm we discover 50 new centers by this algorithm.

Image quantization Both learning and testing images are represented by the *bag of features* approach [2], i.e by frequency histograms computed using the occurrence of each visual word of our codebook. We associate each descriptor to the closest codebook element. To measure the distance between SIFT features we used the Euclidean distance as in [5].

Classification We used the implementation of [1] to train linear SVM classifiers on the normalized image histograms.

References

[1] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[2] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorizationwith bags of keypoints. In ECCV04 workshop on Statistical Learning in ComputerVision, pages 5974, 2004.

[3] F. Jurie and W. Triggs. Creating efficient codebooks for visual recognition. Proceedings of the 9th International Conference on Computer Vision, Beijing, China, 2005.

[4] D. Larlus. Creation de vocabulaires visuels efficaces pour la categorisation dimages. Masters thesis, Image Vision Robotic, INPG and UJF, June 2005.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2):91110, November 2004.

[6] R. R. Mettu and C. G. Plaxton. The online median problem. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, page 339. IEEE Computer Society, 2000.

2.8 INRIA_Marszalek

Participants: Marcin Marszalek[1], Jianguo Zhang[2], Cordelia Schmid[1]
Affiliation: [1]INRIA Rhone-Alpes; [2]Queen Mary, University of London
E-mail: marcin.marszalek@inrialpes.fr

The submitted results were obtained using an extended version of our local features and kernels framework [7]. We start by finding a sparse set of salient image regions using the Harris-Laplace [3] and the Laplacian [1] interest point detectors. The two sets of interest points are kept separately and form two channels. The local regions are described by the SIFT [2] descriptor combined with a local hue-histogram [5]. The images are represented as a histogram of visual words drawn from a vocabulary, resulting in a bag-of-features representation [6]. A problem-specific vocabulary is constructed by separately clustering features from the positive and the negative training examples. The classification is performed with a non-linear Support Vector Machine [4]. We have used channels combination and χ^2 kernel as in [7].

References

[1] T. Lindeberg. Feature detection with automatic scale selection. IJCV, 30(2), 1998.

[2] D. Lowe. Distinctive image features form scale-invariant keypoints. IJCV, 60(2), 2004.

[3] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. IJCV, 60(1), 2004.

[4] B. Scholkopf and A. Smola. Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge, MA, 2002.

[5] J. van de Weijer and C. Schmid. Coloring local feature extraction. In ECCV, 2006.

[6] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In IWLAVS, 2004.

[7] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV, 2006. To appear.

2.9 INRIA_Moosmann

Participants: Frank Moosmann
Affiliation: INRIA Rhone-Alpes
E-mail: frank.moosmann@inrialpes.fr

The method used follows the method described in [1]. First random subwindows are extracted from the training images (random size, random position) and resized to 16x16 pixel. Then a wavelet-transform (Haar basis functions) is performed on each color channel. Extremely randomized trees are then used to cluster the labeled training features in a supervised way. The leaf nodes form the visual vocabulary. A linear SVM is afterwards trained with binarized histograms, created from the training images. The extraction of features is done as before (but on the whole image) and the PDF responsible for the selection of the random windows is adjusted after each selection and propagation through the trees depending on the output of the trees. This leads to more features to be extracted in regions where the object is estimated. To build the trees 50000 features were extracted in total. To create histograms 10000 features per image were used.

References

[1] F. Moosmann, D. Larlus and F. Jurie, Learning Saliency Maps for Object Categorization, ECCV International Workshop on The Representation and Use of Prior Knowledge in Vision, 2006

2.10 INRIA_Nowak

Participants: Eric Nowak
Affiliation: INRIA Rhone-Alpes
E-mail: eric.nowak@inrialpes.fr

The method is the one described in: Eric Nowak, Frederic Jurie, Bill Triggs, Sampling Strategies for Bag-of-Features Image Classification, European Conference on Computer Vision 2006.

It is based on the standard Bag-Of-Features algorithm, except:

- 10000 interest points are randomly sampled per image.
- a large generic codebook (4000 elements) is built with online k-means.
- feature matrices are normalized with an adaptive thresholding.

2.11 INSARouen

Participants: F. Suard, Alain Rakotomamonjy
Affiliation: INSA de Rouen
E-mail: `alain.rakoto@insa-rouen.fr`

The approach we have used is based on dense sift representation of images. Each image is represented according to a feature vector which is based on the occurrence of a codebook sift representation in the image. The codebook has been build iteratively by using a L1 penalized learning algorithm (Least Angle Regression).

Given a learning and a validation set, we have selected at random a set of patches and use them as a codebook, we learn the decision function and then keep the set of patches that maximizes the AUC. and we iterate this procedure a 100 times.

The learning algorithm is a Least Angle Regression algorithm.

2.12 KUL

Participants: Alexander Thomas[1], Vittorio Ferrari[3], Bastian Leibe[2]
Tinne Tuytelaars[1], Bernt Schiele[2], Luc Van Gool[1]
Affiliation: [1]KU Leuven, [2]TU Darmstadt, [3]INRIA Rhones-Alpes
E-mail: `athomas@esat.kuleuven.be`

This method is a combination of the ISM model by Leibe & Schiele [2] and the multi-view object recognition system from Ferrari et al. [3]. The goal of this system is to greatly reduce the computational cost of a battery of separate detectors trained on multiple views, while achieving equal or better performance. The training set for this system consists of images from several different instances of the object class. For each object, there need to be at least a few images from different viewpoints (which is why we can only tackle competitions 2 and 4). Viewpoints are roughly aligned to a set of reference viewpoints. For the motorbike model that was used for this challenge, 16 viewpoints were used in a circle around the motorbikes (i.e. the views are 22.5 degrees apart). There are 41 motorbikes in the training set (11 more than in [1]), with an average of 11 views per motorbike. For each image, a segmentation is provided which separates figure from background (this segmentation may be quite rough).

The first stage of training a model with our system, is to use the methods from [3] to derive multi-view region tracks for each training object. Next, for each viewpoint an ISM model like in [2] is trained, yielding a battery of detectors. The details of these two steps are not explained here, as they can be reviewed in their respective publications [2,3]. One key concept from the ISM system, however, are so-called occurrences. For each entry in the appearance codebook of an ISM, a set of occurrences is stored. An occurrence is the relative position to the center of the training object, where that codebook entry matched in the training image. The next step in constructing the multi-view model, is finding relations between the ISM models, by using the multi-view tracks. If two regions in a multi-view track match sufficiently with two occurrences in the same two views of the same object, an activation link is established between these two occurrences. The final model consists of the set of ISM models, together with their activation links.

The recognition procedure works as follows. First, the codebooks for all the ISM models (16 in the case of our motorbike model) are matched to features extracted from the test image. Like in [2], each ISM then casts votes in its own voting space, based on the occurrences. Based on initial hypotheses for object instances in these voting spaces, we select a set of candidate viewpoints that are likely to correspond to the pose of the object in the image. This is done by looking for clusters of hypotheses across neighboring viewpoints. Due to similarity between adjacent viewpoints, a strong hypothesis in the correct viewpoint will also produce fairly strong hypotheses at similar positions in its neighboring viewpoints. Next, we only consider each candidate view separately. We try to transfer evidence from each other view into that candidate view, using the activation links. This procedure is called 'vote transfer'. Let C be the candidate view and A another view. If an occurrence was activated in A, and it has links to occurrences in C, these occurrences are activated and cast votes in C. The position of the vote is the position in which the original occurrence was activated, plus the linked occurrence's vote coordinates. The weight for a transferred vote is calculated in a similar statistical fashion as the weights for regular votes. After vote transfer, we re-detect local maxima in the voting spaces of each candidate view, and perform the MDL selection procedure from [2]. This produces a set of final hypotheses. The confidence score for each hypothesis is the MDL savings score. Because the final hypotheses often overlap, we apply a simple overlap removal procedure, based on the bounding boxes. A weaker hypothesis is rejected when its bounding box overlaps more than a certain percentage with the one from a stronger hypothesis. We intend to improve this procedure by using the same MDL approach as is currently used in each view separately, but at the time of this writing the implementation is not finished yet.

Classification results were simply derived from the detection results, by taking the maximum detection score for all hypotheses in the image.

References

[1] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, L. Van Gool, Towards Multi-View Object Class Detection, to appear in the proceedings of the Conference on Computer Vision and Pattern Recognition, New York, 2006.

[2] B. Leibe and B. Schiele. Scale-Invariant Object Categorization using a Scale-Adaptive Mean-Shift Search, DAGM, pp. 145-153, 2004.

[3] V. Ferrari, T. Tuytelaars, and L. Van Gool, Integrating Multiple Model Views for Object Recognition, CVPR, Vol. II, pp. 105-112, 2004.

2.13 MIT_Fergus

Participants: Rob Fergus
Affiliation: CSAIL, MIT
E-mail: fergus@csail.mit.edu

Method. Constellation Model based on CVPR '05 paper.

Training. Due to time constaints, I did not train any models on the PASCAL 2006 training data. Instead, I used the models I applied to the PASCAL 2005 challenge (see my thesis for results on those datasets).

Motorbike model - The motorbike model was trained on the PASCAL 2005 training data. Various combinations of feature types were tried, but the Kadir and Brady alone gave the best performance. A 6-part model, 30 detections per image model was used. These are conservative settings but the small size of many instances hindered the use of the large model. Increasing the number of detections/image increased the false alarm rate significantly while giving a modest reduction in the false negative rate due to the difficulty in finding small instances.

Car model - The Car model was trained on the Caltech Cars Rear dataset. Although a cars side model was also trained for use on the 2005 data, I didn't use it on the 2006 data since combining the models was fiddly and the side view models was conserably weaker than the rear view model. The car rear model has 6 parts with 3 using Kadir & Brady features and 3 using multiscale harris. Again, 30 detections per image (of each type) were used for the same reasons as above. The model seems to have learnt the shadow under the car, which may explain why it also find instances in different viewpoints that which it was trained on.

Validation. Both models were run once on the validation set, just to check I hadn't screwed something up with the feature extraction. No settings changes were made.

Test. Both models were run once on the test data. Glancing at the results, the motorbike model seems to get very confused by bicycles.

2.14 MIT_Torralba

Participants: Antonio Torralba
Affiliation: CSAIL, MIT
E-mail: torralba@csail.mit.edu

We construct a view invariant car detector by training a set of classifiers each tuned to one specific view point (we use 12 different viewpoints). The classifiers for each view points can share features with the others in order to increase efficiency and improving generalization. Features are based on convolution with filtered image patches/fragments. For those image locations in which the detector is above the detection threshold, we can estimate the pose of the object by looking at the classifier with the maximal response. The different aspect ratios of the bounding boxes correspond to the hypothesized car orientations.

The classifier is trained on 12 views of cars from the LabelMe dataset (50 positive examples for each view and 12860 background samples) and uses 300 shared features. The classifier was trained for the 2005 PASCAL dataset and has not been tuned or retrained since. We have just taken the same classifier and applied it to the new dataset. Images are scaled, before running the detector, to have a maximum size of 150 pixels in the vertical dimension. The bounding box of the object has a vertical size of 24 pixels (for all viewpoints). Targets smaller than that dimension (after image scaling) will not be detected. If there are more than three detections in one image, only the three most confident are retained.

References

[1] A. Torralba, K. P. Murphy and W. T. Freeman. (2004). Sharing features: efficient boosting procedures for multiclass object detection. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp 762–769.

2.15 MUL

Participants: Martin Antenreiter
Affiliation: Montanuniversitat Leoben
E-mail: `martin.antenreiter@unileoben.ac.at`

It is not always clear which feature types are advisable for learning a certain class, therefore we use 10 different types of features. We use the scale invariant Harris-Laplace detector from K. Mikolajczyk and C. Schmid, a segmentation algorithm from Pedro F. Felzenszwalb and Daniel P. Huttenlocher and another segmentation algorithm from M. Fussenegger et al. to obtain regions of interest. Our ten features are:

1. SIFT features from David Lowe
2. PCA-SIFT: SIFT features reduced to their 40 largest components using PCA
3. Sub-sampled grayvalues
4. Sub-sampled grayvalues with region normalization
5. Basic moments
6. Basic moments with region normalization
7. Moment invariants

8. Moment invariants with region normalization
9. Segments with color information and blob size.
10. Segments with textural moments 'Object Recognition Using Segmentation for Feature Detection'.

To reduce computational efforts we cluster each feature type using a k-means cluster algorithm. A boosting approach is used as learning method, due to the fact, that boosting algorithms are able to select important features from a large feature set.

Learning Method 1: One-vs-One. Method 1 learned 90 one-vs-one classifiers for the 10-class problem with LPBoost. The used weak learner selects a reference feature with an optimal threshold. Finally, we address the multi-class classification problem and provide a weight optimization method for the one-vs-one classifiers using Support Vector Machines (SVMs).

Learning Method 2: One-vs-All. The method 2 learned ten one-vs-all classifiers using a variant of the LPUBoost algorithm - Jure Leskovec and John Shawe-Taylor: 'Linear Programming boost for Uneven Datasets'. The weak learner of the boosting algorithm was the same as in method 1.

2.16 QMUL

Participants: Jianguo Zhang[1], Cordelia Schmid[2],
Svetlana Lazebnik[3], Jean Ponce[4,3]
Affiliation: [1]Queen Mary, University of London;
[2]INRIA Rhones-Alpes; [3]University of Illinois;
[4]Ecole Normale Superieure, Paris
E-mail: jgzhang@dcs.qmul.ac.uk

Bag of Features and Spatial Pyramid. We start with our kernel-based bag of visual features approach [6]. In this approach, object images are characterized by orderless histograms of appearance-based descriptors (such as SIFT [5]) of either sparse features computed at a set of keypoint locations, or dense features computed at a set of points on a fixed image grid. We use $\chi 2$ distance to compare the obtained histograms of bag of features. The approach of spatial pyramid matching [4], is proposed to augment the basic bag-of-features representation by adapting the pyramid matching scheme of Grauman and Darrell [3]. It is worth noting that the spatial pyramid method is global, i.e., it represents spatial information of features in a global coordinate system as opposed to an object-centered coordinate system [2], and is thus not translationor scale-invariant. This makes spatial pyramids more appropriate for scene as opposed to category recognition. However, as the results in [4] show, it still works well for objects even in the presence of clutter and geometric transformations. See [4] for details. To compare the spatial histograms, one strategy is the pyramid matching kernel used as in [3, 4]. Another strategy is presented in the following section.

A Two-layer Spatial Pyramid SVM Classifier. We propose to modify the approach of [4] in two ways. One major issue is that in [4], matching weights between different levels of the pyramid are fixed based on the formulation of a maximum-weight matching problem instead of being adaptively learned to yield optimal classification performance. So here we argue that it may be useful to learn the weights using a classifier, e.g. SVM, instead of having them fixed. A second concern is that the image representation tested in [4] is dense, i.e., SIFT features on fixed dense grid, and good recognition performance is achieved on datasets that lack heavy background clutter and large scale changes, e.g., CalTech101[1]. Performance on more challenging datasets with large viewpoint changes or diverse poses is not fully demonstrated. To address this limitation, we investigate how well the spatial pyramid works with sparse local features. Intrinsically, the construction of the spatial pyramid matching can be explained as a set of bag of features representations at each pyramid level combined with different weights. Thus, it is straightforward to classify each pyramid level separately, and then use a classifier to combine the outputs. Based on this idea, we construct a two-layer SVM classifier. More specifically, we first learn a set of SVM classifiers with the $\chi 2$ kernel based on the histograms of each level in the pyramid. Next, the outputs of these SVM classifiers are concatenated into a feature vector for each image and used to learn another SVM classifier based on a Gaussian RBF kernel. Note that the bag of features representation can be considered as the global histogram at the ground level in the spatial pyramid. The spatial pyramid can be built either using sparse or dense features, thus resulting in different methods. Note that the two-layer SVM classifier can also be used to combine several bag-of-features image representations based on different types of keypoints, e.g. Harris-Laplacian and Laplacian.

Discussion. In the challenge, we randomly select 50000 local descriptors from the training images of each class. We then cluster these features with k-means (k = 300) and concatenate the cluster centers of the 10 classes to build a global vocabulary of 3000 words. Based on our experiments on the validation set, we have observed that both basic bag-of-features and spatial pyramid image representations achieve excellent results for object category classification. However, using the spatial pyramid up to level 2 does not give much improvement upon basic bag of features when the visual vocabulary is sufficiently rich. This may due to the diversity of the pose and viewpoint of the objects presented in this dataset. We can also see that the pyramid matching kernel might not be necessary for comparing spatial pyramids. Using the proposed two-layer pyramid SVM classifier can achieve similar or even better results. This might be because the weights of each level in the pyramid are automatically learned by SVM in the two-layer pyramid scheme, while in pyramid matching, they are fixed a priori. Based on these observations, we have presented two major methods for the PASCAL06 challenge, (1) Spatial pyramid with a two-layer SVM classifier on Laplacian points, denoted by LSPCH; (2) Bag of key points with Laplacian and Harris-Laplacian combined by the two layer SVM strategy, denoted by HSLS.

References

[1] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on

101 object categories. In IEEE CVPR Workshop on Generative-Model Based Vision, 2004.

[2] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from googles image search. In ICCV, October 2005.

[3] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. In CVPR, volume 2, pages 627634, 2005.

[4] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In CVPR, 2006. [5] D. Lowe. Distinctive image features form scale-invariant keypoints. IJCV, 60(2):91110, 2004.

[5] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. IJCV, 2006. To appear.

2.17 RWTH_DiscHist

Participants: Thomas Deselaers
Affiliation: RWTH Aachen
E-mail: `deselaers@informatik.rwth-aachen.de`

The method for discriminative training of image patch histograms which has been proposed in [1] consists of two steps: 1. feature extraction and 2. training and classification. These steps are laid out in the following. Additionally, we describe some extensions we used for our submission.

Feature Extraction. Given an image, we extract image patches around up to 500 points of interest and 300 points from a regular grid. In contrast to the interest points, the grid points can also fall onto very homogeneous areas of the image. This property is important for capturing homogeneity in objects in addition to points that are detected by interest point detectors, which are usually of high variance. To the extracted image patches, a PCA dimensionality reduction is applied, keeping 40 coefficients. These data are then clustered using a Linde-Buzo-Gray algorithm. Then, we discard all information for each patch except its corresponding closest cluster center identifier. For the test data, this identifier is determined by evaluating the Mahalanobis distance to all cluster centers for each patch. From these cluster center identifiers we create a histogram representation for each image.

Classification. Having obtained this representation by histograms of image patches, we need to define a decision rule for the classification of images. It was shown that a method using discriminative training of log-linear models outperforms other methods. Discriminative training means to use the information of competing classes during training. This is done by maximizing the posterior probability instead of maximizing the class-conditional probability as is done in maximum likelihood approaches.

In initial experiments to tune all system parameters (e.g. number of histogram bins, feature extraction points, feature vectors) using the train and the validation data, we found out that color is useful for the classes cat, cow, dog, horse, and sheep. For the remaining tasks, the images were converted to gray values.

References

1 T. Deselaers, D. Keysers, and H. Ney. Improving a Discriminative Approach to Object Recognition using Image Patches. In DAGM 2005, Pattern Recognition, 26th DAGM Symposium, Lecture Notes in Computer Science, pages 326–333, Vienna, Austria, August 2005.

2.18 RWTH_GMM

Participants: Thomas Deselaers
Affiliation: RWTH Aachen
E-mail: `deselaers@informatik.rwth-aachen.de`

The method uses Gaussian Mixture Models to recognize images represented by patches. The method is described in [1,2].

Feature Extraction and Training. Given an image, we extract image patches around interest points of various types. We use wavelet-based salient points, difference-of-Gaussian interest points, and points taken from a regular grid. For those points where no size is automatically extracted, patches are extracted of various sizes.

Then, the patches are PCA transformed keeping 40 dimensions to reduce the amount of data to be handled and a class-dependent Gaussian mixture model is estimated for each class. These models are estimated using the EM algorithm for Gaussian mixture models.

In a refinement step, the cluster weights can be trained discriminatively which has been proven to lead to much better results. Unfortunately, for the PASCAL VOC06 we were unable to use this refinement due to memory and computing time limitations.

Classification. Given an image to be classified, the patches are extracted in the same way as for the training images and Bayes' decision rule is used to classify the image.

References

[1] Andre Hegerath. Patch-based Object Recognition. Diploma Thesis. RWTH Aachen University, Aachen, Germany. 2006.

[2] Andre Hegerath, Thomas Deselaers, Hermann Ney. Patch-based Object Recognition Using Discriminatively Trained Gaussian Mixtures. In 17th British Machine Vision Conference (BMVC06), in press, Edinburgh, UK, September 2006.

2.19 RWTH_SparseHists

Participants: Thomas Deselaers
Affiliation: RWTH Aachen
E-mail: `deselaers@informatik.rwth-aachen.de`

The method uses sparse histograms of image patches that are extracted from the images. It was proposed in [1,2]. The method consists of two steps: 1. creation of sparse histograms of image patches. 2. classification.

Create of sparse histograms. In contrast to other methods this method can cope with arbitrary large numbers of patches extracted from the images. Thus we extract patches of four different sizes at each position in the images. These patches are PCA reduced and inserted into the histogram.

Classification. The histograms are classified using a maximum entropy trained log-linear model using Bayes' decision rule.

References

[1] Andre Hegerath. Patch-based Object Recognition. Diploma Thesis. RWTH Aachen University, Aachen, Germany. 2006.

[2] Thomas Deselaers, Andre Hegerath, Daniel Keysers, Hermann Ney. Sparse Patch-Histograms for Object Classification in Cluttered Images. In DAGM 2006, Pattern Recognition, 26th DAGM Symposium, Lecture Notes in Computer Science, volume 4174, pages 202–211, Berlin, Germany, September 2006.

2.20 Siena

Participants: Gariele Manfardini, Vincenzo Di Massa
Affiliation: Universita degli Studi di Siena
E-mail: `gabrimonfa@gmail.com`

We have submitted results for the classification competition, using VOC data for training, in all the 10 classes.

Each image was preprocessed in order to obtain a Region Adjacency Graph (RAG) that represents it. A RAG is a graph where nodes denote homogeneous regions of the image and edges stand for the adjacency relationships. Nodes and edges are labeled. Node labels contain geometric features of the corresponding regions (area, perimeter, orientation of the principal axes, and so on), while edge labels encode the mutual orientation of the two regions and the difference between their average colors. RAGs were obtained by the following steps:

1. The image was filtered using the Mean Shift algorithm.
2. A k-means color quantization procedure was performed to locate an initial big number of homogeneous regions.

3. Some adjacent regions were merged to achieve the desired number of final regions. The procedure evaluated a dissimilarity function between all the pairs of regions and merged those that achieve the lowest values, updating the list at each fusion. The dissimilarity function has been chosen heuristically considering the distance between the average colors of the two regions and their dimension.

Then the RAGs was processed by using a GNN (Graph Neural Network) model. Graph Neural Networks have been recently proposed to process very general types of graphs and can be considered an extension of RNNs (Recursive Neural Networks). Actually, the main difference is that RNNs require input graphs to be directed and acyclic, while cyclic or non-directed structures must undergo a preprocessing phase. On the contrary GNNs can process directly very general kind of graphs, cyclic, acyclic, directed, undirected, with labeled nodes and edges, without any preprocessing. Given a graph G and one of its node n, GNNs allow to approximate functions

$$\phi : set(G) \times N \to R^m$$

where set(G) is the set of graphs G, N is the set of the nodes of the chosen graph and R is the set of real numbers. In other words GNNs can evaluate an output for one or more nodes of each graph G, i.e. is able to perform graph classification/regression and node classification/regression. Some interesting results on approximation capabilities of this neural network model had been proposed in literature.

References

[1] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in Proc. International Joint Conference on Neural Networks (IJCNN2005), 2005, pp. 729–734.

[2] F. Scarselli, S. Yong, M. Gori, M. Hagenbuchner, A. Tsoi, and M. Maggini, “Graph neural networks for ranking web pages,” in Proc. of the 2005 IEEE/WIC/ACM Conference on Web Intelligence (WI2005), 2005, pp. 666–672.

[3] V. Di Massa, G. Monfardini, L. Sarti, F. Scarselli, M. Maggini, M. Gori, “A Comparison between Recursive Neural Networks and Graph Neural Networks” World Conference on Computational Intelligence 2006 (WCCI’06) (accepted)

[4] G. Monfardini, V. Di Massa, F. Scarselli, M. Gori, “Graph Neural Networks for Object Localization”, European Conference on Artificial Intelligence (ECAI 2006) (accepted)

2.21 TKK

Participants: Ville Viitaniemi
Affiliation: Helsinki University of Technology
E-mail: `Ville.Viitaniemi@tkk.fi`

In the PicSOM image analysis framework [1] a neural representation is symmetrically formed for both training and test set images. To this end, the images are automatically segmented and a large number of statistical descriptors is calculated for the segments as well as for the whole images. The set of descriptors includes a subset of MPEG-7 visual descriptors, several non-standard descriptors, and their combinations.

The descriptors are partitioned into feature spaces. Each of the feature spaces is quantised with a Tree-Structured Self-Organising Map (TS-SOM). Because the quantisations preserve the topology of original feature spaces, we can classify the reduced representations using standard vector space classification methods. Within a feature space, a confidence value for a whole image is obtained by summing the contributions of all its segments. Confidence values from the different feature spaces are summed together to obtain the final confidence value for each image.

The detection task is solved by classifying the automatically obtained image segments within images using the PicSOM framework. This gives an approximation of the conditional probability that tells which of the image segments correspond to the target class if the image as a whole does. This probability is further modulated with results from the image classification task. For additional details, see reference [2].

References

[1] Jorma Laaksonen, Markus Koskela, and Erkki Oja. PicSOM – Self-Organizing Image Retrieval With MPEG-7 Content Descriptors. IEEE Transactions on Neural Networks, 13(4): 841–853, July 2002

[2] Ville Viitaniemi and Jorma Laaksonen. Techniques for Still Image Scene Classification and Object Detection. In Proceedings of 16th International Conference on Artificial Neural Networks (ICANN 2006), 2006 (to appear)

2.22 TUD

Participants: Nikodem Majer, Mario Fritz, Edgar Seemann,
Gyuri Dorkó, Bastian Leibe, Bernt Schiele
Affiliation: TU Darmstadt
E-mail: `fritz@mis.tu-darmstadt.de`

For object localization (competition #3), we have submitted results on the categories motorbikes and people. Our method is based on the *Implicit Shape Model (ISM)* [4]. Since the method is similar to the one presented on the PASCAL Visual Object Challenge 2005—that time including a discriminative extension [3]—our submission this year can be seen as a baseline experiment with the ISM. However, we have made some modification to adapt the model to the new challenge. First, the requirement of pixel-level segmentation is relaxed to bounding box annotation, and second discriminative codebook selection is used to improve runtime performance. In the followings, we briefly outline the ISM and describe our parameters used in the experiments.

Method description. The ISM approach first builds a codebook of local appearance-based descriptors using an agglomerative clustering scheme. The object category is modeled by a set of non-parametric spatial distribution of feature occurrences learned for each codebook entry, relatively to the object center. For detection, extracted local features are first matched to the codebook, then based on the matched entries, object hypotheses are computed using a voting scheme. At the second stage, the original ISM [4] uses pixel-wise segmentation to increase the performance. Since the challenge datasets do not contain this information, we have substituted the segmentation masks with rectangular annotations based on the provided bounding boxes. We note, that this is the first time, when ISM is used without the pixel-based segmentations. Discriminative feature selection based on the likelihood ratio [2] is used to significantly reduce the size of the codebook, which leads to substantial speed-up for the detection. For a more detailed description of the original ISM—in particular how to achieve scale-invariance—we refer to [5]. We would like to note that in contrast to [7] our results are corresponding to the performance of the original ISM which was not designed to learn objects from a different viewpoints.

Preprocessing. The provided bounding box information was used to rescale the training object to have a constant height of 160 pixel for the motorbikes and 200 pixels for the people. Additionally we selected 220 most representative training examples for the category *people*.

Parameters. For appearance representation we used the scale-invariant Hessian-Laplace [6] detector, with shape context descriptors [1, 6]. Due to the feature selection the size of the codebook is reduced from 3968 to 80, and 5963 to 204, for motorbikes and people respectively.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[2] G. Dorko and C. Schmid. Selection of scale invariant parts for object class recognition. In *Ninth International Conference on Computer Vision (ICCV'03)*, 2003.

[3] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminant models for object category detection. In *International Conference on Computer Vision (ICCV'05)*, Beijing, China, October 2005.

[4] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech Republic, May 2004.

[5] B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *DAGM'04 Annual Pattern Recognition Symposium*, Tuebingen, Germany, Aug. 2004.

[6] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[7] E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *CVPR'06*, 2006. To appear.

2.23 UVA_big5

Participants: Jan van Gemert, Arjan Gijsenij
Affiliation: University of Amsterdam
E-mail: `jvgemert@science.uva.nl`

This method combines several detectors and descriptors. The output of each descriptor on the train+val set is clustered with a radius-based clustering algorithm. These clusters are subsequently used to characterize an image in the whole set.

The detectors consist of

- An overlapping 2D grid
- Maximally Stable Extremal Regions (mser)
- Harris Laplacian
- Hessian Affine

The descriptors consist of:

- Color Invariant Weibull Features as developed by the UvA.
- Sift
- Spin
- Gloh
- Shape Context

Based on cross validation performance on the train+val set, we chose the best representative for each descriptor. The best results were given by these five: mser.spin + grid.weibull + mser.shapeCtx + harlap.sift + hesaff.gloh

The image characterizations based on the clusters were used to train a non-linear classifier on the test+val set, which was used to predict scores on the test set.

2.24 UVA_weibull

Participants: Jan van Gemert, Arjan Gijsenij
Affiliation: University of Amsterdam
E-mail: `jvgemert@science.uva.nl`

This method utilizes a novel descriptor developed at the University of Amsterdam. A 2D overlapping grid is placed over an image. For each region in this grid, a Weibull distribution is fitted over the edge responses of the region. The beta and gamma parameters of the Weibull distribution are used as a descriptor. The Weibulls are computed on each color channel, and makes use of color invariance to cope with different lighting conditions.

The similarity of the regions in an image are aggregated with typical Weibull parameters of 15 proto-concepts like vegetation, water, fire, sky etc. These similarities are subsequently used to characterize an image.

The image characterizations were used to train a non-linear classifier on the test+val set, which was used to predict scores on the test set.

2.25 XRCE

Participants: Florent Perronnin
Affiliation: Xerox Research Centre Europe
E-mail: `Florent.Perronnin@xrce.xerox.com`

The Xerox generic visual categorizer (GVC) is based on a novel and practical approach described in the following paper: ”Adapted vocabularies for generic visual categorization” by Florent Perronnin, Christopher Dance, Gabriela Csurka and Marco Bressan, to appear in the proceedings of ECCV 2006. This approach extends the traditional bag of visual words. It is based on a universal vocabulary, which describes the content of all the considered classes of images, and class vocabularies obtained through the adaptation of the universal vocabulary using class-specific data. An image is characterized by a set of histograms, one per class, where each histogram describes whether the image content is best modeled by the universal vocabulary or the corresponding class vocabulary. Our local low-level features contain gradient orientation and color information. They are extracted on regular grids at multiple scales. For the classification of the bi-partite histograms, one linear classifier per class is trained in a one-vs-all manner using sparse logistic regression. Running our unoptimized code on a modern PC, the categorization of one image, which includes low level feature extraction, histogram computation and histogram classification, takes less than 700 ms. Note that we used the default parameters of our categorizer and did not tune the system specifically for the challenge.

2.26 ROUND2_INRIA_Moosmann

Participants: Frank Moosmann
Affiliation: INRIA Rhone-Alpes
E-mail: `frank.moosmann@inrialpes.fr`

The method used follows the method described in [1]. First random subwindows are extracted from the training images (random size, random position). Each window is then described by the SIFT-Descriptor, concatenated with color information in HSL color space. Extremely randomized trees are then used to cluster the labeled training features in a supervised way. The leaf nodes form the visual vocabulary. A linear SVM is afterwards trained with binarized histograms, created from the training images. The extraction of features is done as before (but on the whole image) and the PDF responsible for the selection of the random windows is adjusted after each selection and propagation through the trees depending on the output of the trees. This leads to more features to be extracted in regions where the object is estimated. To build the trees 50000 features were extracted in total. To create histograms 10000 features per image were used.

References

[1] F.Moosmann, D.Larlus and F.Jurie, Learning Saliency Maps for Object Categorization, ECCV International Workshop on The Representation and Use of Prior Knowledge in Vision, 2006

3 Results: Classification

The following sections present the results for the classification competitions. For each competition, the area under ROC curve (AUC) is reported by participant and class. The 'best' result, as measured by AUC, is underlined; where several methods obtained the same AUC to three decimal places, all are underlined.

In the figures of ROC curves, the curves have been sorted by decreasing AUC to aid visibility of the most successful entries. Two sets of ROC curves are shown: (i) all submitted results; (ii) the top 5 results submitted, as measured by AUC.

3.1 Competition 1

- Train on `trainval` data provided, classify object present/absent.

There were 20 submissions for this competition. All but two tackled all ten object classes. In the second round of the challenge, three participants submitted additional results, listed at the bottom of table 2. For the second round, ground truth annotation was not available for the test data, but participants did have additional time, so the results should not be directly compared to the others.

	bicycle	bus	car	cat	cow	dog	horse	motorbike	person	sheep
AP06_Batra	0.791	0.637	0.833	0.733	0.756	0.644	0.607	0.672	0.550	0.792
AP06_Lee	0.845	0.916	0.897	0.859	0.838	0.766	0.694	0.829	0.622	0.875
Cambridge	0.873	0.864	0.887	0.822	0.850	0.768	0.754	0.844	0.715	0.866
INRIA_Larlus	0.903	0.948	0.943	0.870	0.880	0.743	0.850	0.890	0.736	0.892
INRIA_Marszalek	0.929	0.984	0.971	0.922	0.938	0.856	0.908	0.964	0.845	0.944
INRIA_Moosmann	0.903	0.933	0.957	0.883	0.895	0.825	0.824	–	0.780	0.930
INRIA_Nowak	0.924	0.973	0.971	0.906	0.892	0.797	0.904	0.961	0.814	0.940
INSARouen	–	–	0.895	–	–	0.764	–	–	–	0.869
MUL_1vALL	0.857	0.852	0.914	0.562	0.632	0.584	0.525	0.831	0.616	0.758
MUL_1v1	0.864	0.945	0.928	0.826	0.789	0.764	0.733	0.906	0.718	0.872
QMUL_HSLS	0.944	0.984	0.977	0.936	0.936	0.874	0.922	0.966	0.845	0.946
QMUL_LSPCH	0.948	0.981	0.975	0.937	0.938	0.876	0.926	0.969	0.855	0.956
RWTH_DiscHist	0.874	0.955	0.930	0.879	0.910	0.799	0.854	0.938	0.764	0.906
RWTH_GMM	0.882	0.935	0.942	0.866	0.856	0.825	0.802	0.905	0.718	0.892
RWTH_SparseHists	0.863	0.941	0.935	0.883	0.883	0.704	0.844	0.858	0.776	0.907
Siena	0.671	0.749	0.842	0.696	0.774	0.677	0.644	0.701	0.660	0.768
TKK	0.857	0.928	0.943	0.871	0.892	0.811	0.806	0.908	0.781	0.900
UVA_big5	0.897	0.929	0.945	0.845	0.862	0.785	0.806	0.923	0.774	0.885
UVA_weibull	0.855	0.880	0.910	0.818	0.849	0.762	0.759	0.888	0.723	0.811
XRCE	0.943	0.978	0.967	0.933	0.940	0.866	0.925	0.957	0.863	0.951
ROUND2_INRIA_Moosmann	0.924	0.962	0.972	0.893	0.925	0.851	0.877	0.934	0.840	0.942
ROUND2_Siena	–	–	–	–	0.699	–	–	–	–	–
ROUND2_TKK	0.843	0.940	0.951	0.876	0.898	0.835	0.856	0.908	0.778	0.899

Table 2: Competition 1

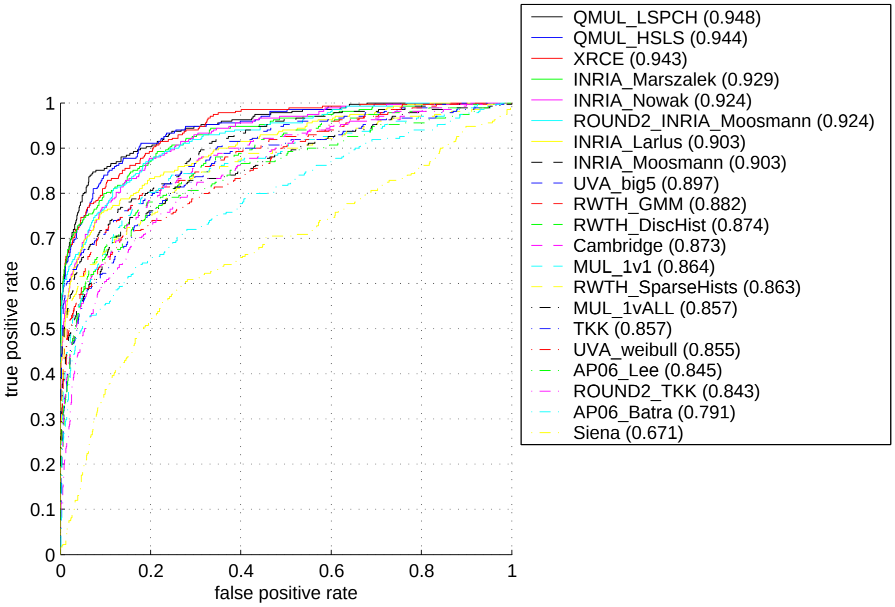

Figure 1: Competition 1.1: bicycle (all entries)

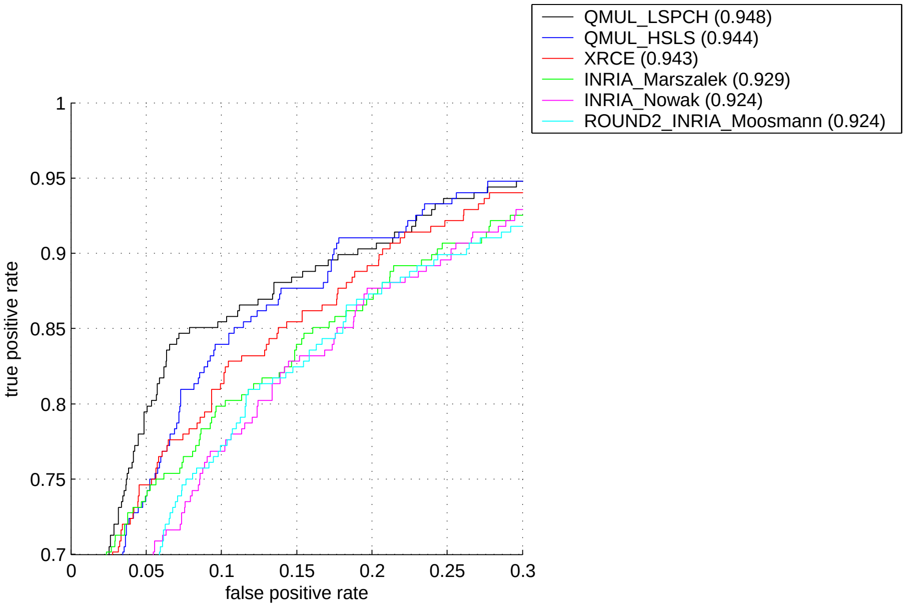

Figure 2: Competition 1.1: bicycle (top 5 by AUC)

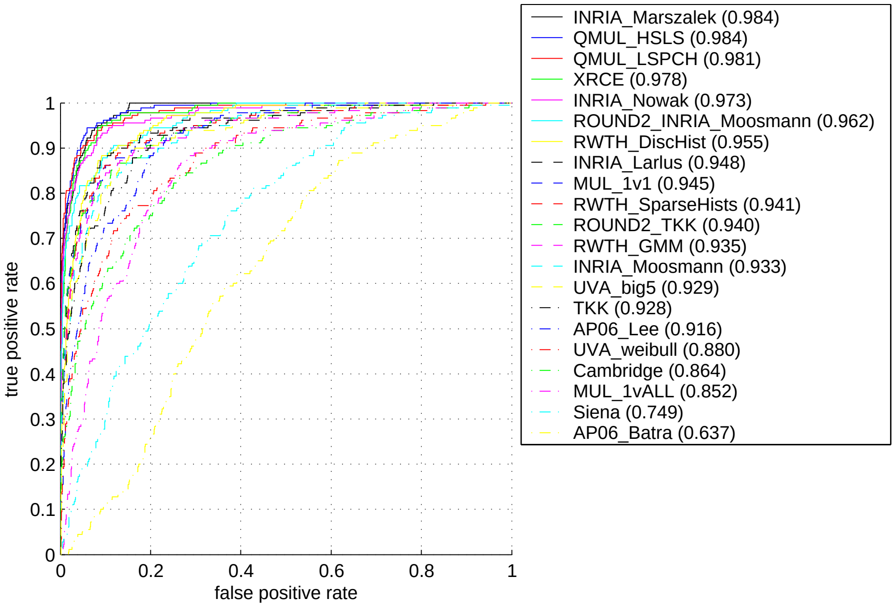

Figure 3: Competition 1.2: bus (all entries)

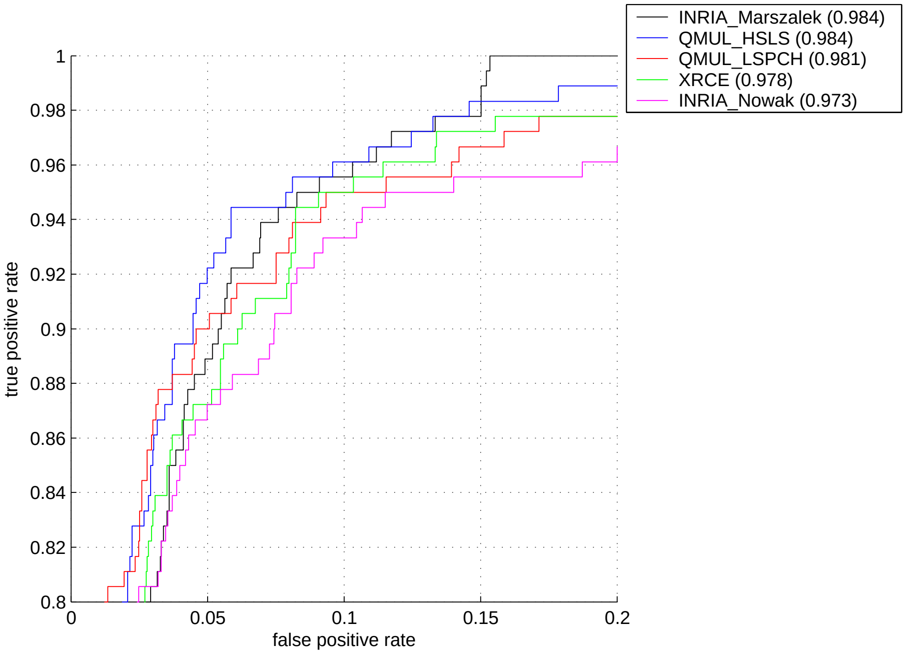

Figure 4: Competition 1.2: bus (top 5 by AUC)

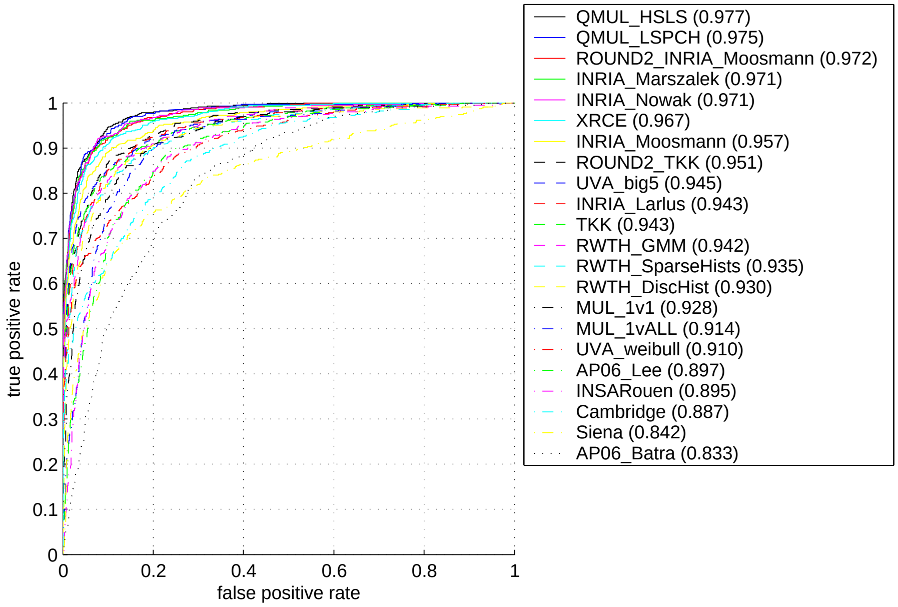

Figure 5: Competition 1.3: car (all entries)

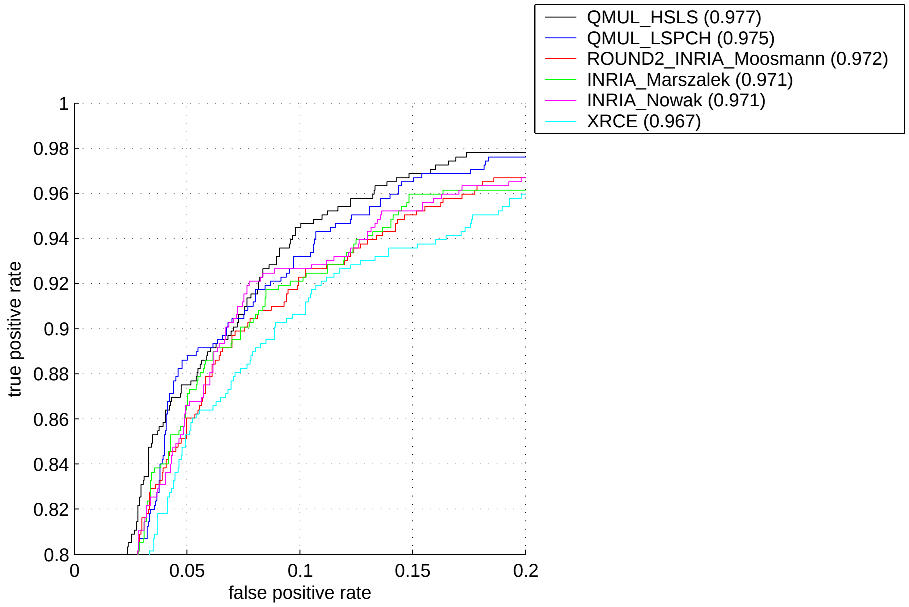

Figure 6: Competition 1.3: car (top 5 by AUC)

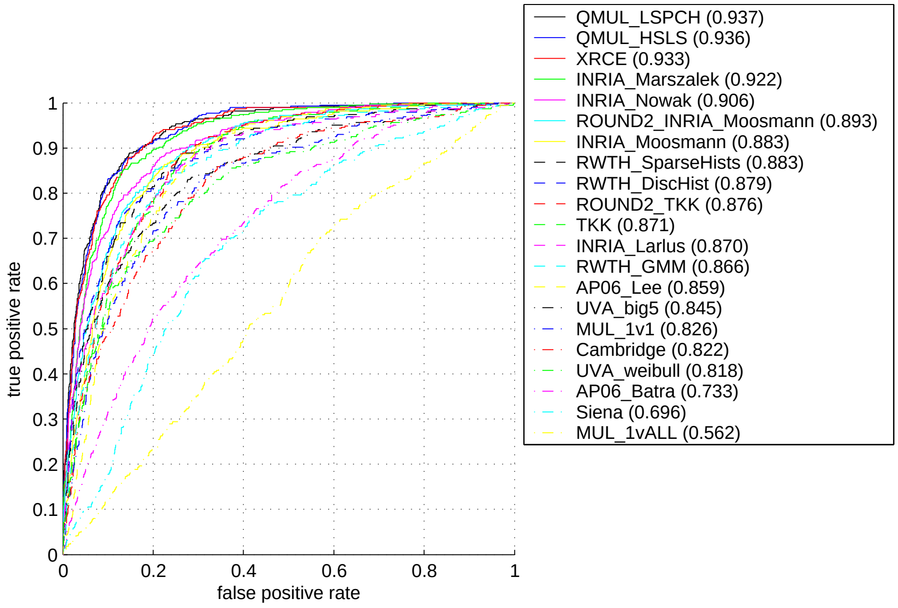

Figure 7: Competition 1.4: cat (all entries)

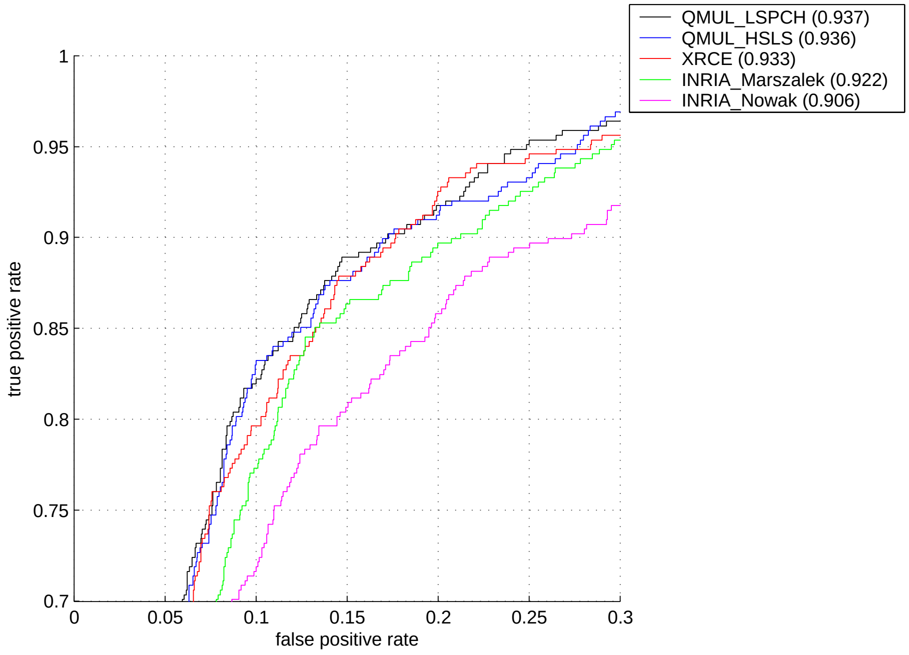

Figure 8: Competition 1.4: cat (top 5 by AUC)

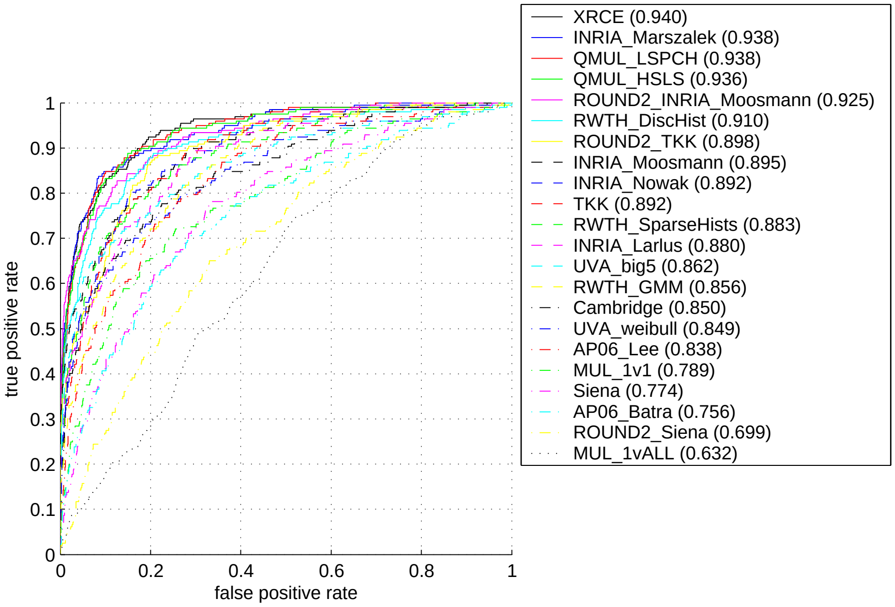

Figure 9: Competition 1.5: cow (all entries)

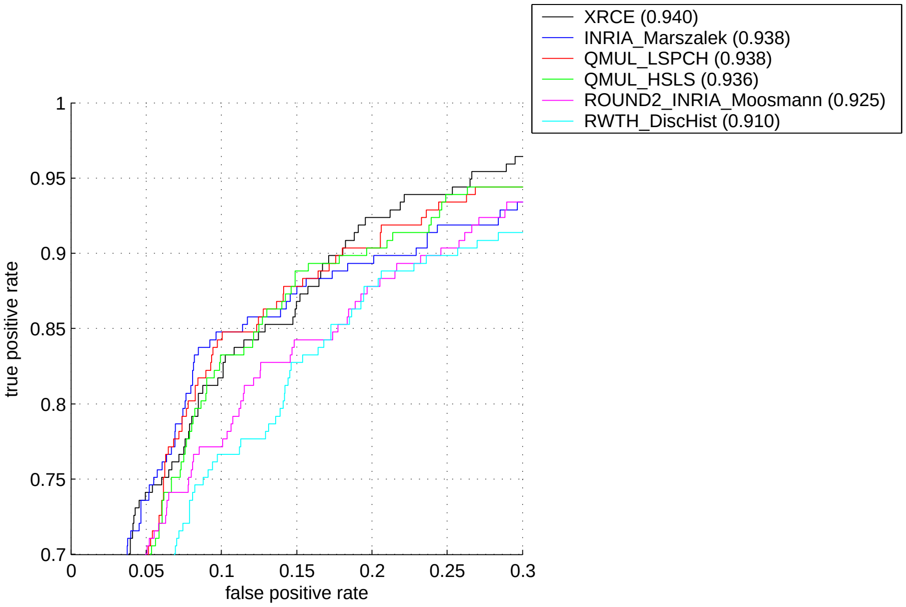

Figure 10: Competition 1.5: cow (top 5 by AUC)

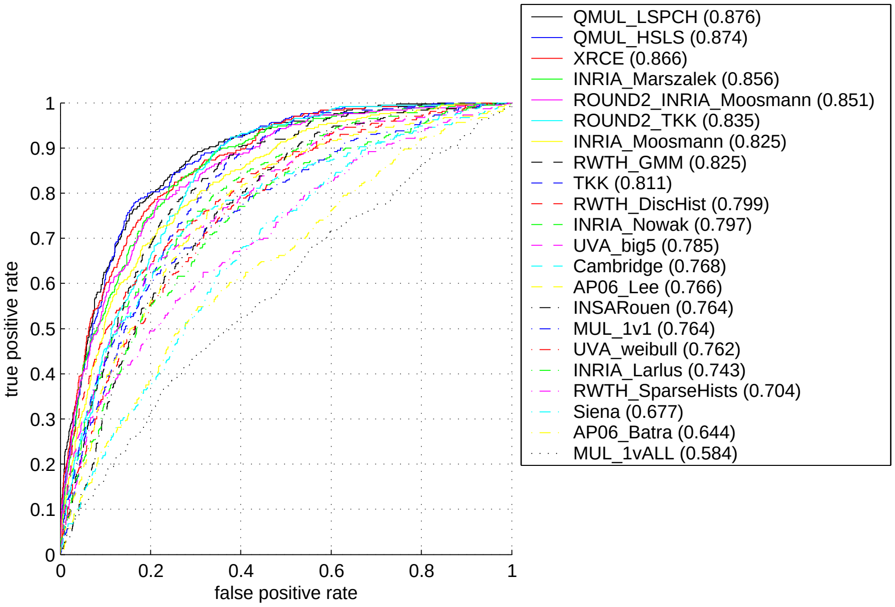

Figure 11: Competition 1.6: dog (all entries)

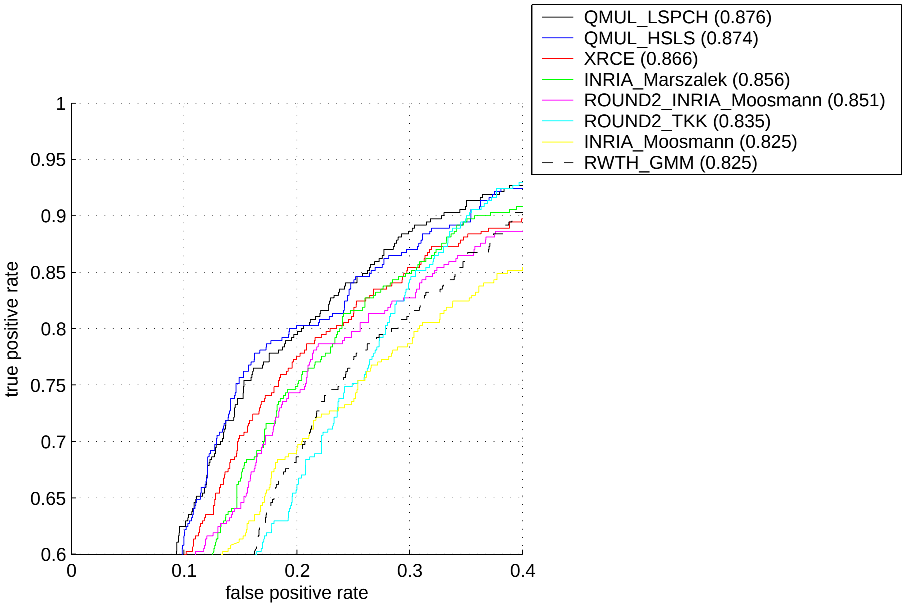

Figure 12: Competition 1.6: dog (top 5 by AUC)

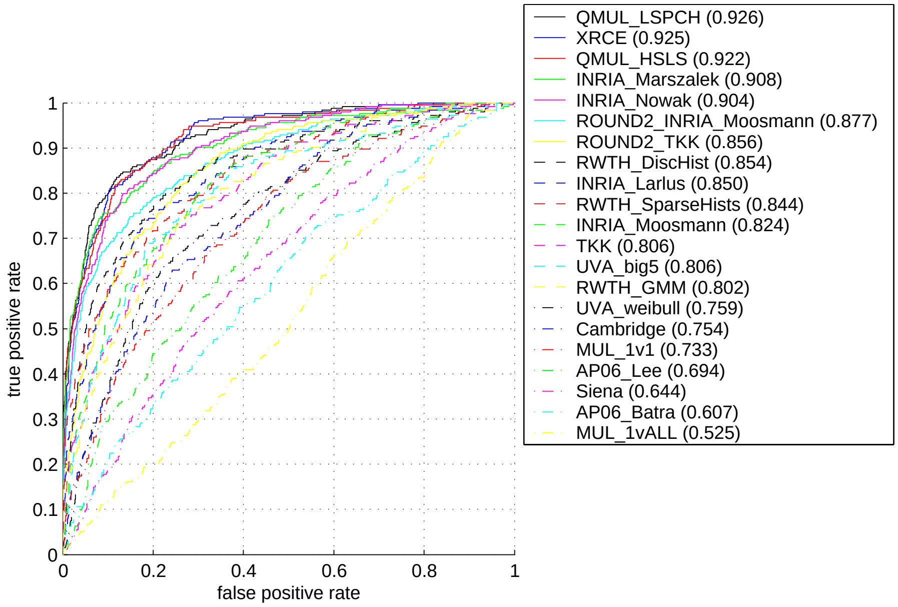

Figure 13: Competition 1.7: horse (all entries)

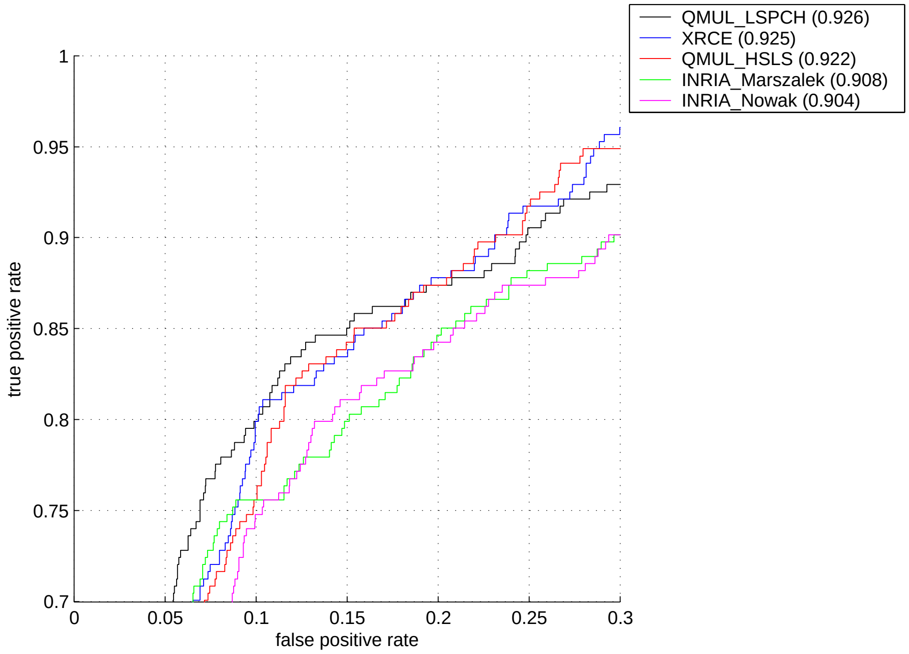

Figure 14: Competition 1.7: horse (top 5 by AUC)

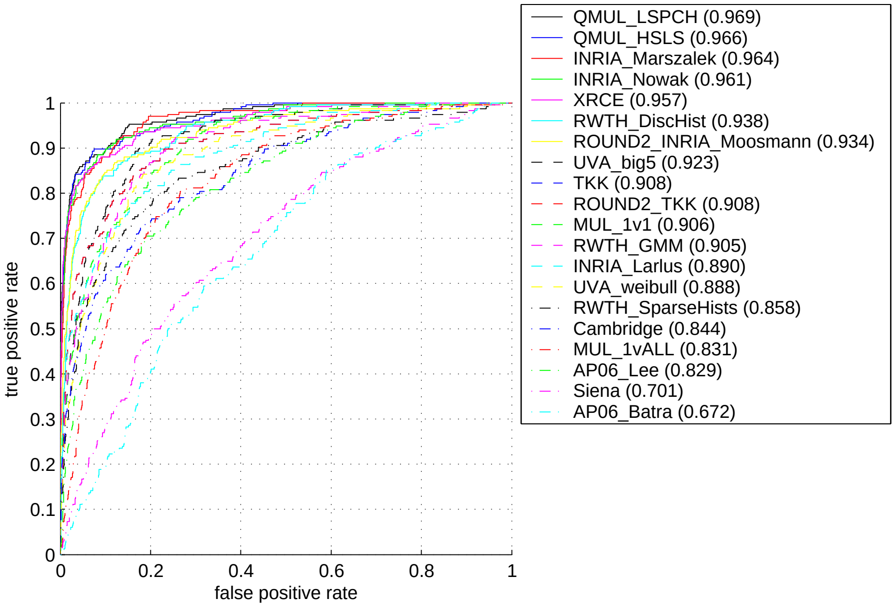

Figure 15: Competition 1.8: motorbike (all entries)

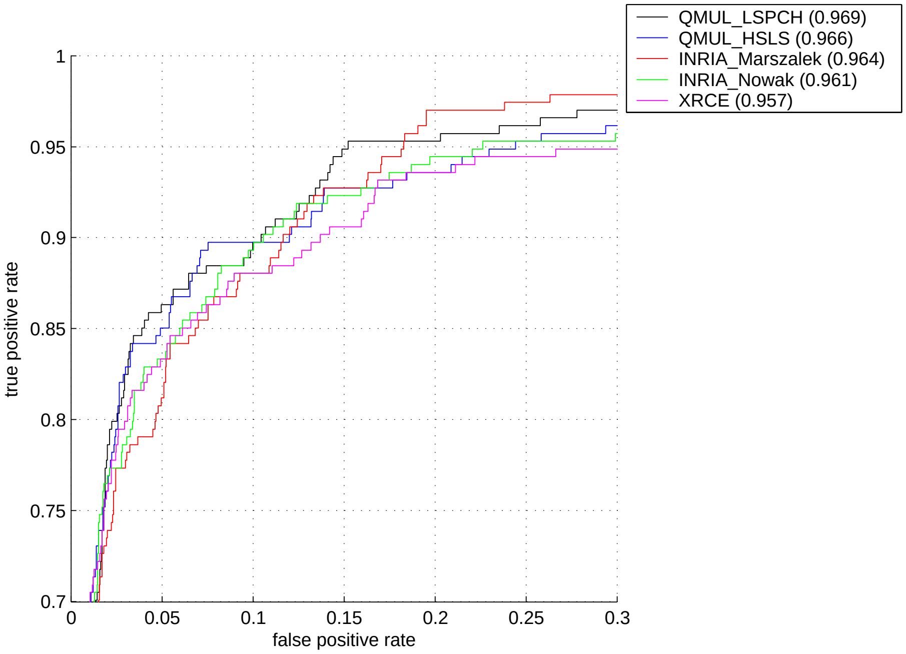

Figure 16: Competition 1.8: motorbike (top 5 by AUC)

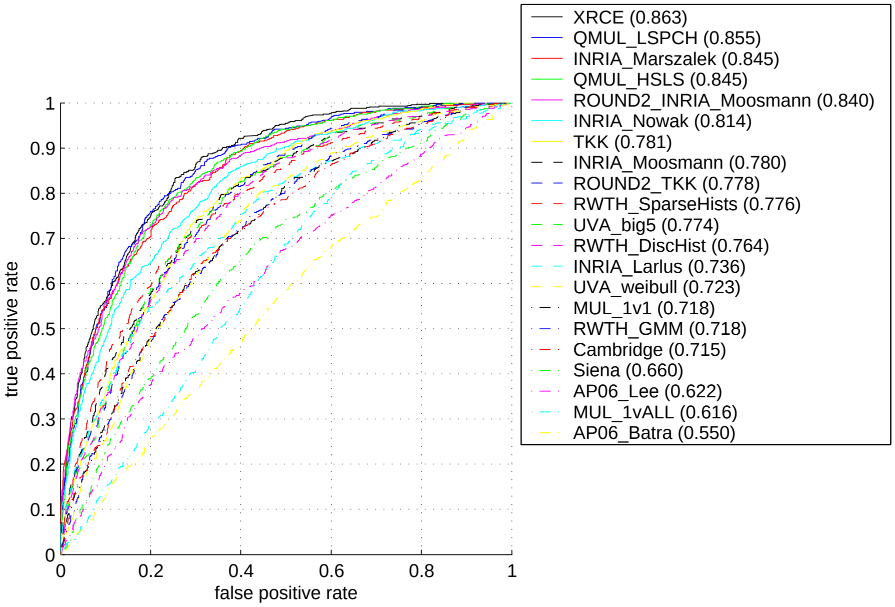

Figure 17: Competition 1.9: person (all entries)

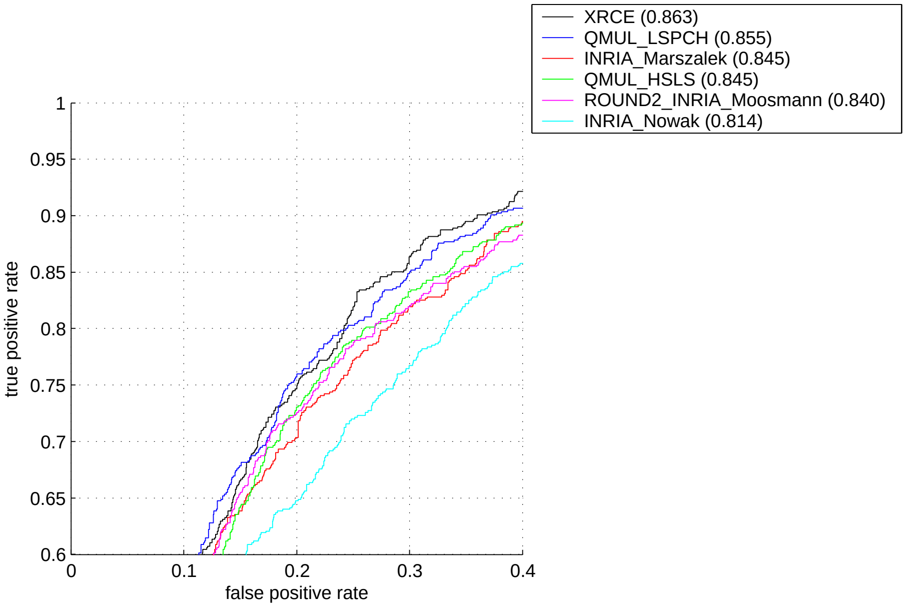

Figure 18: Competition 1.9: person (top 5 by AUC)

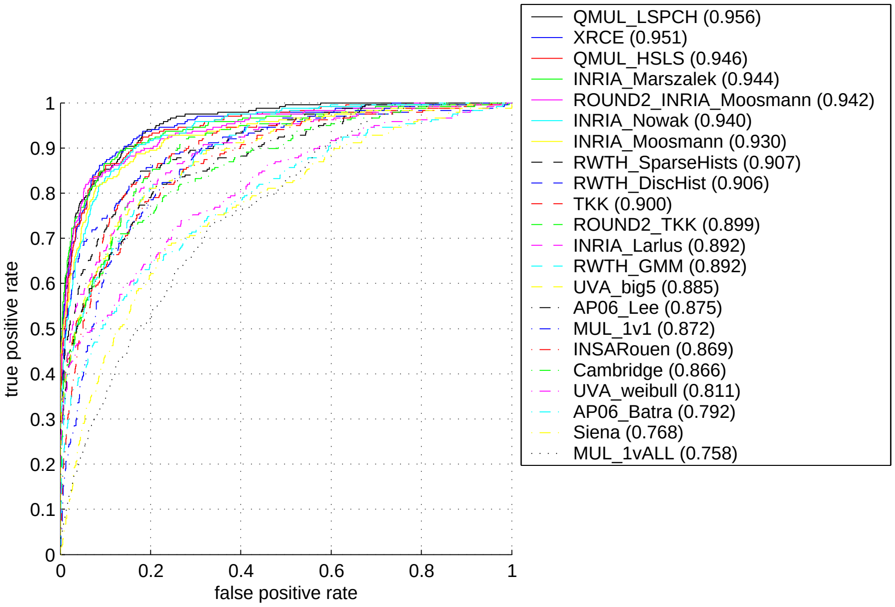

Figure 19: Competition 1.10: sheep (all entries)

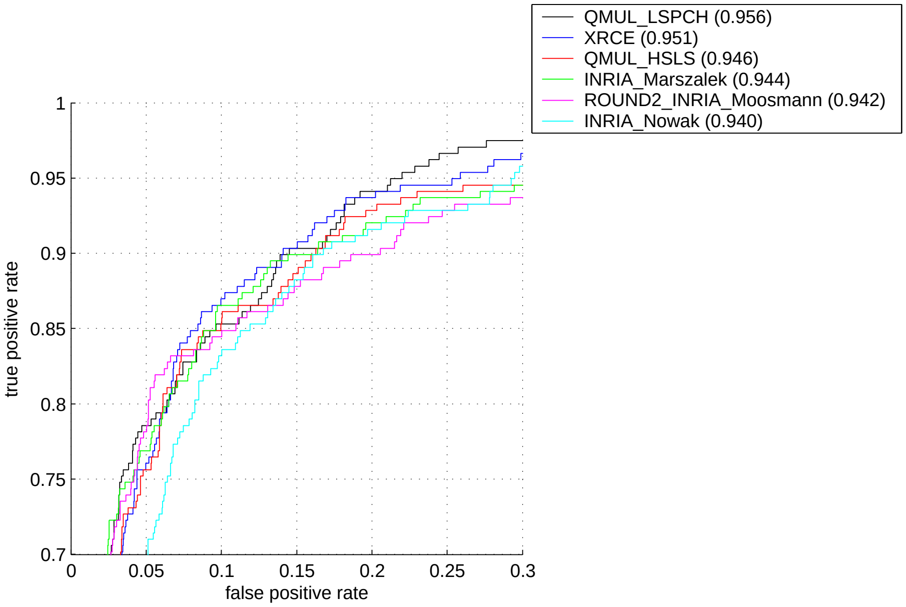

Figure 20: Competition 1.10: sheep (top 5 by AUC)

3.2 Competition 2

- Train on any (non-test) data, classify object present/absent.

Three submissions were received for this competition, in which participants were to submit results of classifiers trained on their own data. KUL submitted results for ‘motorbike’ alone, MIT_Torralba for ‘car’, and MIT_Fergus for ‘car‘ and ‘motorbike’.

	bicycle	**bus**	**car**	**cat**	**cow**	**dog**	**horse**	**motorbike**	**person**	**sheep**
KUL	–	–	–	–	–	–	–	0.797	–	–
MIT_Fergus	–	–	0.763	–	–	–	–	0.821	–	–
MIT_Torralba	–	–	0.745	–	–	–	–	–	–	–

Table 3: Competition 2

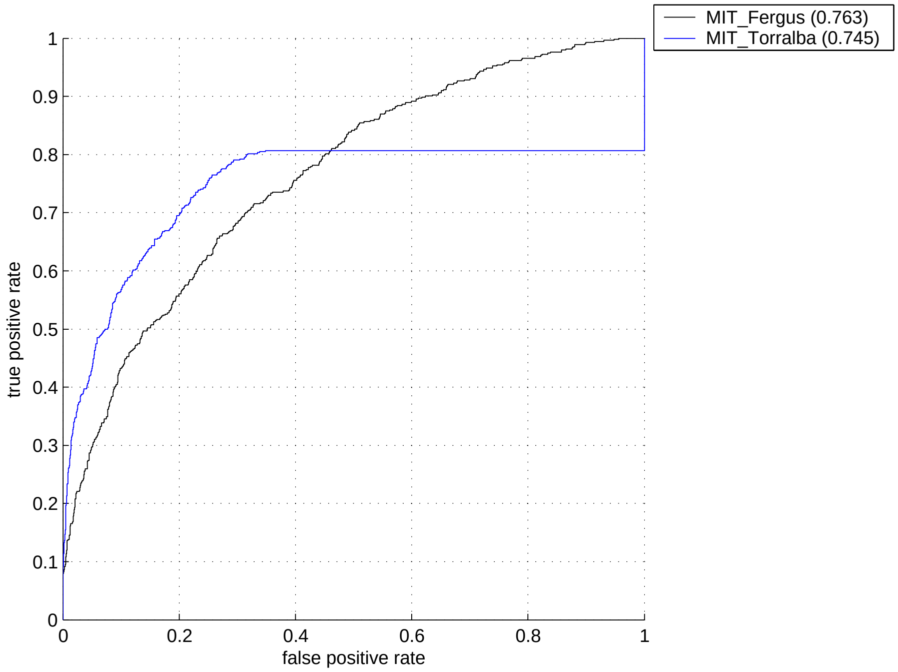

Figure 21: Competition 2.3: car (all entries)

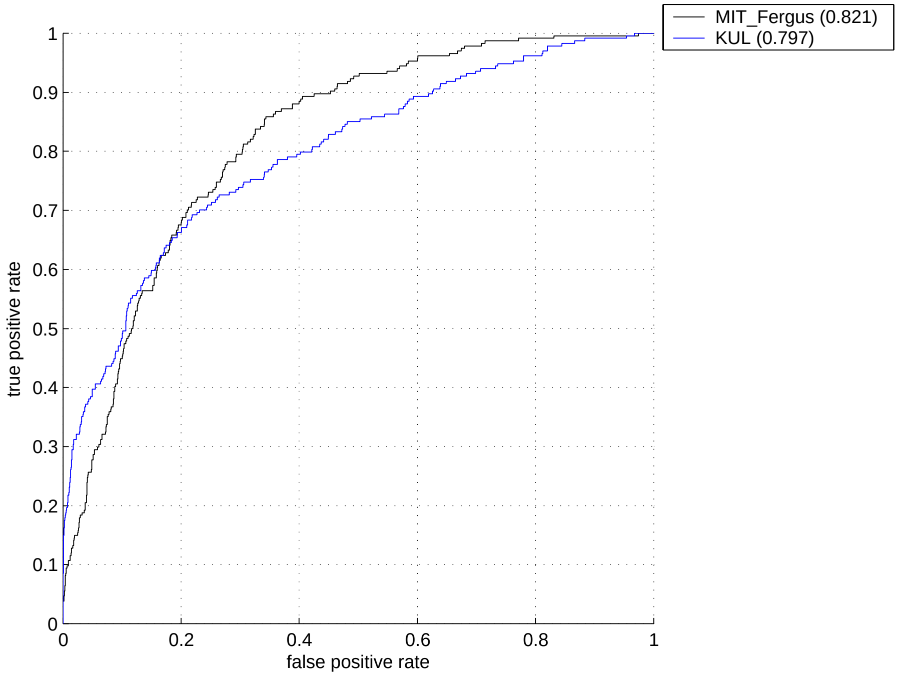

Figure 22: Competition 2.8: motorbike (all entries)

4 Results: Detection

The following sections present the results for the detection competitions. For each competition, the ‘average precision’ (AP) is reported by participant and class. Since the average precision is computed across the full range of recall it penalizes methods which have either overall low precision, or fail to achieve high recall. The ‘best’ result, as measured by AP, is underlined; where several methods obtained the same AP to three decimal places, all are underlined.

4.1 Competition 3

- Train on `trainval` data provided, detect object bounding boxes.

There were five submissions for this competition. Two participants, Cambridge and TKK, submitted results for all ten classes. The other three participants submitted results for different subsets of the classes.

	bicycle	**bus**	**car**	**cat**	**cow**	**dog**	**horse**	**motorbike**	**person**	**sheep**
Cambridge	0.249	0.138	0.254	0.151	0.149	0.118	0.091	0.178	0.030	0.131
ENSMP	–	–	0.398	–	0.159	–	–	–	–	–
INRIA_Douze	0.414	0.117	0.444	–	0.212	–	–	0.390	0.164	0.251
INRIA_Laptev	0.440	–	–	–	0.224	–	0.140	0.318	0.114	–
TKK	0.303	0.169	0.222	0.160	0.252	0.113	0.137	0.265	0.039	0.227
TUD	–	–	–	–	–	–	–	0.153	0.074	–

Table 4: Competition 3

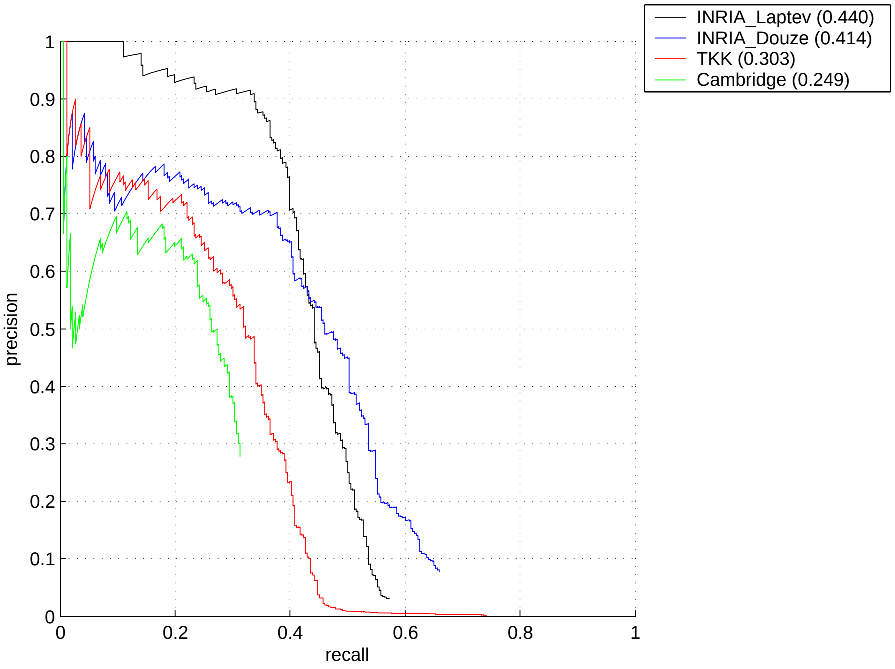

Figure 23: Competition 3.1: bicycle (all entries)

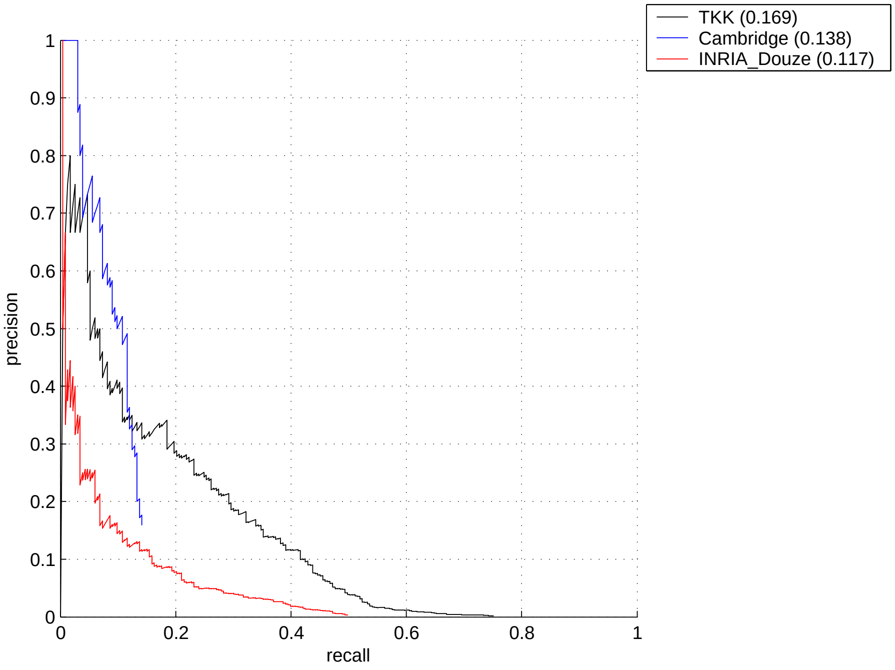

Figure 24: Competition 3.2: bus (all entries)

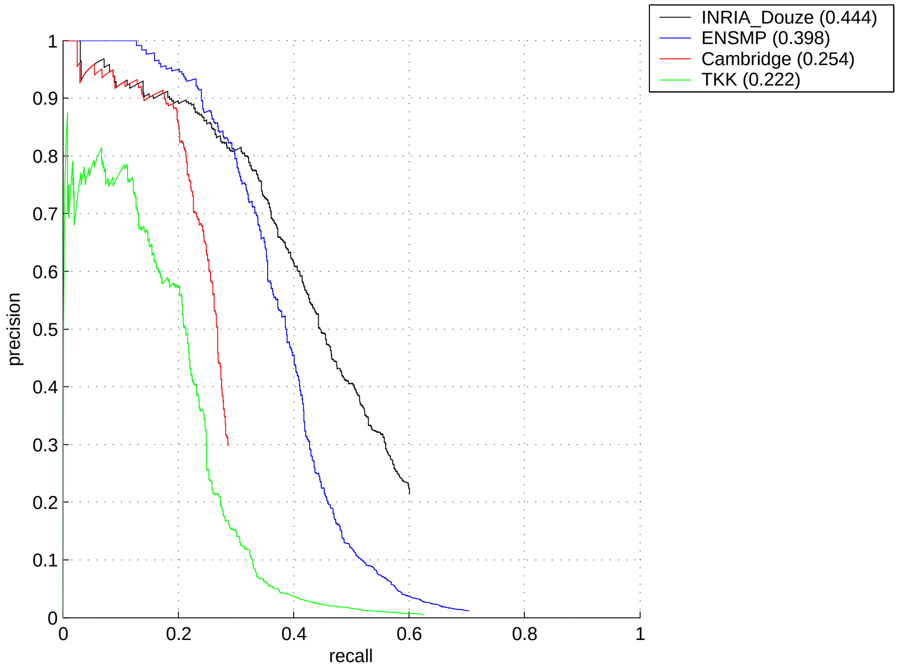

Figure 25: Competition 3.3: car (all entries)

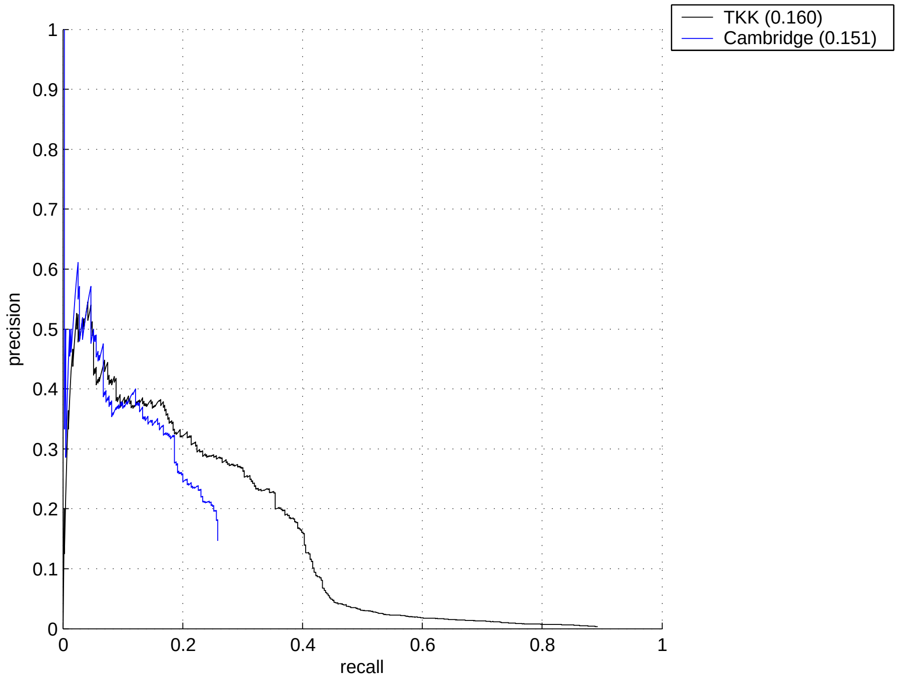

Figure 26: Competition 3.4: cat (all entries)

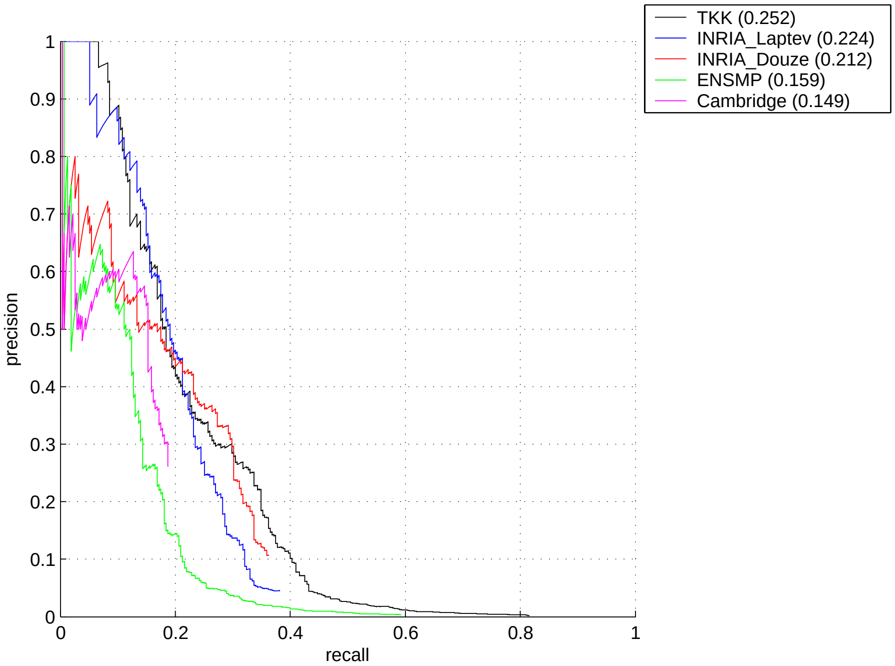

Figure 27: Competition 3.5: cow (all entries)

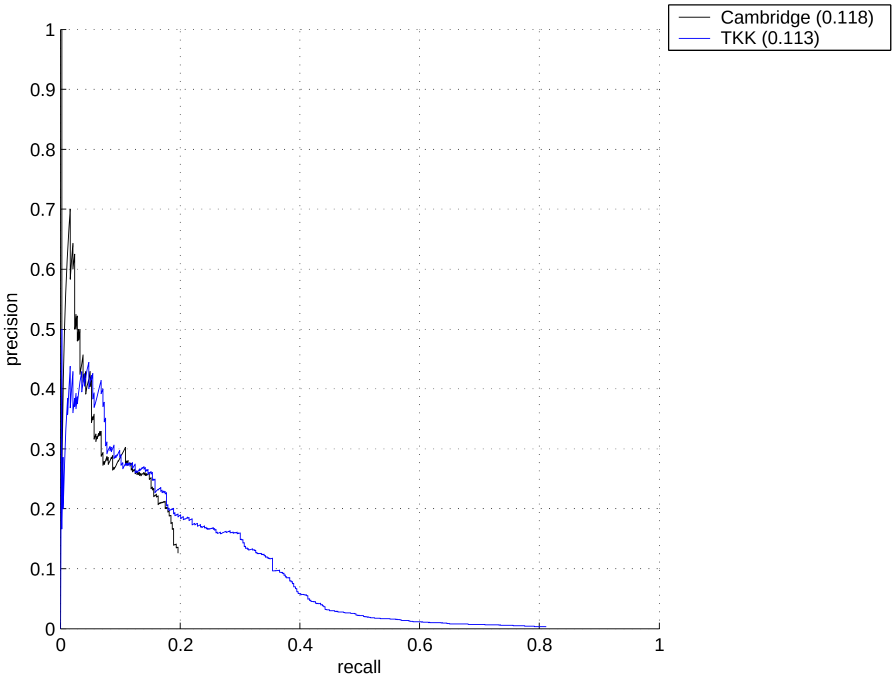

Figure 28: Competition 3.6: dog (all entries)

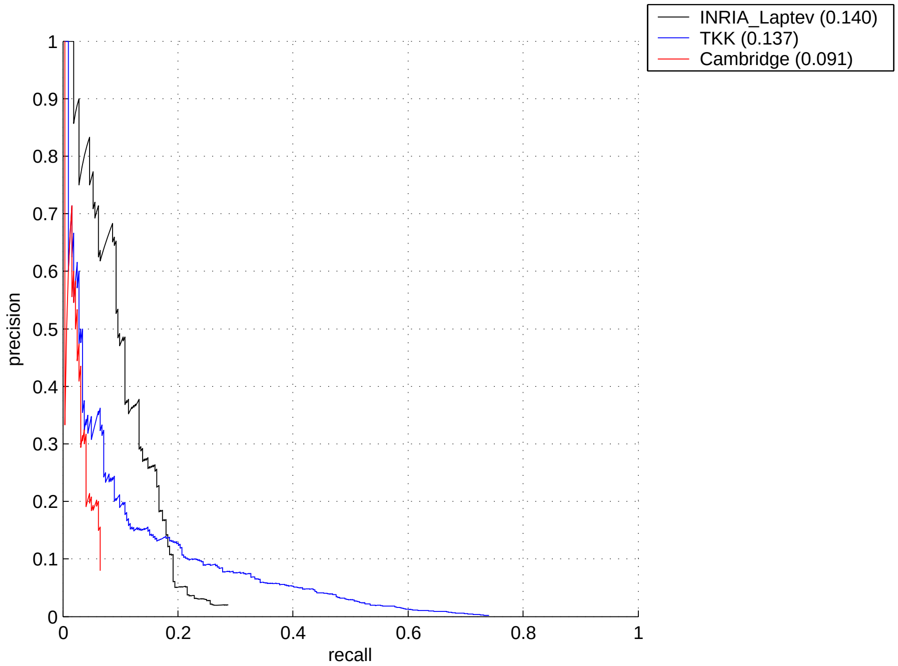

Figure 29: Competition 3.7: horse (all entries)

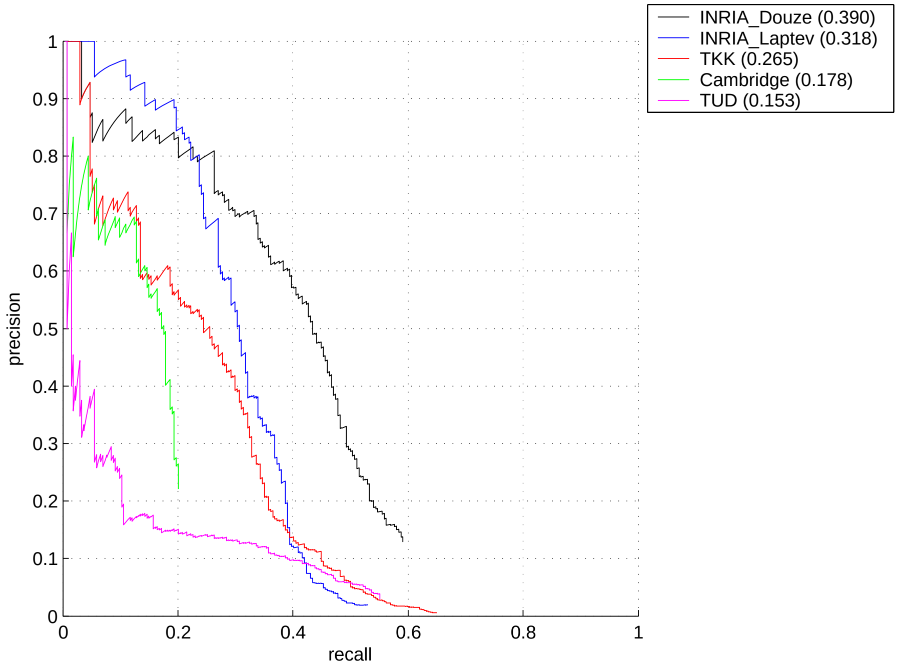

Figure 30: Competition 3.8: motorbike (all entries)

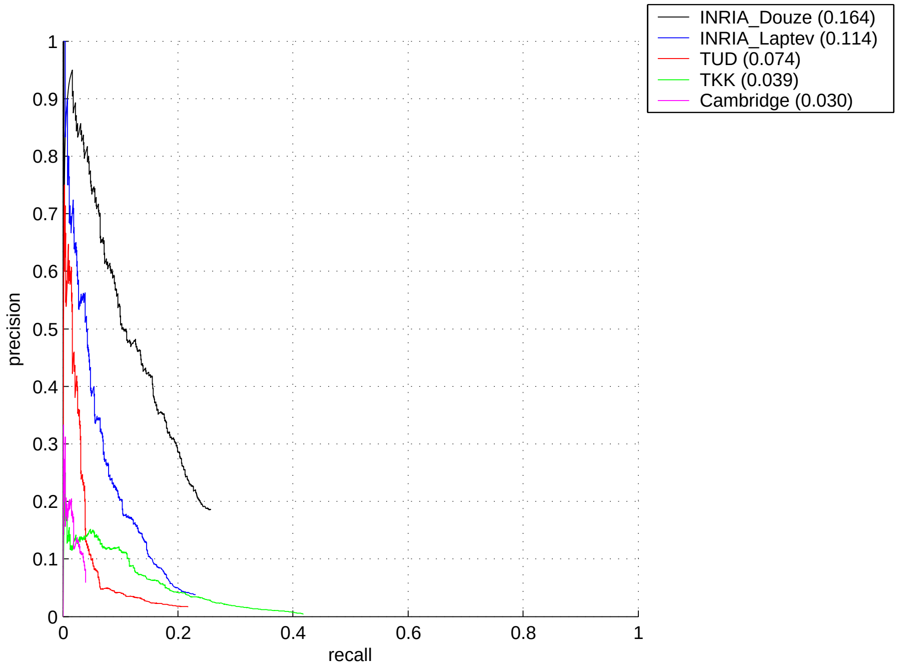

Figure 31: Competition 3.9: person (all entries)

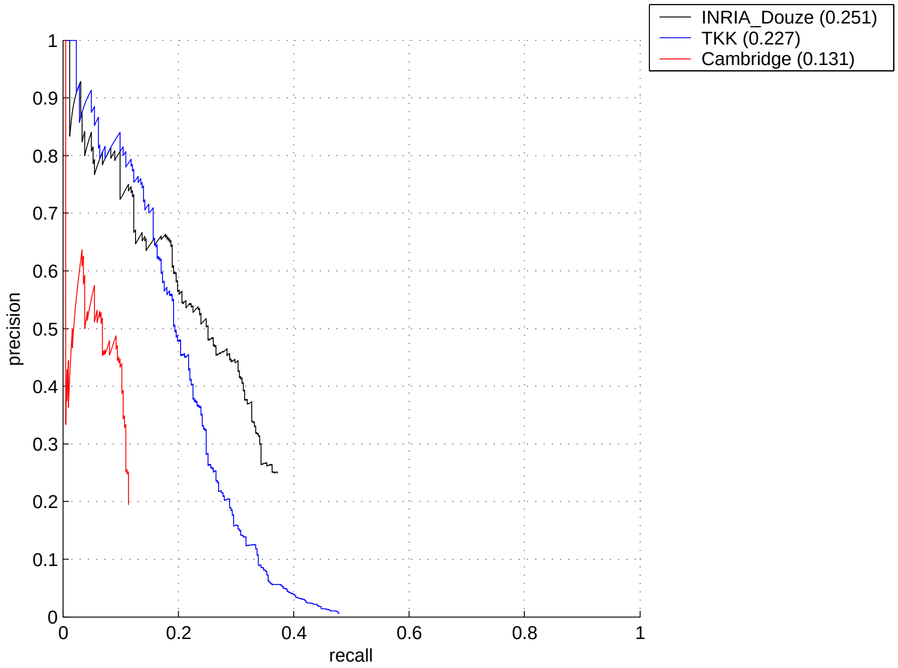

Figure 32: Competition 3.10: sheep (all entries)

4.2 Competition 4

- Train on any (non-test) data, detect object bounding boxes.

Four submissions were received for this competition, in which participants were to submit results of detectors trained on their own data. There were two submissions for 'car' (MIT_Fergus and MIT_Torralba), two for 'motorbike' (KUL and MIT_Fergus), and one for 'person' (INRIA_Douze).

	bicycle	**bus**	**car**	**cat**	**cow**	**dog**	**horse**	**motorbike**	**person**	**sheep**
INRIA_Douze	–	–	–	–	–	–	–	–	0.162	–
KUL	–	–	–	–	–	–	–	0.229	–	–
MIT_Fergus	–	–	0.160	–	–	–	–	0.159	–	–
MIT_Torralba	–	–	0.217	–	–	–	–	–	–	–

Table 5: Competition 4

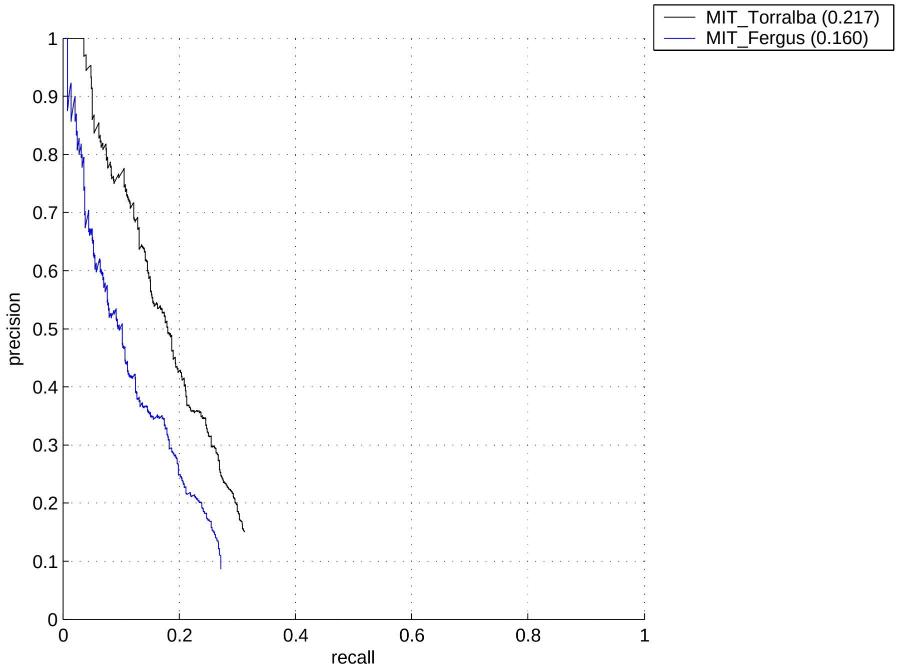

Figure 33: Competition 4.4: car (all entries)

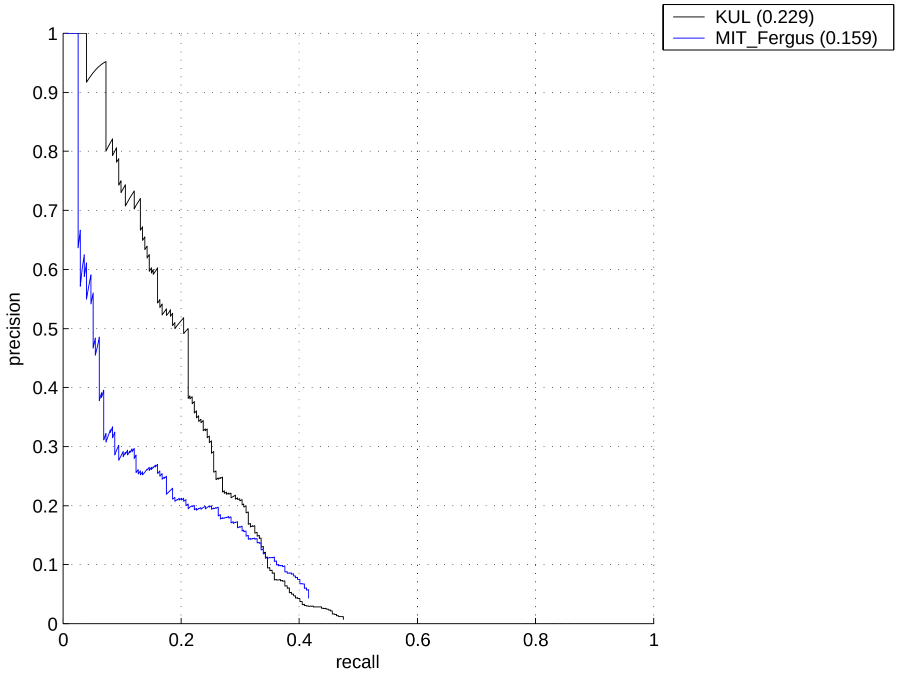

Figure 34: Competition 4.8: motorbike (all entries)

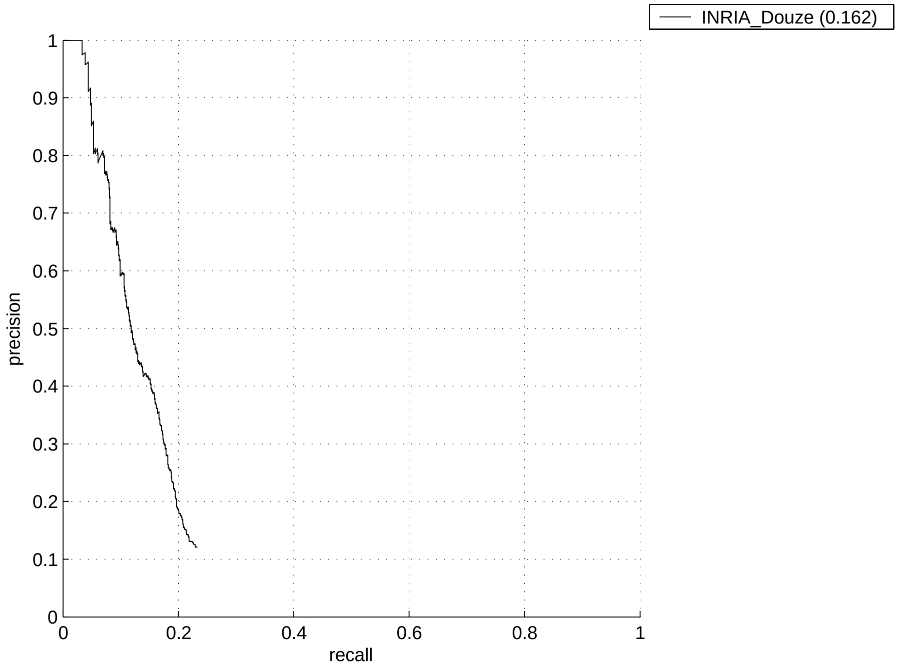

Figure 35: Competition 4.9: person (all entries)

5 Acknowledgements

We are very grateful to those who contributed images and annotation; these include: Moray Allen, James Bednar, Matthijs Douze, Mark Everingham, Stefan Harmeling, Juan Huo, Lindsay Hutchison, Fiona Jamieson, Maria-Elena Nilsback, John Quinn, Florian Schroff, Kira Smyllie, Mark Van Rossum, Chris Williams, John Winn, Andrew Zisserman.

Funding for this challenge was provided by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

A Annotation Guidelines

This appendix lists the guidelines on annotation which were given to annotators.

A.1 Guidelines on what and how to label

What to label. All objects of the defined categories, unless:

- you are unsure what the object is.
- the object is very small (at your discretion).
- less than 10-20% of the object is visible.

If this is not possible because of too many objects, mark the image as bad.

Viewpoint. Record the viewpoint of the 'bulk' of the object e.g. the body rather than the head. Allow viewpoints within 10-20 degrees. If ambiguous, leave as 'Unspecified'.

Bounding box. Mark the bounding box of the visible area of the object (not the estimated total extent of the object). The bounding box should contain all visible pixels, except where the bounding box would have to be made excessively large to include a few additional pixels ($< 5\%$) e.g. a car aerial.

Occlusion/truncation. If more than 15-20% of the object is occluded and lies outside the bounding box, mark as 'Truncated'. Do not mark as truncated if the occluded area lies within the bounding box.

Image quality/illumination. Images which are poor quality (e.g. excessive motion blur) should be marked bad. However, poor illumination (e.g. objects in silhouette) should not count as poor quality unless objects cannot be recognized.

Clothing/mud/snow etc. If an object is 'occluded' by a close-fitting occluder e.g. clothing, mud, snow etc., then the occluder should be treated as part of the object.

Transparency. Do label objects visible through glass, but treat reflections on the glass as occlusion.

Mirrors. Do label objects in mirrors.

Pictures. Label objects in pictures/posters/signs only if they are photorealistic but not if cartoons, symbols etc.

A.2 Guidelines on categorization

Car. Includes cars, vans, people carriers etc. Do not label where only the vehicle interior is shown.

A.3 "Difficult" flag

Objects were marked as "difficult" by a single annotator. Only the image area corresponding to the bounding box of each object was displayed and a subjective judgement of the difficulty of recognizing the object was made. Reasons for marking an object as difficult included small image area, blur, clutter, high level of occlusion, occlusion of a very characteristic part of the object, etc.