Pascal Visual Object Classes Challenge Results

Mark Everingham, Luc Van Gool,
Chris Williams, Andrew Zisserman

April 5, 2005

Abstract

The goal of this challenge is to recognize objects from a number of visual object classes in images of realistic scenes. It is fundamentally a supervised learning learning problem in that a training set of labelled images is provided. The object classes are: motorbikes, bicycles, people and cars. Twelve participants entered the challenge. A full description of the challenge including software and image sets is available on the web page `http://www.pascal-network.org/challenges/VOC/voc/index.html`.

Contents

1 Challenge

The goal of this challenge is to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects). There are four object classes:

- motorbikes
- bicycles
- people
- cars

There were two main competitions:

- Classification – predicting presence/absence of an example of that class in the test image
- Detection – Predicting the bounding box and label of each object from the 4 target classes in the test image

These are described in more detail below. Contestants could enter either (or both) of these competitions, and could choose to tackle any (or all) of the four object classes. The challenge allows for two approaches to each of the competitions:

1. Training using any data excluding the provided test sets
2. Training using only the provided training data

The intention in the first case is to establish just what level of success can currently be achieved on these problems and by what method; in the second case the intention is to establish which method is most successful given a specified training set.

1.1 Classification task

For each of the four object classes predict the presence/absence of at least one object of that class in a test image. The output of the classifier is 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 four 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. 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:

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

Class	Images	Objects
motorbikes	107	109
bicycles	57	63
people	42	81
cars	136	159
Total	342	

(a) `train`

Class	Images	Objects
motorbikes	107	108
bicycles	57	60
people	42	71
cars	136	161
Total	342	

(b) `val`

Class	Images	Objects
motorbikes	214	217
bicycles	114	123
people	84	152
cars	272	320
Total	684	

(c) `train+val`

Class	Images	Objects
motorbikes	216	220
bicycles	114	123
people	84	149
cars	275	341
Total	689	

(d) `test1`

Class	Images	Objects
motorbikes	202	227
bicycles	279	399
people	526	1038
cars	275	381
Total	1282	

(e) `test2`

Table 1: Number of images (containing at least one object of the corresponding class) and object instances in the image sets

MATLAB code for computing this overlap measure is given in the example code. Multiple detections of the *same* object in an image will be 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 user's system to filter multiple detections from its output.

1.3 Image sets

There are five sets of images provided. The image sets are to be used both for the classification and detection tasks.

`train`: Training data

`val`: Validation data (suggested). The validation data may be used as additional training data (see below).

`train+val`: The union of `train` and `val`.

`test1`: First test set. This test set is taken from the same distribution of images as the training and validation data, and is expected to provide an 'easier' challenge.

`test2`: Second test set. This test set has been freshly collected for the challenge. It is not therefore expected to have the same distribution as the training data, and should provide a 'harder' challenge.

Statistics of each of the image sets are summarized in table 1.

1.4 Competitions

Eight competitions are defined according to the task, the choice of training data: (i) taken from the VOC `train+val` data provided, or (ii) from any source excluding the VOC {`test1`|`test2`} data provided; and the choice of test data: (i) `test1` ('easier') or (ii) `test2` ('harder'):

No.	Task	Training data	Test data
1	Classification	`train+val`	`test1`
2	Classification	`train+val`	`test2`
3	Classification	**not** VOC `test1` or `test2`	`test1`
4	Classification	**not** VOC `test1` or `test2`	`test2`
5	Detection	`train+val`	`test1`
6	Detection	`train+val`	`test2`
7	Detection	**not** VOC `test1` or `test2`	`test1`
8	Detection	**not** VOC `test1` or `test2`	`test2`

To emphasize, in competitions 3–4 and 7–8, *any* source of training data may be used *except* the provided test data `test1` or `test2`. Competitions 1–2 and 5–6 must use *only* the provided test data `train` and `val`.

Note that any annotation provided in the VOC `train` and `val` sets may be used for training, for example bounding boxes, particular class labels e.g. `PAScarFrontal` or `PAScarSide`, polygonal outlines where provided, etc.

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

2 Participants

This section lists (in no significant order) the 12 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 Aachen

Participants: Thomas Deselaers, Daniel Keysers
Affiliation: Lehrstuhl fur Informatik VI
Computer Science Department
RWTH Aachen University
52062 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 Euclidean 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.

Extensions. This baseline method was improved by the following extensions: To be able to account better for objects of different sizes, we extract image patches of various sizes at each extraction point instead of extraction image patches from one size only. The extracted patches are then scaled to a common size for the remaining steps. Additionally it was observed that the first PCA coefficient accounts strongly for overall brightness of the image patches and thus we discarded it obtaining a method which is more invariant to lighting changes.

References

[1] T. Deselaers, D. Keysers, and H. Ney. Discriminative Training for Object Recognition using Image Patches. In Proc. CVPR05, San Diego, CA, in press, June 2005.

2.2 Darmstadt

Participants: Mario Fritz, Bastian Leibe, Edgar Seemann, Bernt Schiele
Affiliation: TU-Darmstadt
E-mail: mario.fritz@informatik.tu-darmstadt.de

We submit results on the categories car and motorbike obtained with 2 different approaches :

- ISM, as presented in [1].
- ISM integrated with a novel SVM verification stage.

In both approaches, we use a codebook representation as a first generalization step. The ISMs are trained on the following subsets of the training/validation sets:

- 55 car images consisting of
  - 26 images of the TU Darmstadt database
  - 29 images of the TU Graz database
- 153 motorbike images of the CalTech database

Up to now, our approaches have only been evaluated on single viewpoints. In order to stay consistent with those experiments, we only selected side views from the training set. As the ISM facilitates the use of segmentation masks for increased performance, we included the provided annotations in the training. The SVM validation stage is trained on detections and false alarms of the ISM on the whole training set for cars and motorbikes.

All experiments were performed on the test sets exactly as specified in the PASCAL challenge. For computational reasons, the test images were rescaled to a uniform width of 400 pixels. We report results on both the object detection and the present/absent classification task. Detection performance is evaluated using the hypothesis bounding boxes returned by the ISM approach. For the classification task, an object-present decision is taken if at least one hypothesis is detected in an image. Since our integrated ISM+SVM approach allows for an additional precision/recall tradeoff, we report two performance curves for the detection tasks one for optimal equal error rate (EER) performance and one for optimized precision (labeled ISM+SVM v2 in the plots).

References

[1] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. pages 17–32, ECCV 2004, Prague, Czech Republic, May 2004.

2.3 Edinburgh

Participants: Tom Griffiths, Moray Allan, Amos Storkey, Chris Williams
Affiliation: University of Edinburgh
E-mail: moray@sermisy.org

Our aim in these experiments was to assess the performance that can be obtained using a simple approach based on classifiers and detectors using SIFT representations of interest points. We deliberately did not use state-of-the-art class-specific detectors. All the systems described below begin by detecting Harris-Affine interest points in images1 [3]. SIFT representations are then found for the image regions chosen by the interest point detector [2]. The SIFT representations for all the regions chosen in the training data are then clustered using k-means. A test image can now be represented as a vector of activations by matching the SIFT representation of its interest point regions against these clusters and counting how many times each cluster was the best match for a region from the test image. This approach was suggested by recent work of Csurka, Dance et al. [1]. All the systems were trained only on the original training data, with parameters optimised using the validation data. The test data sets were only used in the final runs of the systems to obtain results for submission.

Edinburgh_C_bagoffeatures_train. This classifier uses logistic regression, based on a 1500-dimensional bag-of-features representation of each image. Interest points were detected using the Harris-Affine region detector and encoded as SIFT descriptors. These were pooled from all images in the training set and clustered using simple k-means ($k = 1500$). The 1500-dimensional bag-of-features representation for each image is computed by counting, for each of the 1500 cluster centres, how many regions in the image have no closer cluster centre in SIFT space. This run was trained on the train data set.

Edinburgh_D_meanbbox_train. This naive approach is intended to act as a baseline result. All images in the test set are assigned the class probability as their confidence level. This class probability is calculated from the class frequency as the number of positive examples of the class in the training set divided by the total number of training images. All detections are made using the class mean bounding box, scaled according to the size of the image. The class mean bounding box is calculated by finding all the bounding boxes for this class in the training data, and normalising them with respect to the sizes of the images in which they occur, then taking the means of the normalised coordinates. This detector assumes that there is one object in each test image. This run was trained on the train data set.

Edinburgh_D_wholeimage_train. This naive approach is intended to act as a baseline result. All images in the test set are assigned the class probability as their confidence level. This class probability is calculated from the class frequency as the number of positive examples of the class in the training set divided by the total number of training images. The object bounding box is

simply set to the perimeter of the test image. This detector assumes that there is one object in each test image. This run was trained on the train data set.

Edinburgh_D_purityweightedmeanbbox_train. We define the purity of a cluster with respect to an object class as the fraction of all the Harris-Affine interest points in the training images for which it is the closest cluster in SIFT space (subject to a maximum distance threshold t) that are located within a bounding box for an object of the class. In detection, the centre of the bounding box is set as the weighted mean of the location of all Harris-Affine interest points in the test image, where the weight of each interest points location is the purity of its nearest cluster in SIFT space (with respect to the current object class, subject to a maximum distance threshold t). All detections are made using the class mean bounding box, scaled according to the size of the image. The class mean bounding box is calculated by finding all the bounding boxes for this class in the training data, and normalising them with respect to the sizes of the images in which they occur, then taking the means of the normalised coordinates. Confidences are calculated by the bag-of-features classifier, as described for `Edinburgh_C_bagoffeatures_train`, with the addition of a maximum distance threshold t (so descriptors very far from any cluster do not count). Throughout, t was set to three times the standard deviation of the distances of all SIFT descriptors from their nearest cluster centre, a value chosen by experiment on the validation data. This detector assumes one object in each test image. This run was trained on the train data set.

Edinburgh_D_siftbbox_train. This detector assigns the confidence levels calculated by the bag-of-features classifier, as described for `Edinburgh_C_bagoffeatures_train`, while bounding boxes are predicted as the tight bounding box of the interest points found in the image by the Harris-Affine detector. This detector assumes that there is one object in each test image. This run was trained on the train data set.

References

[1] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual Categorization with Bags of Keypoints. In Workshop on Statistical Learning in Computer Vision, at ECCV, 2004.

[2] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2):91–110, 2004.

[3] Krystian Mikolajczyk and Cordelia Schmid. Scale & Affine Invariant Interest Point Detectors. International Journal of Computer Vision, 60(1):63–86, 2004.

2.4 FranceTelecom

Participants: Stefan Duffner, Christophe Garcia
Affiliation: Image Indexing and Coding Group
France Telecom division R&D
4 rue du clos Courtel
35512 Cesson Svign
France
E-mail: christophe.garcia@francetelecom.com

Our method is based on a convolutional neural network architecture as described in: "Convolutional face finder: a neural architecture for fast and robust face detection" by Christophe Garcia, and Manolis Delakis. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 26, Issue 11, Nov. 2004 Page(s):1408 - 1423

The network is basically composed of an input layer of fixed size (the so-called retina) and several alternating convolution and subsampling layers each containing several neuron 'maps' which are eventually connected to a single neuron indicating whether the input image corresponds to the searched object (motorbike, car etc.) or not. This neural architecture allows a supervised training of the system as a whole just by presenting an input pattern and desired output (-1 or 1) and using a back-propagation algorithm. Thus the training and validation sets contain positive as well as negative examples, i.e. images that show an example of the object to learn and images that don't).

In our case, positive examples are created by 'cutting out' objects from the supplied training data using the given annotations and resizing them to the retina dimensions. Negative examples were created using the same images by cutting out regions that don't contain the object. Additionally, the images are converted into grey scale images, so colour information is not used at all by this approach.

Having accomplished the training for a given object the neural network can be applied to images of arbitrary size using a sliding window approach. At each step the window content is resized to fit into the retina of the neural network. This search procedure is repeated for different scales and different granularity finally giving a number of enclosing bounding boxes that are supposed to contain an example of the corresponding object class. Note that by using this method no further pre-processing (contrast enhancement, noise reduction etc.) of the input data is required.

In principle, the method can be applied to any object simply by adjusting the training set and optionally the dimensions (ratio) of the retina.

2.5 HUT

Participants: Ville Viitaniemi, PicSOM-group
Affiliation: PicSOM-group
Laboratory of Computer and Information Science
Helsinki University of Technology
E-mail: Ville.Viitaniemi@hut.fi

For all the experiments we used a similar setup utilising the PicSOM general purpose content-based image retrieval system. The system operates by the principle of query-by-examples. Given a set of positive and negative example images, the system looks through the image collection and returns images most similar to the positive and most dissimilar to the negative example images. Normally the system operates on-line, gathering the example image sets incrementally by user supplied relevance feedback. In this experiment, however, the system was operated in batch mode as if the user had given relevance feedback on all images in the training set at once.

The visual features that we used to describe the images were chosen among the ones that happened to be available in the PicSOM system. These are targeted to the general domain image description, i.e. the feature set was not specialised a priori to the target image classes. Within the limits of the effort we allocated to the VOC challenge we were not able to utilise the training set annotations beyond the present/absent information, i.e. the bounding boxes and other additional annotations were not used. All the results were obtained by using only the provided training+validation data as training set. System parameters were tuned using the validation set performance as a target.

Details of the experimental procedure

The training set images were automatically segmented to a few parallel segmentations with predetermined numbers of segments. Set of features were extracted from the segments and the whole images. The features were partitioned into subsets for each which Tree-Structured SOMs were trained. The system was used for classification by

1. extracting features similarly from the test set images
2. projecting training set images to the parallel feature TS-SOM bottom levels
3. comparing locally the distance of projection of a given test image to nearby projections of positive and negative training images by means of kernel smoothing
4. combining the results from parallel feature TS-SOMs

The set of features was chosen among the available features to give good performance in the validation set. As performance measure we used the area under the ROC curve. The set of available features consisted of:

- MPEG-7 content descriptors ColorLayout, DominantColor, EdgeHistogram, RegionShape and ScalableColor
- average colour in CIE L*a*b* colour space
- first three colour moments in CIE L*a*b* colour space
- Fourier descriptors of object contours
- a texture feature measuring relative brightness of neighbouring pixels

All four target classes were processed separately. More details can be found in e.g. [1,2].

References

[1] Jorma Laaksonen, Markus Koskela, Sami Laakso, and Erkki Oja. Self-organizing maps as a relevance feedback technique in content based image retrieval. Pattern Analysis & Applications, 4(2+3):140–152, June 2001.

[2] Jorma Laaksonen, Markus Koskela and Erkki Oja. PicSOM - Self-organizing image retrieval with MPEG-7 content descriptions. IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing, 13(4):841–853, July 2002.

2.6 INRIA: dalal

Participants: Navneet Dalal, Bill Triggs
Affiliation: LEAR group, INRIA Rhone-Alpes
E-mail: navneet.dalal@inrialpes.fr

We use dense grid of histogram of oriented gradients, similar to SIFT, as descriptor over a fixed sized window adopting linear SVM as a classifier. Test images are searched over full scale-space, followed by non-maximum suppression. Further details of method will be presented at CVPR 2005.

2.7 INRIA: dorko

Participants: Gyuri Dorko, Cordelia Schmid
Affiliation: LEAR group, INRIA Rhone-Alpes
E-mail: gyuri.dorko@inrialpes.fr

We used sparse local features to described images: Our modified version of the scale-invariant Harris detector with SIFT (D.Lowe) representation was used to extract features. We run clustering and feature selection in a semi-supervised manner as described in RR-5497 (INRIA Technical Report) "Object Class Recognition Using Discriminative Local Features", our INRIA technical report. We used 1200 clusters on the whole training set and selected the best 100 for each category. We built statistics on the position of these 100 clusters relative to the center and scale of the bounding boxes on the training images. For detection we used the normalized Hough transform to estimate the location, width and height of the objects on the test images. This method is similar to the voting step of Leibe et al. "Scale Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search" DAGM04. The difference here, is that we vote on a 4 dimensional space (we use separate width and height) and we also use the ranks of the clusters from our feature selection as weight in the Mean-Shift space. Unfortunatelly, our method is not well adapted to multi-label cases, therfore we only allow one detection per image causing a bias to better precision in the cost of lower recall.

2.8 INRIA: jurie

Participants: Diane Larlus, Gyuri Dorko, Frederic Jurie, Bill Triggs
Affiliation: LEAR group, INRIA Rhone-Alpes
E-mail: frederic_jurie@inrialpes.fr

The approach is based on a linear SVM classifier trained with given learning and validation images. Each image is summarized by a feature vector providing a description of the image content.

To build feature vectors, a codebook is computed by quantizing a multi-scale dense sift representation of training images. We use for this step a clusterer specially made for this kind of quantization. Because of dense sampling, the set of vectors to be quantized is huge and the distribution of vectors makes the quantization process very specific.

Feature vectors are built in two different ways:

- p1 files: features vectors are binary vectors where 1 means that the corresponding codeword occurs at last one time (at any scale) in the image.
- p2 files: features are normalized histogram representative of codeword frequencies

In both case a 2000-centres long codebook is used. Results slightly decrease if the codebook is smaller, but the length of the codebook is not critical.

2.9 INRIA: zhang

Participants: Jianguo Zhang, Cordelia Schmid
Affiliation: INRIA Rhone-Alpes
E-mail: jianguo.zhang@inrialpes.fr

The approach learns a representation for an object category with a kernel-based classifier and a sparse image representation. The steps of the approach are the following:

- Extraction of a sparse set of descriptors. Here the descriptors are extracted for scale-invariant interest regions.
- Vocabulary construction. Here we use k-means clustering to obtain a vocabulary.
- Distance between images. Here we use the χ^2-distance to compare frequency histograms based on the vocabulary.
- Classification. Here we use Support Vector Machines and a χ^2-kernel.

Each of the points is detailed in the following.

Sparse image representation. Harris-Laplace [1] and Laplacian [2] interest regions are extracted for each image. SIFT [3] is used as region descriptor, resulting in 128 dimensional description vectors. Note that the version of SIFT used here is not rotation invariant. For the training images and test set 1 (1373 images in total), the average number of points detected per image is 796 for Harris-Laplace and 2465 for the Laplacian. The minimum number of points detected for an image is 15 (Harris-Laplace) and 71 (Laplacian).

Vocabulary construction. We cluster the descriptors of each class separately and then concatenate them. Here we extract 250 clusters per class with the k-means algorithm. The concatenation results in 1000 clusters; the cluster centers are called visual words in the following.

Distance between images. For each image we compute a frequency histogram for our set of visual words. The distance between images is then obtained by comparing these histograms with the χ^2 distance:

$$\chi^2(h_1, h_2) = \sum_i \frac{(h_1(i) - h_2(i))^2}{(h_1(i) + h_2(i)}$$

where h_1 and h_2 are the vocabulary histograms of two different images.

Classification. We use Support Vector Machines (SVM) for classification [5]. Our kernel is a Gaussian kernel based on the χ^2-distance [4]:

$$K(I_1, I_2) = \exp(-1/A \cdot \chi^2(h_1, h_2))$$

The parameter A is obtained by 5-fold cross validation on the training images. The distance between images is computed separately for each detector/descriptor pair. Results are combined by adding the distances and estimating A for the combination. Here we combine Harris- Laplace/SIFT and Laplacian/SIFT. For each of the 4 classes we train a binary classifier which separates a class from the others. The output of the SVM is normalized to [0; 1] and used as a confidence measure.

References

[1] K. Mikolajczyk and C. Schmid, .Scale and affine invariant interest point detectors,. International Journal of Computer Vision, vol. 60, no. 1, pp. 63–86, 2004.

[2] J. Garding and T. Lindeberg, "Direct computation of shape cues using scale-adapted spatial derivative operators," International Journal of Computer Vision, vol. 17, no. 2, pp. 163–191, Feb. 1996.

[3] D. G. Lowe, "Distinctive image features form scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[4] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," IEEE Trans. on Neural Networks, vol. 10, no. 5, pp. 1055–1064, Oct. 1999.

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

2.10 METU

Participants: Ilkay Ulusoy
Affiliation: METU
EEE Department
06531 Ankara
Turkey
E-mail: ilkay@metu.edu.tr

Please contact the participants for a description.

2.11 MPITuebingen

Participants: Jan Eichhorn, Olivier Chapelle
Affiliation: Max Planck Institute for Biological Cybernetics
Spemannstr. 38
72076 Tuebingen
Germany
E-mail: {je|chapelle}@tuebingen.mpg.de

Our method uses the support vector algorithm combined with a SIFT-feature extractor. In contrast to original SIFT we use instead a Harris corner detector to extract interest points. More details can be found in the MPI technical report 'Object categorization with SVM: kernels for local features' available from: http://www.kyb.mpg.de/publication.html?user=je.

2.12 Southampton

Participants: Jason D. R. Farquhar, Hongying Meng, Sandor Szedmak, John Shawe-Taylor
Affiliation: ISIS group
School of Electronics and Computer Science
University of Southampton
E-mail: ss03v@ecs.soton.ac.uk

Our method consists of two main phases, a machine vision phase which computes highly discriminative local image features, and a machine learning phase which learns the image categories based upon these features. We present two main innovations. Firstly, we use the Bhattacharyya kernel, which is designed to work with sets of vectors, to measure the similarity of the distribution of local features in each image. Secondly, we use a multiple-feature extension of the well-known maximum margin SVM learner, to combine different features so we can exploit the advantages that can vary among the features.

Image feature extraction. On every image an interest point detector is applied to find the *interesting* local patches of the image (usually centred around corners). To increase robustness to common image transformations (such as illumination or perspective) a local feature is generated for each patch using the *SIFT* descriptor. This describes a patch in terms of 8 directional gradients computed at 16 different positions within the patch, giving a 128 dimensional output. These are the base features.

Dimensionality reduction. The dimension of the base features is relatively large employing dimension reduction may improve the generalisation and diminish the overall training time. The two types of dimensionality reduction tried are; Principal Component analysis (PCA) and Partial Least Squares Regression (PLS).

PDF Computation. The output of the image feature generation and dimensionality reduction steps is a set of local image features per image. As most machine learning algorithms cannot cope with variable length feature sets histogramming can be used to map from sets of vectors to a fixed length representations. An alternative to mapping the sets of feature to fixed length representations so conventional learning methods can be used is to use a learning algorithm which works directly with feature sets. The approach used here is to model the set of features as a probability distribution (PDF) over feature space and then define a kernel between such PDFs. This kernel can then be used in any conventional kernel learning algorithm, such as the SVM. In this work the set of features is modelled using a full covariance Gaussian distribution and the Bhattacharyya kernel, $K(\mathrm{Pr}_1(x), \mathrm{Pr}_2(x)) = \int \sqrt{\mathrm{Pr}_1(x)}\sqrt{\mathrm{Pr}_2(x)}dx$, to measure similarity of distributions.

Classifier Learning. To date only maximum margin based classifiers have been used, specifically either a conventional SVM or our modified multi-feature SVM, called SVM_2K, which allows us to combine two (or more) input features and compute a single combined classifier. This is sometimes called co-training or multi-view learning.

Results. We report three sets of results, all based upon the PDF approach with the Bhattacharyya kernel, differing only in the types of interest point detector used; Laplacian of Gaussians (LoG) or multi-scale harris affine (Har-Aff), and the classifier used, single SVM for each feature type or combined features with SVM_2K.

3 Results: Classification

The following sections present the results for the classification competitions. For each competition and object class the equal error rates (EER) and area under ROC curve (AUC) are presented. The region of the ROC curves shown covers an area of 1.5 times the equal error rate. The curves have been sorted by decreasing equal error rate to aid visibility of the most successful entries. Two ROC curves are shown: (i) all submitted results, and (ii) the best result

submitted by each participant, with the top 5 results shown; for the latter, equal error rate was used to rank results.

The 'best' result in each competition and for each object class is indicated by an asterisk in the tables. As can be seen, choice of EER or AUC as the measure of success changes the choice of 'best' result in only two cases, for which the EER is equal for the top two results, and the small difference in AUC might reasonably be considered insignificant.

3.1 Competition 1

- Train on provided data, classify object present/absent in `test1`

10 of the 12 participants took part in this competition. All but one participant (Darmstadt) tackled all four object classes.

Participant	motorbikes	bicycles	people	cars
Aachen	×	×	×	×
Darmstadt	×	–	–	×
Edinburgh	×	×	×	×
FranceTelecom	–	–	–	–
HUT	×	×	×	×
INRIA: dalal	–	–	–	–
INRIA: dorko	–	–	–	–
INRIA: jurie	×	×	×	×
INRIA: zhang	×	×	×	×
METU	×	×	×	×
MPITuebingen	×	×	×	×
Southampton	×	×	×	×

Table 2: Competition 1 participation

Submission	EER	AUC
Aachen: motorbikes-test1-ms-2048-histo	0.926	0.979
Aachen: motorbikes-test1-n1st-1024	0.940	0.987
Darmstadt: ISMbig3	0.829	0.919
Darmstadt: ISMSVMbig3	0.856	0.882
Edinburgh: Edinburgh_C_bagoffeatures_train	0.722	0.765
HUT: hut_final1	0.921	0.974
HUT: hut_final2	0.917	0.970
HUT: hut_final3	0.912	0.952
HUT: hut_final4	0.898	0.960
INRIA: jurie: dcb_p1	0.968	0.997
INRIA: jurie: dcb_p2	0.977*	0.998*
INRIA: zhang: prediction	0.964	0.996
METU: ms_metu	0.903	0.966
MPITuebingen: Pascal_FINAL_test1	0.875	0.945
Southampton: pascal_develtest	0.972	0.994
Southampton: UoS_LoG.SIFT.PLS20ppker	0.949	0.989
Southampton: UoS_mhar.aff.SIFT.PLS20ppker	0.940	0.985

Table 3: Competition 1.1: test1: motorbikes

Submission	EER	AUC
Aachen: bicycles-test1-ms-2048-histo	0.842	0.931
Aachen: bicycles-test1-n1st-1024	0.868	0.954
Edinburgh: Edinburgh_C_bagoffeatures_train	0.689	0.724
HUT: hut_final1	0.795	0.891
HUT: hut_final2	0.816	0.895
HUT: hut_final3	0.781	0.864
HUT: hut_final4	0.767	0.880
INRIA: jurie: dcb_p1	0.918	0.974
INRIA: jurie: dcb_p2	0.930*	0.981
INRIA: zhang: prediction	0.930*	0.982*
METU: ms_metu	0.781	0.822
MPITuebingen: Pascal_FINAL_test1	0.754	0.838
Southampton: pascal_develtest	0.895	0.961
Southampton: UoS_LoG.SIFT.PLS20ppker	0.868	0.943
Southampton: UoS_mhar.aff.SIFT.PLS20ppker	0.851	0.930

Table 4: Competition 1.2: test1: bicycles

Submission	EER	AUC
Aachen: people-test1-ms-2048-histo	0.861	0.928
Aachen: people-test1-n1st-1024	0.861	0.936
Edinburgh: Edinburgh_C_bagoffeatures_train	0.571	0.597
HUT: hut_final1	0.850	0.927
HUT: hut_final2	0.833	0.931
HUT: hut_final3	0.845	0.919
HUT: hut_final4	0.857	0.921
INRIA: jurie: dcb_p1	0.917*	0.979*
INRIA: jurie: dcb_p2	0.901	0.965
INRIA: zhang: prediction	0.917*	0.972
METU: ms_metu	0.803	0.816
MPITuebingen: Pascal_FINAL_test1	0.731	0.834
Southampton: pascal_develtest	0.881	0.943
Southampton: UoS_LoG.SIFT.PLS20ppker	0.833	0.918
Southampton: UoS_mhar.aff.SIFT.PLS20ppker	0.841	0.925

Table 5: Competition 1.3: test1: people

Submission	EER	AUC
Aachen: cars-test1-ms-2048-histo	0.925	0.978
Aachen: cars-test1-n1st-1024	0.920	0.979
Darmstadt: ISMbig4	0.548	0.578
Darmstadt: ISMSVMbig4	0.644	0.717
Edinburgh: Edinburgh_C_bagoffeatures_train	0.793	0.798
HUT: hut_final1	0.869	0.956
HUT: hut_final2	0.908	0.968
HUT: hut_final3	0.847	0.934
HUT: hut_final4	0.909	0.971
INRIA: jurie: dcb_p1	0.961*	0.992*
INRIA: jurie: dcb_p2	0.938	0.987
INRIA: zhang: prediction	0.937	0.983
METU: ms_metu	0.840	0.920
MPITuebingen: Pascal_FINAL_test1	0.831	0.918
Southampton: pascal_develtest	0.913	0.972
Southampton: UoS_LoG.SIFT.PLS20ppker	0.898	0.959
Southampton: UoS_mhar.aff.SIFT.PLS20ppker	0.901	0.961

Table 6: Competition 1.4: test1: cars

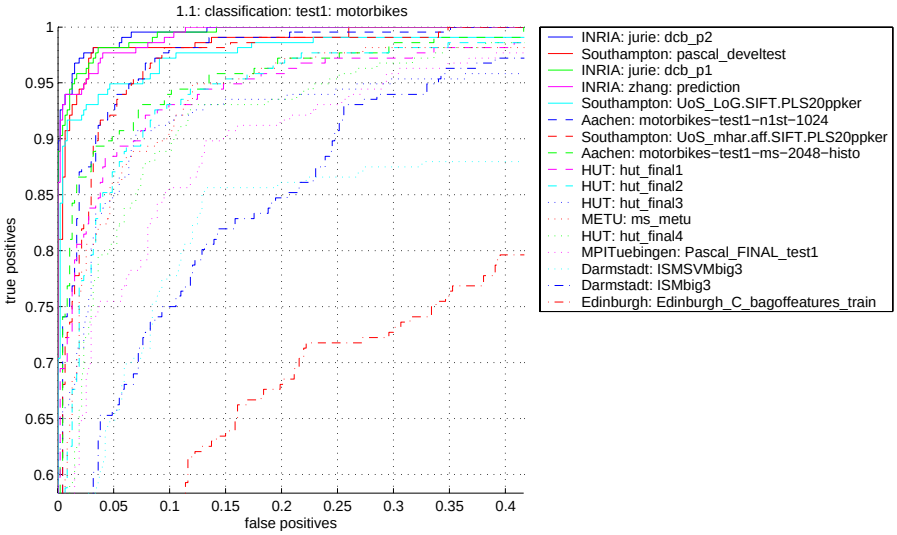

Figure 1: Competition 1.1: test1: motorbikes (all entries)

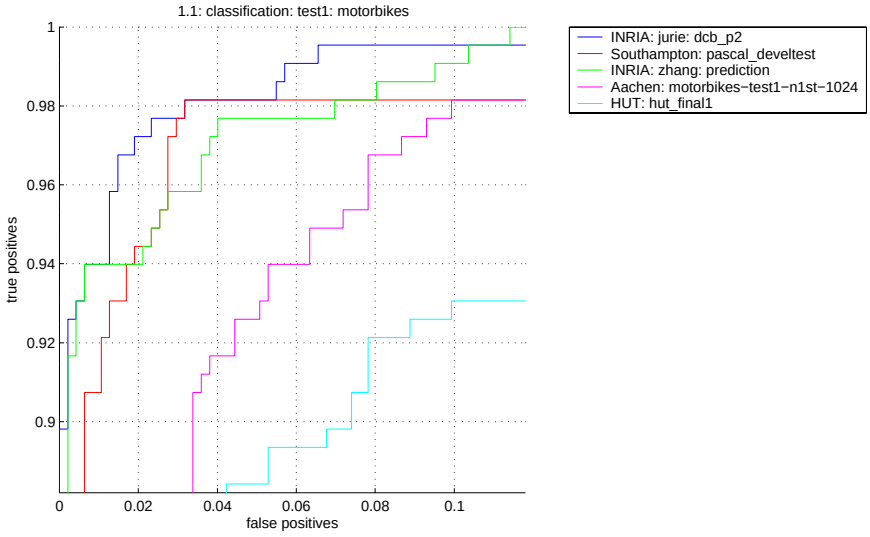

Figure 2: Competition 1.1: test1: motorbikes (top 5 participants by EER)

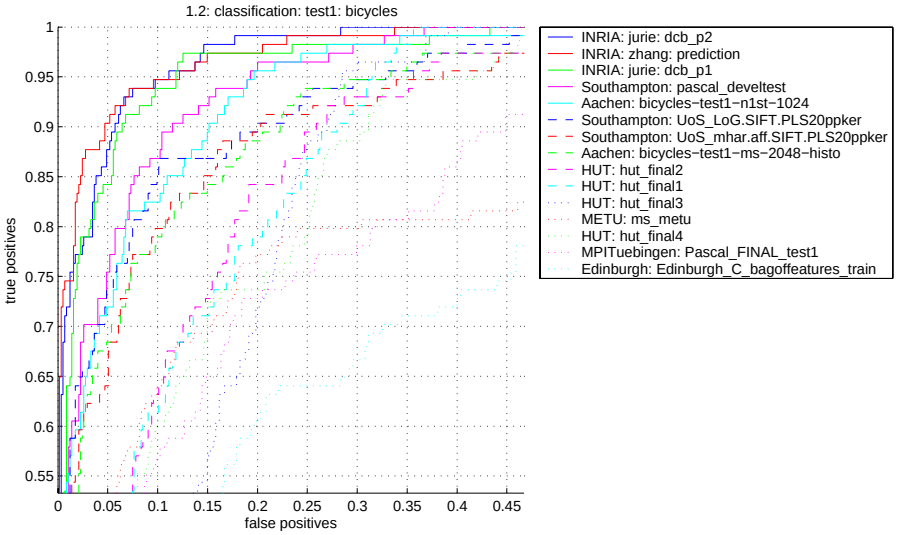

Figure 3: Competition 1.2: test1: bicycles (all entries)

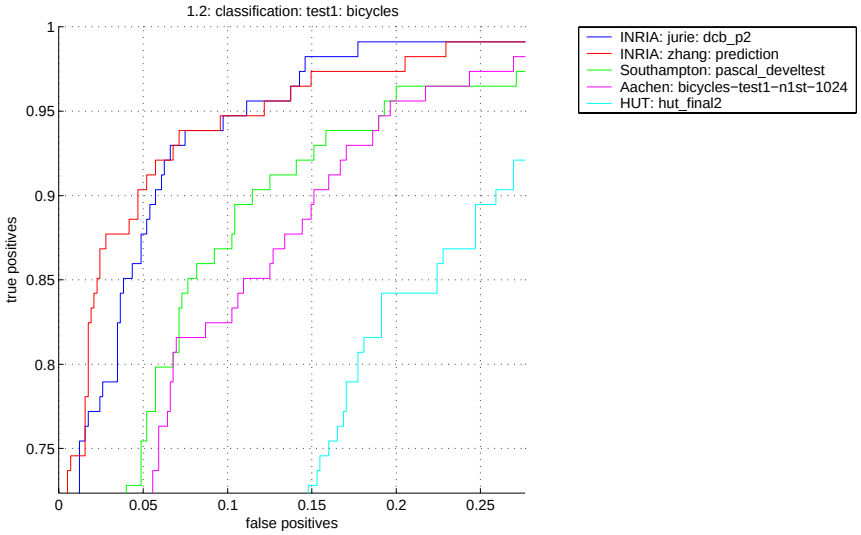

Figure 4: Competition 1.2: test1: bicycles (top 5 participants by EER)

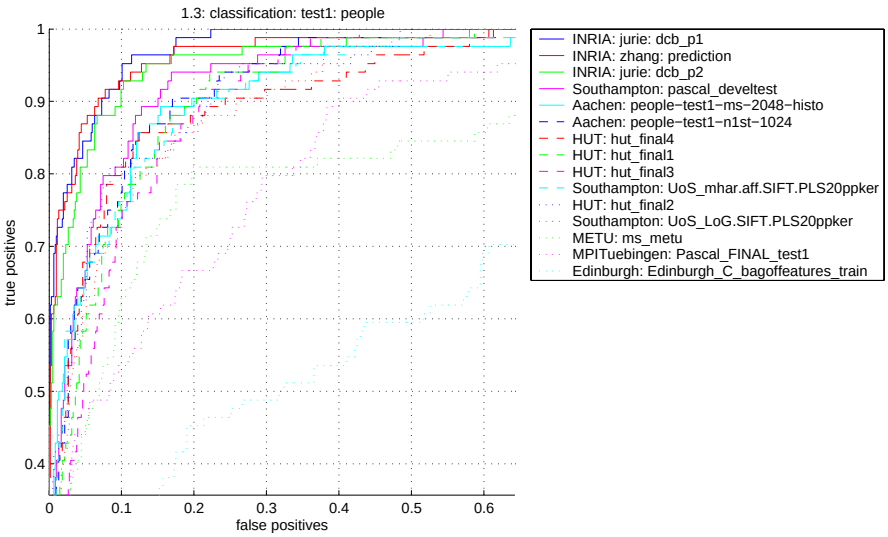

Figure 5: Competition 1.3: test1: people (all entries)

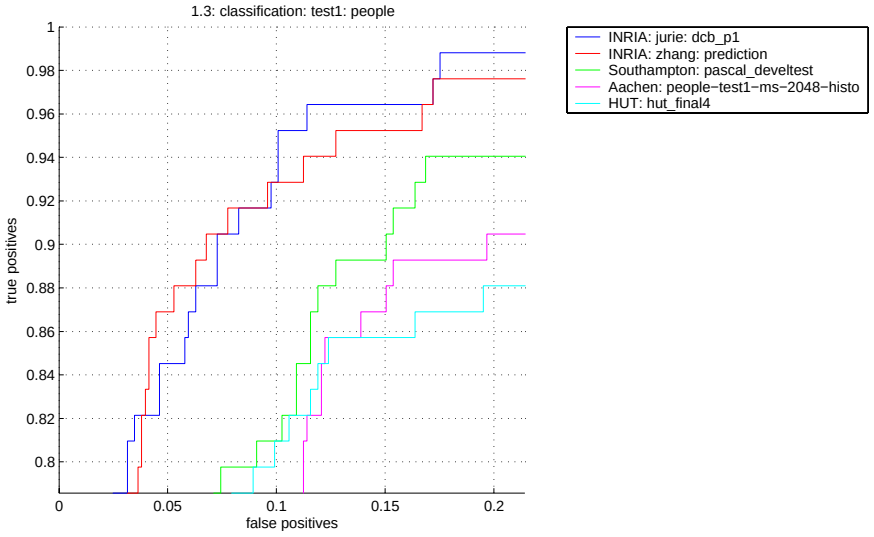

Figure 6: Competition 1.3: test1: people (top 5 participants by EER)

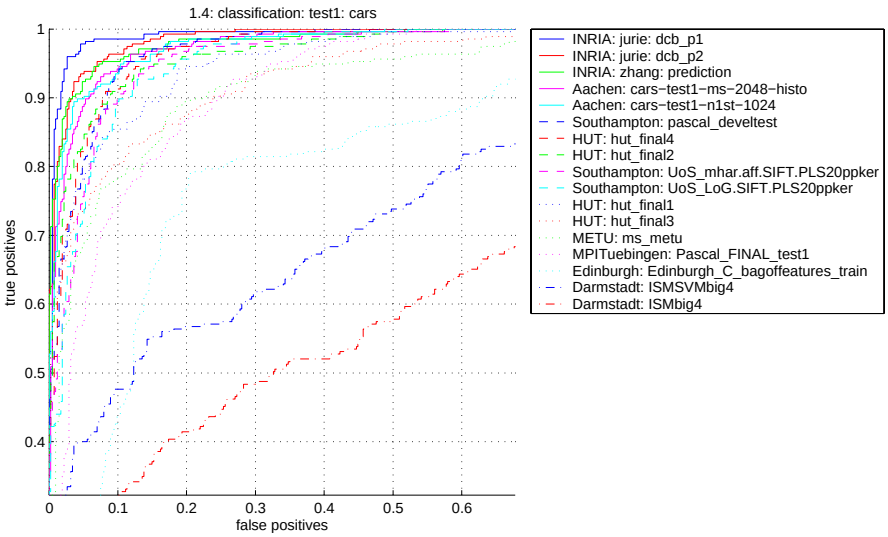

Figure 7: Competition 1.4: test1: cars (all entries)

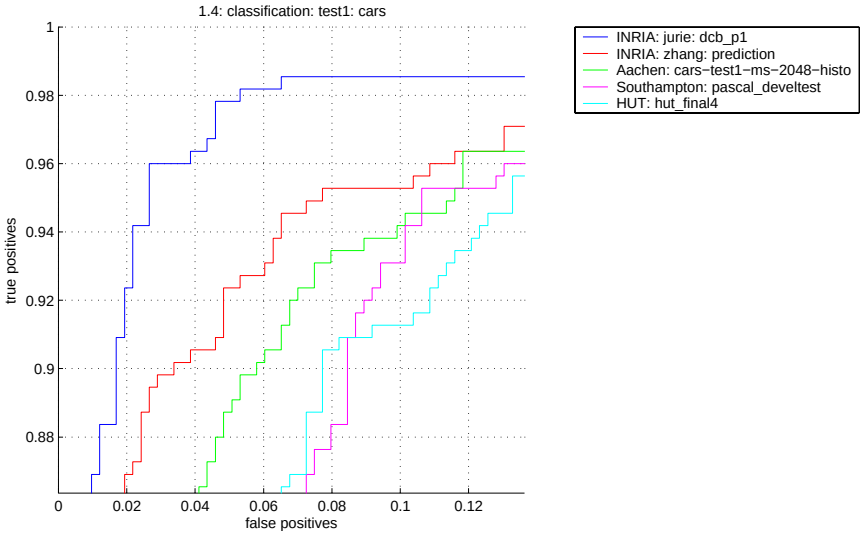

Figure 8: Competition 1.4: test1: cars (top 5 participants by EER)

3.2 Competition 2

- Train on provided data, classify object present/absent in `test2`

6 of the 12 participants took part in this competition. All but one participant (Darmstadt) tackled all four object classes.

Participant	motorbikes	bicycles	people	cars
Aachen	×	×	×	×
Darmstadt	×	–	–	×
Edinburgh	×	×	×	×
FranceTelecom	–	–	–	–
HUT	×	×	×	×
INRIA: dalal	–	–	–	–
INRIA: dorko	–	–	–	–
INRIA: jurie	–	–	–	–
INRIA: zhang	×	×	×	×
METU	–	–	–	–
MPITuebingen	×	×	×	×
Southampton	–	–	–	–

Table 7: Competition 2 participation

Submission	EER	AUC
Aachen: motorbikes-test2-ms-2048-histo	0.767	0.825
Aachen: motorbikes-test2-n1st-1024	0.769	0.829
Darmstadt: ISMbig3	0.663	0.706
Darmstadt: ISMSVMbig3	0.683	0.716
Edinburgh: Edinburgh_C_bagoffeatures_train	0.698	0.710
HUT: hut_final1	0.614	0.666
HUT: hut_final2	0.624	0.693
HUT: hut_final3	0.594	0.637
HUT: hut_final4	0.635	0.675
INRIA: zhang: prediction	0.798*	0.865*
MPITuebingen: Pascal_FINAL_test2	0.698	0.765

Table 8: Competition 2.1: test2: motorbikes

Submission	EER	AUC
Aachen: bicycles-test2-ms-2048-histo	0.667	0.724
Aachen: bicycles-test2-n1st-1024	0.665	0.729
Edinburgh: Edinburgh_C_bagoffeatures_train	0.575	0.606
HUT: hut_final1	0.527	0.567
HUT: hut_final2	0.604	0.647
HUT: hut_final3	0.524	0.546
HUT: hut_final4	0.616	0.645
INRIA: zhang: prediction	0.728*	0.813*
MPITuebingen: Pascal_FINAL_test2	0.616	0.654

Table 9: Competition 2.2: test2: bicycles

Submission	EER	AUC
Aachen: people-test2-ms-2048-histo	0.663	0.721
Aachen: people-test2-n1st-1024	0.669	0.739
Edinburgh: Edinburgh_C_bagoffeatures_train	0.519	0.552
HUT: hut_final1	0.601	0.650
HUT: hut_final2	0.614	0.661
HUT: hut_final3	0.574	0.618
HUT: hut_final4	0.587	0.630
INRIA: zhang: prediction	0.719*	0.798*
MPITuebingen: Pascal_FINAL_test2	0.591	0.655

Table 10: Competition 2.3: test2: people

Submission	EER	AUC
Aachen: cars-test2-ms-2048-histo	0.703	0.767
Aachen: cars-test2-n1st-1024	0.716	0.780
Darmstadt: ISMbig4	0.551	0.572
Darmstadt: ISMSVMbig4	0.658	0.683
Edinburgh: Edinburgh_C_bagoffeatures_train	0.633	0.655
HUT: hut_final1	0.655	0.709
HUT: hut_final2	0.676	0.740
HUT: hut_final3	0.644	0.694
HUT: hut_final4	0.692	0.744
INRIA: zhang: prediction	0.720*	0.802*
MPITuebingen: Pascal_FINAL_test2	0.677	0.717

Table 11: Competition 2.4: test2: cars

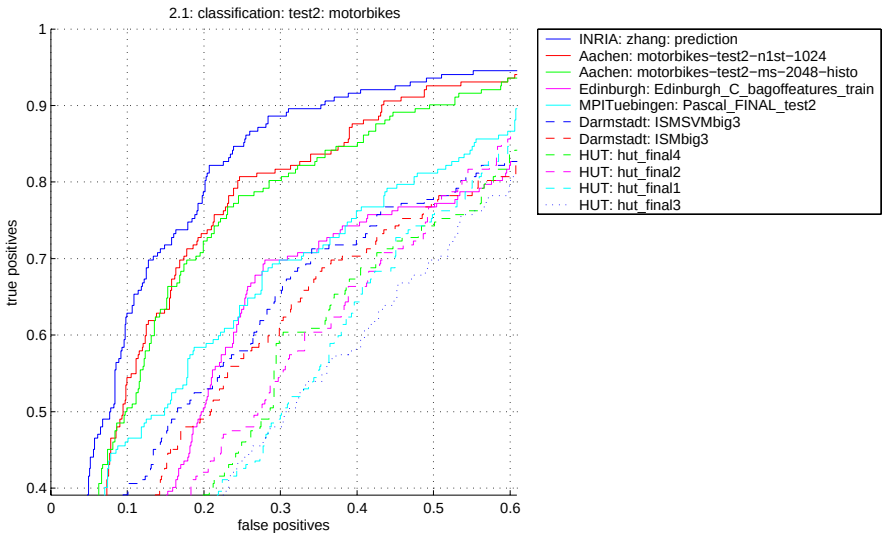

Figure 9: Competition 2.1: test2: motorbikes (all entries)

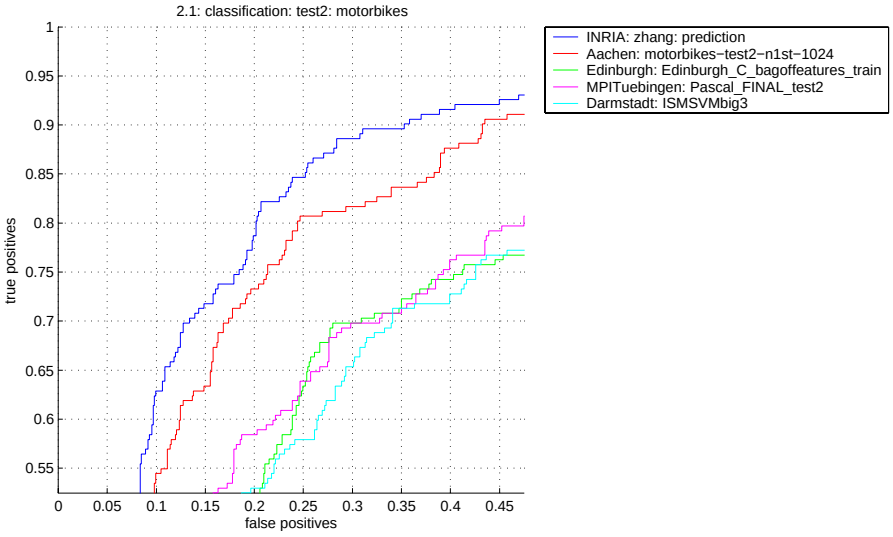

Figure 10: Competition 2.1: test2: motorbikes (top 5 participants by EER)

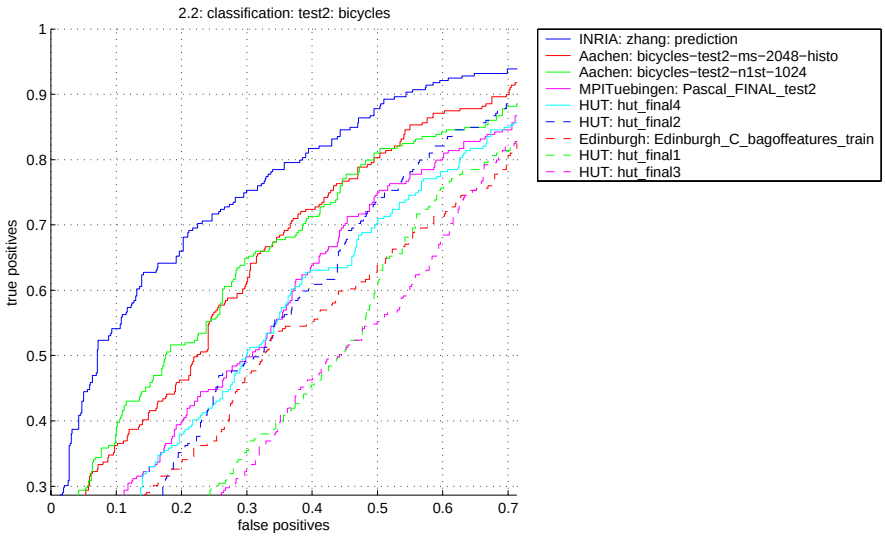

Figure 11: Competition 2.2: test2: bicycles (all entries)

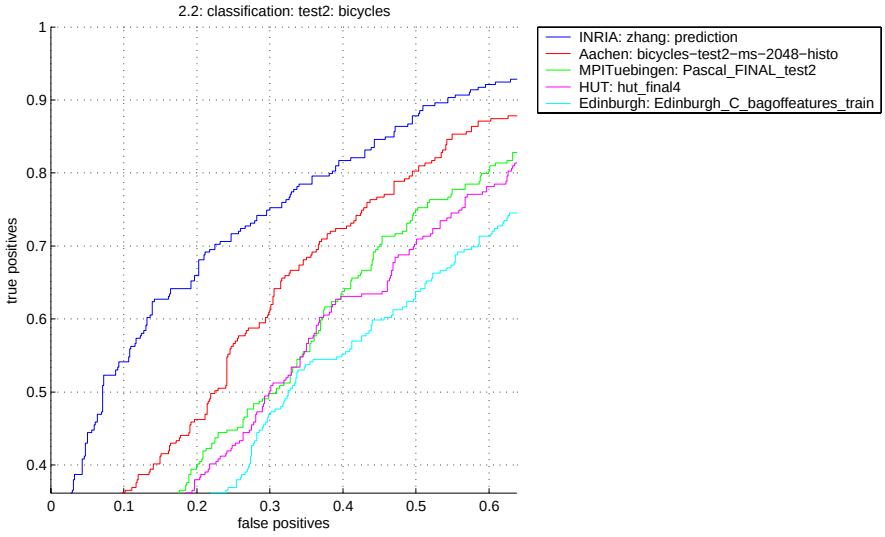

Figure 12: Competition 2.2: test2: bicycles (top 5 participants by EER)

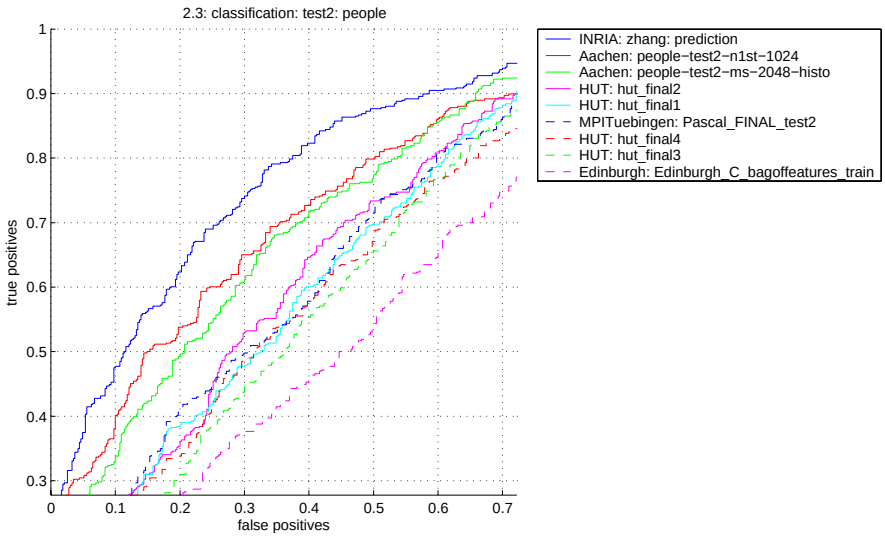

Figure 13: Competition 2.3: test2: people (all entries)

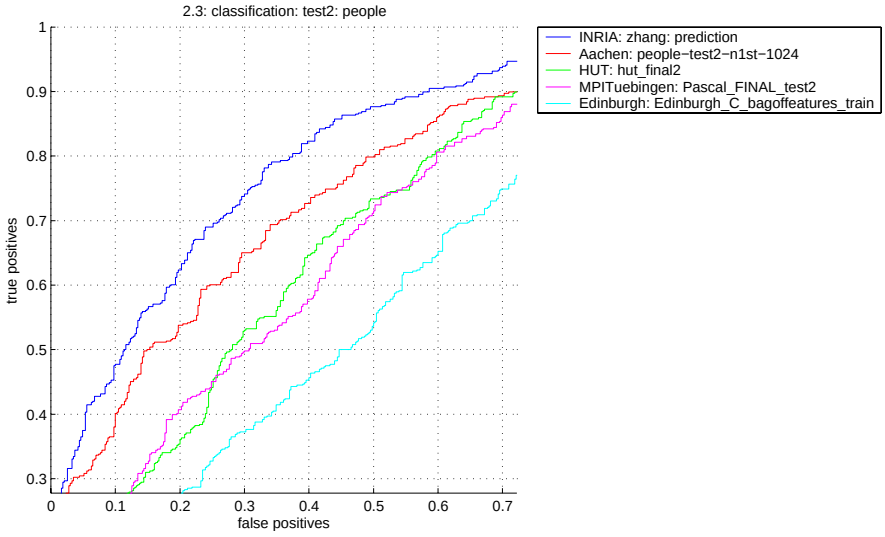

Figure 14: Competition 2.3: test2: people (top 5 participants by EER)

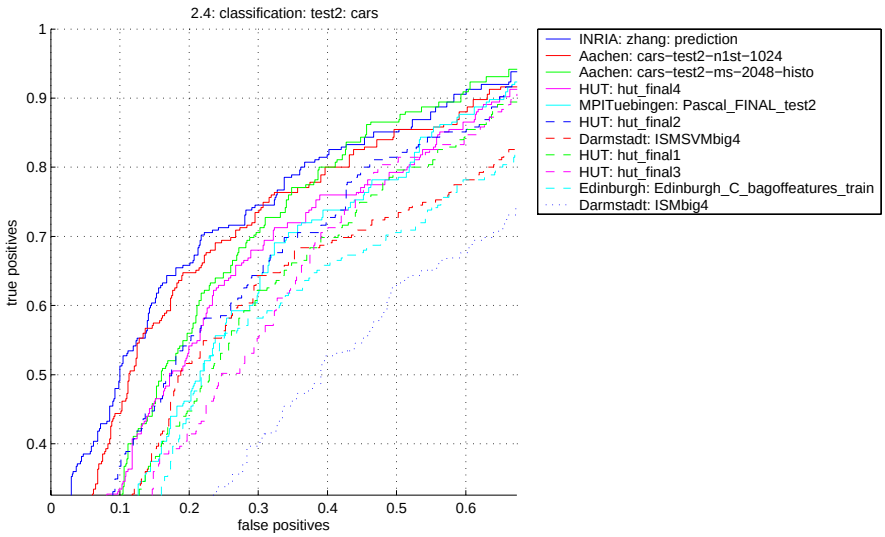

Figure 15: Competition 2.4: test2: cars (all entries)

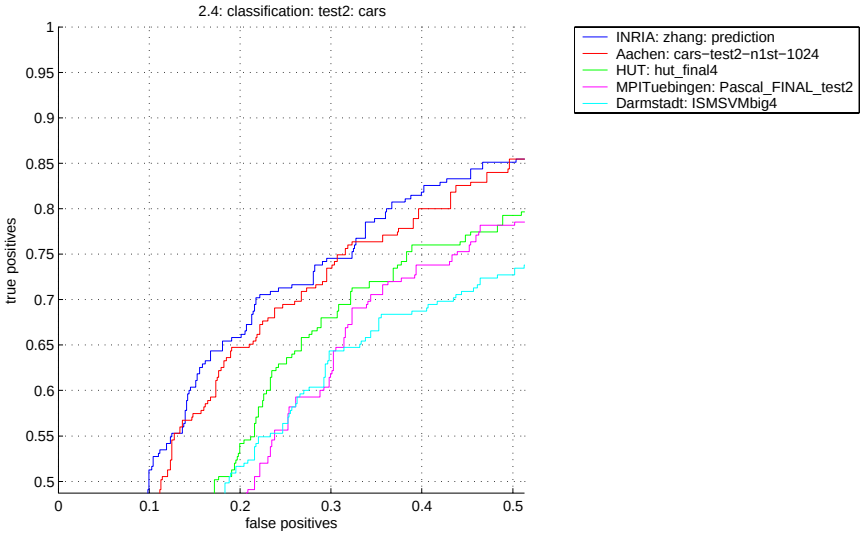

Figure 16: Competition 2.4: test2: cars (top 5 participants by EER)

3.3 Competition 3

- Train on any (non-test) data, classify object present/absent in `test1`

No entries were received for this competition.

3.4 Competition 4

- Train on any (non-test) data, classify object present/absent in `test2`

No entries were received for this competition.

4 Results: Detection

The following sections present the results for the detection competitions. For each competition and object class the 'average precision' (AP) used by TREC is presented, 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.

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 in each competition and for each object class, judged by average precision, is indicated by an asterisk in the tables.

4.1 Competition 5

- Train on provided data, detect object bounding boxes in `test1`

5 of the 12 participants took part in this competition. One participant (Edinburgh) tackled all object classes; others tackled varying subsets.

Participant	motorbikes	bicycles	people	cars
Aachen	–	–	–	–
Darmstadt	×	–	–	×
Edinburgh	×	×	×	×
FranceTelecom	×	–	–	×
HUT	–	–	–	–
INRIA: dalal	×	–	×	×
INRIA: dorko	×	–	×	–
INRIA: jurie	–	–	–	–
INRIA: zhang	–	–	–	–
METU	–	–	–	–
MPITuebingen	–	–	–	–
Southampton	–	–	–	–

Table 12: Competition 5 participation

Submission	AP
Darmstadt: ISMbig3	0.865
Darmstadt: ISMSVMbig3	0.886*
Edinburgh: Edinburgh_D_meanbbox_train	0.216
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.470
Edinburgh: Edinburgh_D_siftbbox_train	0.453
Edinburgh: Edinburgh_D_wholeimage_train	0.118
FranceTelecom: pascal_develtest	0.729
INRIA: dalal: ndalal_competition_number_5	0.490
INRIA: dorko: gydorko	0.598

Table 13: Competition 5.1: test1: motorbikes

Submission	AP
Edinburgh: Edinburgh_D_meanbbox_train	0.007
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.015
Edinburgh: Edinburgh_D_siftbbox_train	0.098
Edinburgh: Edinburgh_D_wholeimage_train	0.119*

Table 14: Competition 5.2: test1: bicycles

Submission	AP
Edinburgh: Edinburgh_D_meanbbox_train	0.000
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.000
Edinburgh: Edinburgh_D_siftbbox_train	0.002
Edinburgh: Edinburgh_D_wholeimage_train	0.000
INRIA: dalal: ndalal_competition_number_5	0.013*
INRIA: dorko: gydorko	0.000

Table 15: Competition 5.3: test1: people

Submission	AP
Darmstadt: ISMbig4	0.468
Darmstadt: ISMSVMbig4_2	0.439
Darmstadt: ISMSVMbig4	0.489
Edinburgh: Edinburgh_D_meanbbox_train	0.000
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.000
Edinburgh: Edinburgh_D_siftbbox_train	0.000
Edinburgh: Edinburgh_D_wholeimage_train	0.000
FranceTelecom: pascal_develtest	0.353
INRIA: dalal: ndalal_competition_number_5	0.613*

Table 16: Competition 5.4: test1: cars

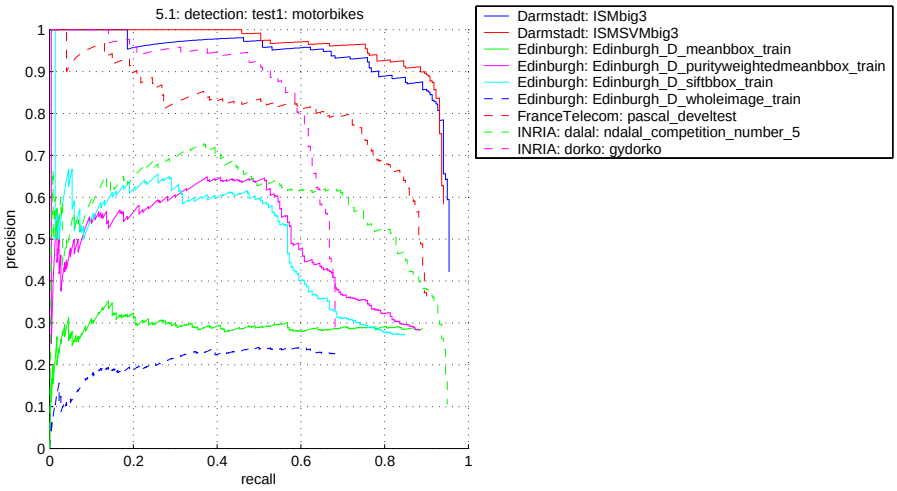

Figure 17: Competition 5.1: test1: motorbikes (all entries)

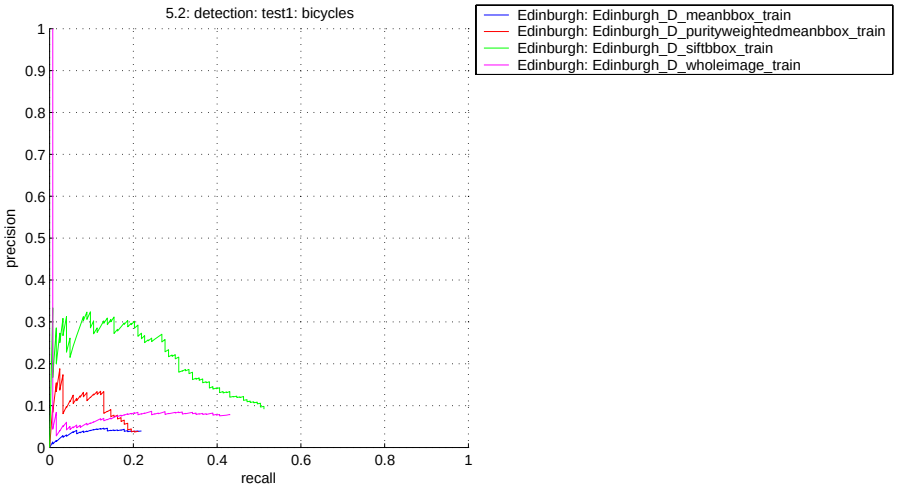

Figure 18: Competition 5.2: test1: bicycles (all entries)

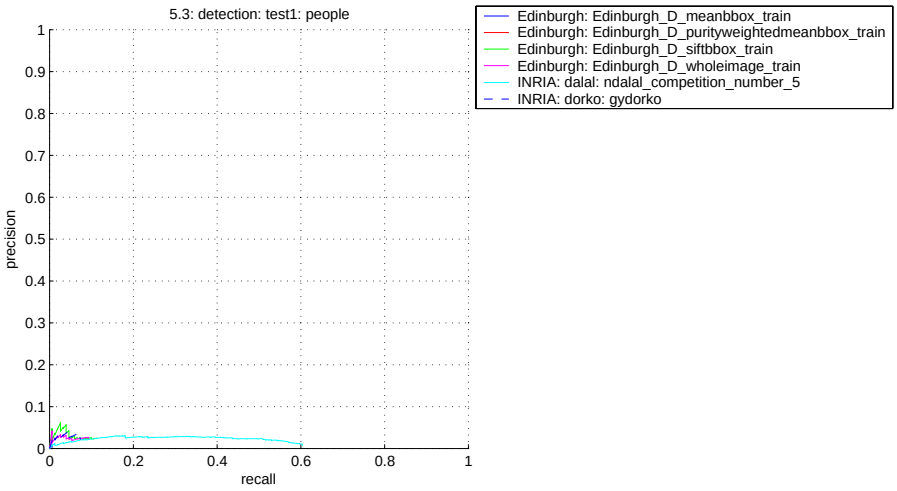

Figure 19: Competition 5.3: test1: people (all entries)

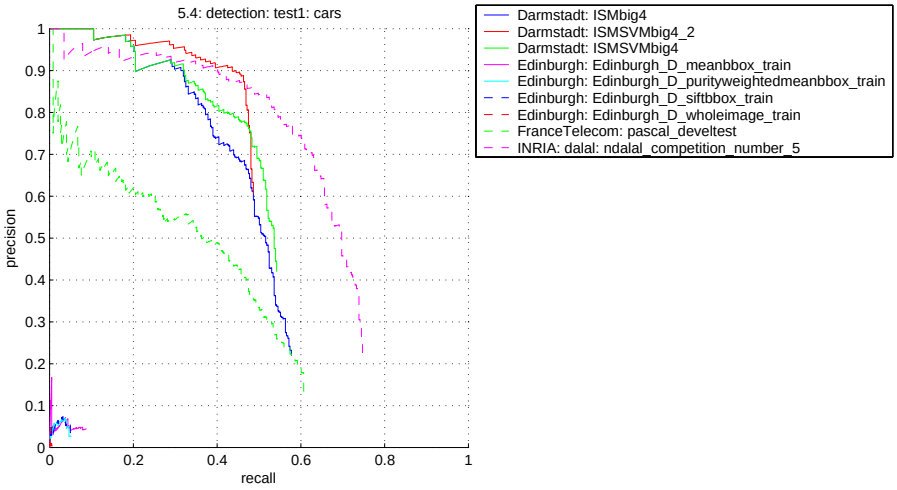

Figure 20: Competition 5.4: test1: cars (all entries)

4.2 Competition 6

- Train on provided data, detect object bounding boxes in `test2`

4 of the 12 participants took part in this competition. One participant (Edinburgh) tackled all object classes; others tackled varying subsets.

Participant	motorbikes	bicycles	people	cars
Aachen	–	–	–	–
Darmstadt	×	–	–	×
Edinburgh	×	×	×	×
FranceTelecom	×	–	–	×
HUT	–	–	–	–
INRIA: dalal	×	–	×	×
INRIA: dorko	–	–	–	–
INRIA: jurie	–	–	–	–
INRIA: zhang	–	–	–	–
METU	–	–	–	–
MPITuebingen	–	–	–	–
Southampton	–	–	–	–

Table 17: Competition 6 participation

Submission	AP
Darmstadt: ISMbig3	0.292
Darmstadt: ISMSVMbig3_2	0.341*
Darmstadt: ISMSVMbig3	0.300
Edinburgh: Edinburgh_D_meanbbox_train	0.055
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.116
Edinburgh: Edinburgh_D_siftbbox_train	0.088
Edinburgh: Edinburgh_D_wholeimage_train	0.020
FranceTelecom: pascal_develtest	0.289
INRIA: dalal: ndalal_competition_number_6	0.124

Table 18: Competition 6.1: test2: motorbikes

Submission	AP
Edinburgh: Edinburgh_D_meanbbox_train	0.000
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.004
Edinburgh: Edinburgh_D_siftbbox_train	0.113*
Edinburgh: Edinburgh_D_wholeimage_train	0.006

Table 19: Competition 6.2: test2: bicycles

Submission	AP
Edinburgh: Edinburgh_D_meanbbox_train	0.000
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.000
Edinburgh: Edinburgh_D_siftbbox_train	0.000
Edinburgh: Edinburgh_D_wholeimage_train	0.000
INRIA: dalal: ndalal_competition_number_6	0.021*

Table 20: Competition 6.3: test2: people

Submission	AP
Darmstadt: ISMbig4	0.083
Darmstadt: ISMSVMbig4	0.181
Edinburgh: Edinburgh_D_meanbbox_train	0.000
Edinburgh: Edinburgh_D_purityweightedmeanbbox_train	0.000
Edinburgh: Edinburgh_D_siftbbox_train	0.028
Edinburgh: Edinburgh_D_wholeimage_train	0.005
FranceTelecom: pascal_develtest	0.106
INRIA: dalal: ndalal_competition_number_6	0.304*

Table 21: Competition 6.4: test2: cars

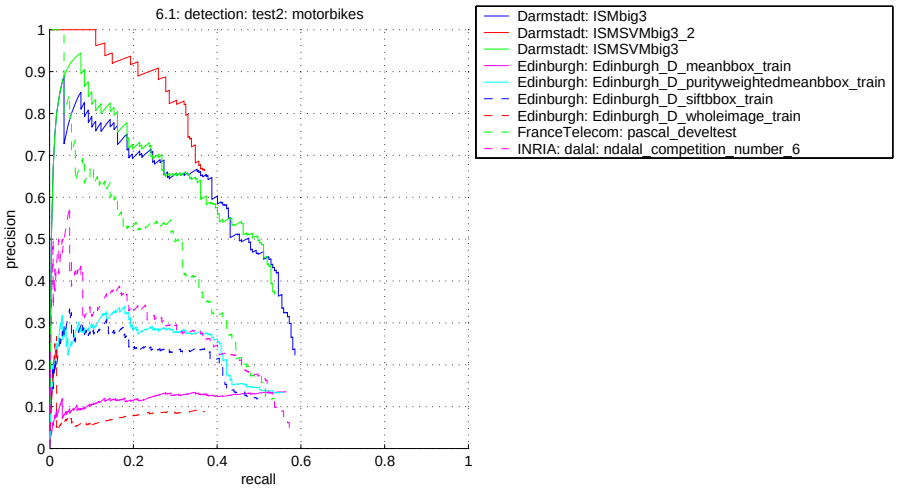

Figure 21: Competition 6.1: test2: motorbikes (all entries)

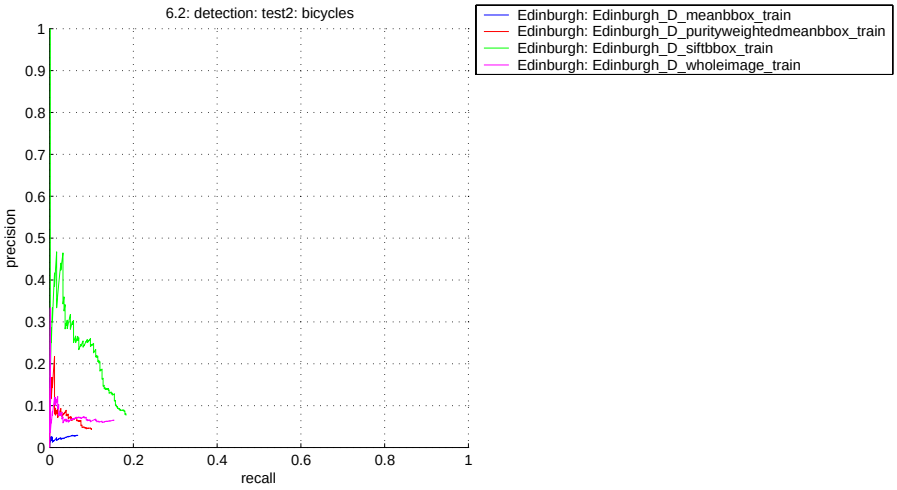

Figure 22: Competition 6.2: test2: bicycles (all entries)

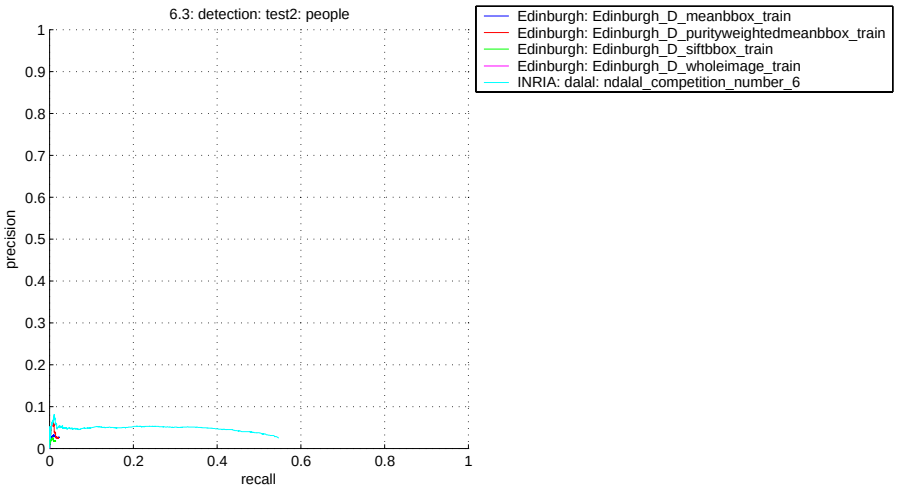

Figure 23: Competition 6.3: test2: people (all entries)

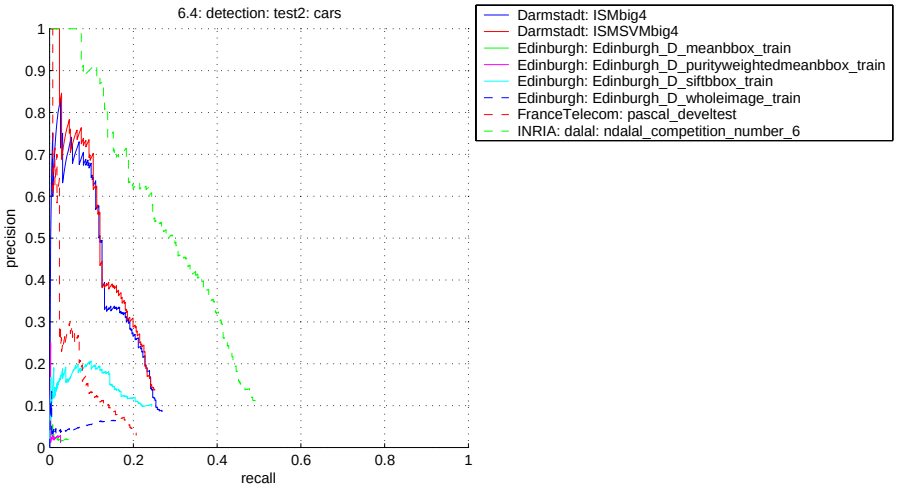

Figure 24: Competition 6.4: test2: cars (all entries)

4.3 Competition 7

- Train on any (non-test) data, detect object bounding boxes in `test1`

A single participant (INRIA: dalal) took part in this competition, tackling one object class (people).

Participant	motorbikes	bicycles	people	cars
Aachen	–	–	–	–
Darmstadt	–	–	–	–
Edinburgh	–	–	–	–
FranceTelecom	–	–	–	–
HUT	–	–	–	–
INRIA: dalal	–	–	×	–
INRIA: dorko	–	–	–	–
INRIA: jurie	–	–	–	–
INRIA: zhang	–	–	–	–
METU	–	–	–	–
MPITuebingen	–	–	–	–
Southampton	–	–	–	–

Table 22: Competition 7 participation

Submission	AP
INRIA: dalal: ndalal_competition_number_7	0.416*

Table 23: Competition 7.3: test1: people

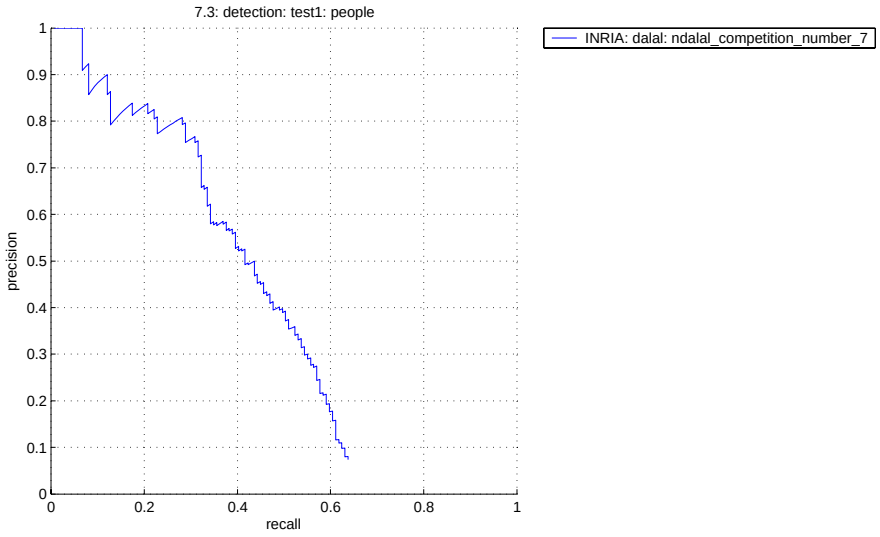

Figure 25: Competition 7.3: test1: people (all entries)

4.4 Competition 8

- Train on any (non-test) data, detect object bounding boxes in `test2`

A single participant (INRIA: dalal) took part in this competition, tackling one object class (people).

Participant	motorbikes	bicycles	people	cars
Aachen	–	–	–	–
Darmstadt	–	–	–	–
Edinburgh	–	–	–	–
FranceTelecom	–	–	–	–
HUT	–	–	–	–
INRIA: dalal	–	–	×	–
INRIA: dorko	–	–	–	–
INRIA: jurie	–	–	–	–
INRIA: zhang	–	–	–	–
METU	–	–	–	–
MPITuebingen	–	–	–	–
Southampton	–	–	–	–

Table 24: Competition 8 participation

Submission	AP
INRIA: dalal: ndalal_competition_number_8	0.438*

Table 25: Competition 8.3: test2: people

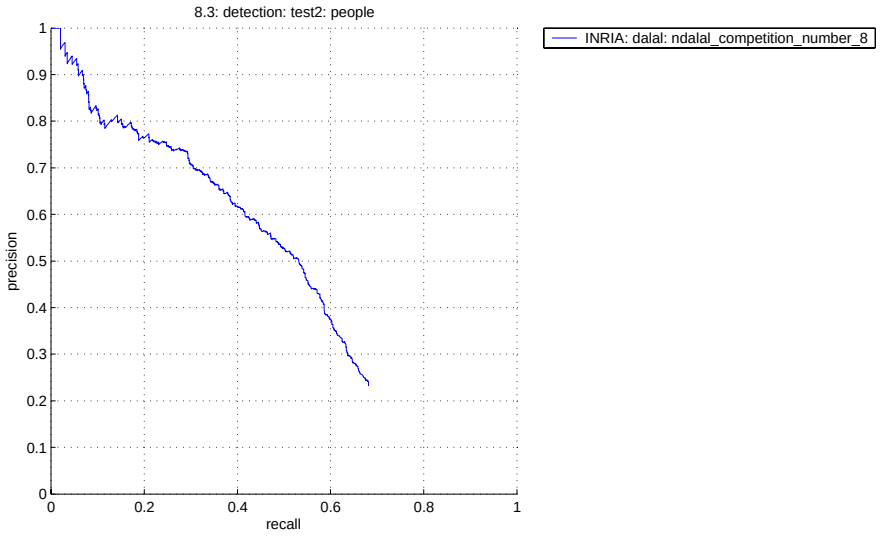

Figure 26: Competition 8.3: test2: people (all entries)

5 Acknowledgements

We are very grateful to those that provided images and their annotations; these include: Bastian Leibe & Bernt Schiele(Darmstadt University of Technology), Shivani Agarwal, Aatif Awan & Dan Roth (University of Illinois at Urbana-Champaign), Rob Fergus & Pietro Perona (California Institute of Technology), Antonio Torralba, Kevin P. Murphy & William T. Freeman (Massachusetts Institute of Technology), Andreas Opelt & Axel Pinz (Graz University of Technology), Navneet Dalal & Bill Triggs (INRIA), Michalis Titsias (University of Edinburgh), and Hao Shao (ETH Zurich). The original PASCAL Object Recognition site `http://www.pascal-network.org/challenges/VOC/` was assembled by Manik Varma (University of Oxford). We are also grateful to Steve Gunn (University of Southampton) for enabling the web page, Rebecca Hoath (University of Oxford) for help assembling the challenge database, and to Kevin Murphy for spotting several glitches in the original development kit.

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.