The 2005 PASCAL Visual Object Classes Challenge

Mark Everingham[1], Andrew Zisserman[1], Christopher K. I. Williams[2], Luc Van Gool[3], Moray Allan[2], Christopher M. Bishop[10], Olivier Chapelle[11], Navneet Dalal[8], Thomas Deselaers[4], Gyuri Dorkó[8], Stefan Duffner[6], Jan Eichhorn[11], Jason D. R. Farquhar[12], Mario Fritz[5], Christophe Garcia[6], Tom Griffiths[2], Frederic Jurie[8], Thomas Keysers[4], Markus Koskela[7], Jorma Laaksonen[7], Diane Larlus[8], Bastian Leibe[5], Hongying Meng[12], Hermann Ney[4], Bernt Schiele[5], Cordelia Schmid[8], Edgar Seemann[5], John Shawe-Taylor[12] Amos Storkey[2], Sandor Szedmak[12], Bill Triggs[8], Ilkay Ulusoy[9], Ville Viitaniemi[7], and Jianguo Zhang[8]

[1] University of Oxford, Oxford, UK.
[2] University of Edinburgh, Edinburgh, UK.
[3] ETH Zentrum, Zurich, Switzerland.
[4] RWTH Aachen University, Aachen, Germany.
[5] TU-Darmstadt, Darmstadt, Germany.
[6] France Télécom, Cesson Sévigné, France.
[7] Helsinki University of Technology, Helsinki, Finland.
[8] INRIA Rhône-Alpes, Montbonnot, France.
[9] Middle East Technical University, Ankara, Turkey.
[10] Microsoft Research, Cambridge, UK.
[11] Max Planck Institute for Biological Cybernetics, Tübingen, Germany.
[12] University of Southampton, Southampton, UK.

Abstract. The PASCAL Visual Object Classes Challenge ran from February to March 2005. The goal of the challenge was to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects). Four object classes were selected: motorbikes, bicycles, cars and people. Twelve teams entered the challenge. In this chapter we provide details of the datasets, algorithms used by the teams, evaluation criteria, and results achieved.

1 Introduction

In recent years there has been a rapid growth in research, and quite some success, in visual recognition of object classes; examples include [1, 5, 10, 14, 18, 28, 39, 43]. Many of these papers have used the same image datasets as [18] in order to compare their performance. The datasets are the so-called 'Caltech 5' (faces, airplanes, motorbikes, cars rear, spotted cats) and UIUC car side images of [1]. The problem is that methods are now achieving such good performance that they have effectively saturated on these datasets, and thus the datasets are failing to challenge the next generation of algorithms. Such saturation can arise because

the images used do not explore the full range of variability of the imaged visual class. Some dimensions of variability include: clean vs. cluttered background; stereotypical views vs. multiple views (e.g. side views of cars vs. cars from all angles); degree of scale change, amount of occlusion; the presence of multiple objects (of one or multiple classes) in the images.

Given this problem of saturation of performance, the Visual Object Classes Challenge was designed to be more challenging by enhancing some of the dimensions of variability listed above compared to the databases that had been available previously, so as to explore the failure modes of different algorithms.

The PASCAL[13] Visual Object Classes (VOC) Challenge ran from February to March 2005. A development kit of training and validation data, baseline algorithms, plus evaluation software was made available on 21 February, and the test data was released on 14 March. The deadline for submission of results was 31 March, and a challenge workshop was held in Southampton (UK) on 11 April 2005. Twelve teams entered the challenge and six presented their findings at the workshop. The development kit and test images can be found at the website `http://www.pascal-network.org/challenges/VOC/`.

The structure of the remainder of the chapter is as follows. Section 2 describes the various competitions defined for the challenge. Section 3 describes the datasets provided to participants in the challenge for training and testing. Section 4 defines the *classification* competitions of the challenge and the method of evaluation, and discusses the types of method participants used for classification. Section 5 defines the *detection* competitions of the challenge and the method of evaluation, and discusses the types of method participants used for detection. Section 6 presents descriptions of the methods provided by participants. Section 7 presents the results of the classification competitions, and Section 8 the results for the detection competitions. Section 9 concludes the chapter with discussion of the challenge results, aspects of the challenge raised by participants in the challenge workshop, and prospects for future challenges.

2 Challenge

The goal of the challenge was to recognize objects from a number of visual object classes in realistic scenes. Four object classes were selected, namely motorbikes, bicycles, cars, and people. There were two main competitions:

1. Classification: For each of the four classes, predicting the presence/absence of an example of that class in the test image.
2. Detection: Predicting the bounding box and label of each object from the 4 target classes in the test image.

Contestants were permitted to enter either or both of the competitions, and to tackle any or all of the four object classes. The challenge further divided the

[13] PASCAL stands for pattern analysis, statistical modelling and computational learning. It is the name of an EU Network of Excellence funded under the IST Programme of the European Union.

competitions according to what data was used by the participants for training their systems:

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

The intention in the first case was to establish just what level of success could currently be achieved on these problems, and by what method. Participants were free to use their own databases of training images which might be much larger than those provided for the challenge, additional annotation of the images such as object parts or reference points, 3D models, etc. Such resources should potentially improve results over using a smaller fixed training set.

In the second case, the intention was to establish which methods were most successful given a specified training set of limited size. This was to allow judgement of which methods generalize best given limited data, and thus might scale better to the problem of recognizing a large number of classes, for which the collection of large data sets becomes an onerous task.

3 Image Sets

Two distinct sets of images were provided to participants: a first set containing images both for training and testing, and a second set containing only images for testing.

3.1 First Image Set

The first image set was divided into several subsets:

`train`: Training data

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

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

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

In the preliminary phase of the challenge, the `train` and `val` image sets were released with the development kit. This gave participants the opportunity to try out the code provided in the development kit, including baseline implementations of the classification and detection tasks, and code for evaluating results. The baseline implementations provided used the `train` set for training, and demonstrated use of the evaluation functions on the `val` set. For the challenge proper, the `test1` set was released for evaluating results, to be used for testing

Table 1. Statistics of the first image set. The number of images (containing at least one object of the corresponding class) and number of object instances are shown.

	`train`		`val`		`train+val`		`test1`	
	images	objects	images	objects	images	objects	images	objects
motorbikes	107	109	107	108	214	217	216	220
bicycles	57	63	57	60	114	123	113	123
people	42	81	42	71	84	152	84	149
cars	136	159	136	161	272	320	275	341

alone. Participants were free to use any subset of the `train` and `val` sets for training. Table 1 lists statistics for the first image set.

Examples of images from the first image set containing instances of each object class are shown in Figure 1. Images were taken from the PASCAL image database collection; these were provided by Bastian Leibe & Bernt Schiele (TU-Darmstadt), 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), and Navneet Dalal & Bill Triggs (INRIA).

The images used in the challenge were manually selected to remove duplicate images, and very similar images taken from video sequences. Subjective judgement of which objects are "recognizable" was made and images containing annotated objects which were deemed unrecognizable were discarded. The subjective judgement required that the object size (in pixels) was sufficiently large, and that the object could be recognized in isolation without the need for "excessive" contextual reasoning e.g. "this blob in the distance must be a car because it is on a road." Images where the annotation was ambiguous were also discarded, for example images of many bicycles in a bike rack for which correct segmentation of the image into individual objects proves impossible even for a human observer.

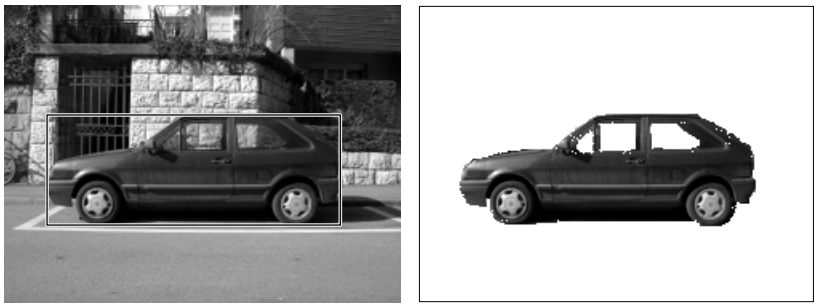
The images contain objects at a variety of scales and in varying context. Many images feature the object of interest in a "dominant" position, i.e. in the centre of the image, occupying a large area of the image, and against a fairly uniform background. The pose variation in this image set is somewhat limited, for example most motorbikes appear in a "side" view, and most cars in either "side" or "front" views (Figure 1). Pose for the bicycles and people classes is somewhat more variable. Most instances of the objects appear un-occluded in the image, though there are some examples, particularly for people (Figure 1) where only part of the object is visible.

Annotation All the images used in the first image set had already been annotated by contributors of the data to the PASCAL image databases collection. The annotation was not changed for the challenge beyond discarding images

Fig. 1. Example images from the first image set. From top to bottom: motorbikes, bicycles, people, and cars. The original images are in colour.

for which the annotation was considered incomplete, ambiguous, or erroneous. For each object of interest (e.g. cars), the annotation provides a bounding box (Figure 2a); for some object instances additional annotation is available in the form of a segmentation mask (Figure 2b) specifying which pixels are part of the object.

Each object is labelled with one of the object classes used in the challenge: motorbikes, bicycles, people or cars; in addition, the original PASCAL object class labels were included in the annotation. For some object instances these specify a more detailed label, typically corresponding to a pose of the object e.g. `PAScarSide` and `PAScarRear` respectively identify side and rear views of a car. Participants were free to use this information, for example the group from TU-Darmstadt chose to only train on side views (Section 6.2).



(a) Bounding box (b) Segmentation mask

Fig. 2. Annotation of objects available for training. (a) all objects are annotated with their bounding boxes. (b) some objects additionally have a pixel segmentation mask.

3.2 Second Test Set

In the first image set, images from the original pool of data were assigned randomly to training sets (`train+val`) and test set (`test1`). This follows standard practice in the machine learning field in which training and test data are assumed to be drawn from the same distribution. To enable a more difficult set of competitions a second test set (`test2`) was also prepared, intended to give a distribution of images with more variability than the training data. This image set was collected from Google Images specifically for the challenge. Example images from `test2` are shown in Figure 3. The image set is less homogenous than the first image set due to the wide range of different sources from which the images were taken. Some images resembling the composition of those in the first image set were selected, but also images containing greater variation in scale, pose, and level of occlusion. Table 2 lists statistics for the `test2` image set.

Table 2. Statistics of the `test2` image set. The number of images (containing at least one object of the corresponding class) and number of object instances are shown.

	`test2`	
	images	objects
motorbikes	202	227
bicycles	279	399
people	526	1038
cars	275	381

Fig. 3. Example images from the `test2` test set. From top to bottom: motorbikes, bicycles, people, and cars. The original images are in colour. There is greater variability in scale and pose, and more occlusion than the images of `test1` shown in Figure 1.

3.3 Negative Examples

For both training and testing it is necessary to have a pool of *negative* images not containing objects of a particular class. Some other work has used a fixed negative image set of generic "background" images for testing; this risks oversimplifying the task, for example finding images of cars might reasonably be achieved by finding images of roads; if however the negative image set contains many images of roads with *no* cars, the difficulty of the task is made more realistic.

The challenge treated both the classification and detection tasks as a set of *binary* classification/detection problems (Sections 4, 5) e.g. car vs. non-car, and made use of images containing *other* object classes as the negative examples. For example in the *car* detection task, images containing motorbikes (but no cars) were among the negative examples; in the *motorbike* detection task, images

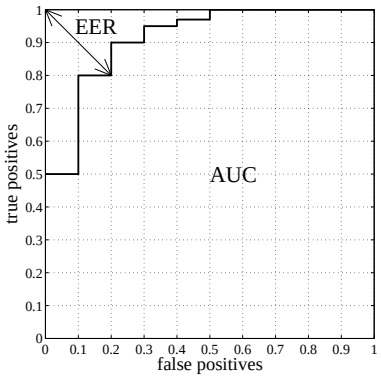


Fig. 4. Example Receiver Operating Characteristic (ROC) curve for the *classification* task. The two quantitative measures of performance are illustrated: the Equal Error Rate (EER) and Area Under Curve (AUC).

containing cars (but no motorbikes) became negative examples. Because the contexts in which the four object classes appear might be considered similar, e.g. cars, motorbikes, bicycles and people may all appear in a street scene, re-use of the images in this way should make for a more realistic (and harder) task.

4 Classification Task

The goal in the classification task is to determine whether a given image contains at least one instance of a particular object class. The task was treated as four independent *binary* classification tasks i.e. "does this image contain an object of type x?" where x was either motorbike, bicycle, people or cars. Treating the task in this way enables the use of the well-established Receiver Operating Characteristic (ROC) for examining results. Other work has also considered the "forced choice" scenario i.e. "is this an image of a motorbike, a bicycle, a person, or a car?"; this scenario is inapplicable in the context of the challenge since a single image may contain instances of objects from more than one class.

4.1 Evaluation of Results

Evaluation of results was performed using ROC curve analysis. This required that participants' methods output a "confidence" for an image, with large values indicating high confidence that the object class of interest is present. Figure 4 shows an example ROC curve, obtained by applying a set of thresholds to the confidence output by a method. On the x-axis is plotted the proportion of false positives (how many times a method says the object class is present when it is not); on the y-axis is plotted the proportion of true positives (how many times

a method says the object class is present when it is). The ROC curve makes it easy to observe the trade-off between the two; some methods may recognize some small proportion of objects very accurately but fail to recognize many, where others may give more balanced performance.

A definitive measure for quantitative evaluation of ROC curves is not possible since, depending on the application, one might wish to place different emphasis on the accuracy of a method at low or high false positive rates. The challenge used two measures to avoid bias: (i) the Equal Error Rate (EER) measures the accuracy at which the number of false positives and false negatives are equal. This measure somewhat emphasizes the behaviour of a method at low false positive rates which might be reasonable for a real-world application; (ii) the Area Under Curve (AUC) measures the total area under the ROC curve. This measure penalizes failures across the whole range of false positives, e.g. a method which recognizes some large proportion of instance with zero error but fails on the remaining portion of the data. In practice, in the experiments, the method judged "best" by each of the two measures was typically the same.

4.2 Competitions and Participation

Four competitions were defined for the classification task, by the choice of training data: provided for the challenge, or the participant's own data; and the test set used: the "easier" `test1` images, or the "harder" `test2` images. Table 3 summarizes the competitions. For each competition, performance on each of the four object classes was evaluated. Participants were free to submit results for any or all of the object classes.

Table 3. Competitions for the *classification* task, defined by the choice of training data and test data.

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`

Table 4 lists the participation in competitions 1 and 2, which used the provided `train+val` image set for training. Nine of the twelve participants entered results for these competitions. All but one tackled all object classes (see Section 4.3). Half the participants submitted results for both test sets. No results were submitted for competitions 3 and 4, in which data other than the provided `train+val` image set could be used.

Table 4. Participation in *classification* competitions 1 and 2 which used the provided `train+val` image set for training. Bullets indicate participation in the competition for a particular test set and object class.

	`test1`				`test2`			
	motorbikes	bicycles	people	cars	motorbikes	bicycles	people	cars
Aachen	•	•	•	•	•	•	•	•
Darmstadt	•	–	–	•	•	–	–	•
Edinburgh	•	•	•	•	•	•	•	•
FranceTelecom	–	–	–	–	–	–	–	–
HUT	•	•	•	•	•	•	•	•
INRIA-Dalal	–	–	–	–	–	–	–	–
INRIA-Dorko	–	–	–	–	–	–	–	–
INRIA-Jurie	•	•	•	•	–	–	–	–
INRIA-Zhang	•	•	•	•	•	•	•	•
METU	•	•	•	•	–	–	–	–
MPITuebingen	•	•	•	•	•	•	•	•
Southampton	•	•	•	•	–	–	–	–

4.3 Overview of Classification Methods

Section 6 gives full details of the methods used by participants. The approaches used for the classification task can be broadly divided into four categories:

Distributions of Local Image Features. Most participants took the approach of capturing the image content as a distribution over local image features. In these methods a set of vector-valued descriptors capturing local image content is extracted from the image, typically around "interest" points; the image is represented by some form of probability distribution over the set of descriptors. Recognition is carried out by training a classifier to distinguish the distributions for a particular class.

All participants in this category used the SIFT descriptor [32] to represent the appearance of local image regions.

All but one participant (INRIA-Jurie) used "interest point" detection algorithms to define points about which local descriptors were extracted, including the Harris and LoG detectors. Aachen additionally extract descriptors around points on a fixed coarse grid; INRIA-Jurie extracted descriptors around points on a dense grid at multiple scales.

Four participants: Aachen, Edinburgh, INRIA-Jurie, and INRIA-Zhang used a "bag of words" representation. In these methods, local descriptors are assigned a discrete "visual word" from a dictionary obtained by clustering. The image representation is then a histogram over the dictionary, recording either the presence of each word, or the number of times each word occurs in the image.

Two participants MPITuebingen and Southampton used an alternative method based on defining a kernel between sets of extracted features. Both

participants used the Bhattacharyya kernel; for Southampton this was defined by a Gaussian distribution in SIFT feature space, while MPITuebingen used a "minor kernel" to lift the calculation into a kernel feature space.

All but two participants in this category used a support vector machine (SVM) classifier. Aachen used a log-linear model trained by iterative scaling; Edinburgh used a functionally equivalent model trained by logistic regression.

Recognition of Individual Local Features. METU proposed a method also employing interest point detection and extraction of local features; the SIFT descriptor and colour features were used. In the METU method, rather than examining the entire distribution of local descriptors for an image, a model is learnt which assigns a class probability to *each* local feature; a class is assigned to the image by a noisy-or operation on the class probabilities for each local feature in the image.

Recognition based on Segmented Regions. HUT proposed a method combining features extracted both from the entire image and from regions obtained by an image segmentation algorithm; features included colour, shape and texture descriptors. A number of Self Organizing Maps (SOMs) defined on the different feature spaces were used to classify descriptors obtained from the segmented regions and the whole image, and these results were combined to produce an overall classification.

Classification by Detection. Darmstadt adopted the approach of "classification by detection" in which a *detector* for the class of object is applied to the image and the image assigned to the object class if a sufficiently confident detection is found. The method is described more fully in Section 5.3. This approach is of particular interest since it is able to show "why" the object class is assigned to the image, by highlighting the image area thought to be an instance of the object class.

4.4 Discussion of Classification Methods

Most participants used "global" methods in which a descriptor of the overall image content is extracted; this leaves the task of deciding which elements of the descriptor are relevant to the object of interest to the classifier. All of these participants used only the class label attached to an image for training, ignoring additional annotation such as the bounding boxes of objects in the image.

One possible advantage of "global" methods is that the image description captures information not only about the object of interest e.g. a car, but also it's context e.g. the road. This contextual information might prove useful in recognizing some object classes; however, the risk is that the system may fail to distinguish the object from the context and thus show poor generalization to other environments, for example recognizing a car in a street vs. in a field.

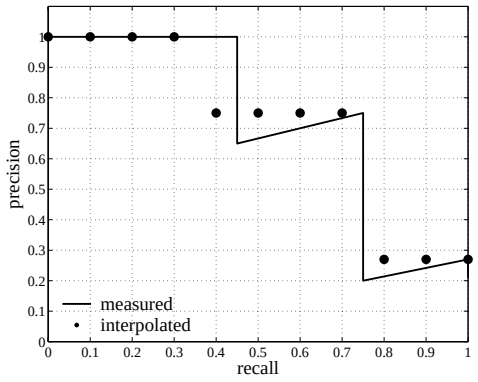


Fig. 5. Example Precision/Recall (PR) curve for the *detection* task. The solid line denotes measured performance (perfect precision at zero recall is assumed). The dots indicate the corresponding interpolated precision values used in the average precision (AP) measure.

The approach used by METU uses very *local* information: the classification may be based on a *single* local feature in the image; interestingly, the learning method used here ignores the bounding box information provided. HUT combined global and more local information by computing feature descriptors from both the whole image and segmented regions.

Darmstadt's "classification by detection" approach explicitly ignores all but the object, using bounding boxes or segmentation masks for training, and looking at local evidence for testing; this ensures that the method is modelling the object class of interest rather than statistical regularities in the image background, but may also fail to take advantage of contextual information.

The Darmstadt method is able to give a visual explanation of *why* an image has been classified as containing an object of interest, since it outputs bounding boxes for each object. For some of the other methods (Aachen, Edinburgh, METU, HUT) it might be possible to obtain some kind of labelling of the objects in the image by back-projecting highly-weighted features into the image.

Only two participants explicitly incorporated any geometric information: HUT included shape descriptors of segmented regions in their image representation, and the Darmstadt method uses both local appearance of object parts and their geometric relations. In the global methods, geometric information such as the positions of object parts might be implicitly encoded, but is not transparently represented.

5 Detection Task

The goal in the detection task is to detect and localize any instances of a particular object class in an image. Localization was defined as specifying a 'bounding

box' rectangle enclosing each object instance in the image. One detection task was run for each class: motorbikes, bicycles, people, and cars.

5.1 Evaluation of Results

Evaluation of results was performed using Precsion/Recall (PR) curve analysis. The output required from participants' methods was a set of bounding boxes with corresponding "confidence" values, with large values indicating high confidence that the detection corresponds to an instance of the object class of interest. Figure 5 shows an example PR curve, obtained by applying a set of thresholds to the confidence output by a method. On the x-axis is plotted the *recall* (what proportion of object instances in the image set have been detected); on the y-axis is plotted the *precision* (what proportion of the detections actually correspond to correct object instances). The PR curve makes it easy to observe the trade-off between the two; some methods may have high precision but low recall, for example detecting a particular view of an object reliably, where other methods may give more balanced performance. Use of Precision/Recall as opposed to the Receiver Operating Characteristic was chosen to provide a standard scale for evaluation which is independent of the algorithmic details of the methods, for example whether a "window scanning" mechanism or other means were used.

As in the classification case, a definitive measure for quantitative evaluation of PR curves is not possible, because of the possible requirements for different emphasis at low or high recall. The challenge used the interpolated Average Precision (AP) measure defined by the Text Retrieval Conference (TREC). This measures the mean precision at a set of eleven equally spaced recall levels $[0, 0.1, \ldots, 1]$:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{interp}(r)$$

The precision at each recall level r is *interpolated* by taking the maximum precision measured for a method for which the corresponding recall exceeds r:

$$p_{interp}(r) = \max_{\tilde{r}:\tilde{r} \geq r} p(\tilde{r}) \tag{1}$$

where $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

Figure 5 shows the interpolated precision values for the measured curve shown. Use of the interpolated precision ameliorates the effects of different sampling of recall that each method may produce, and reduces the influence of the "sawtooth" pattern of temporary false detections typical of PR curves. Because the AP measure includes measurements of precision across the full range of recall, it penalizes methods which achieve low total recall (failing to detect some proportion of object instances) as well as those with consistently low precision.

Evaluation of Bounding Boxes. Judging each detection output by a method as either a true positive (object) or false positive (non-object) requires comparing

the corresponding bounding box predicted by the method with ground truth bounding boxes of objects in the test set. 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} was required to exceed 50% by the formula

$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{2}$$

The threshold of 50% was set deliberately low to account for inaccuracies in bounding boxes in the ground truth data, for example defining the bounding box for a highly non-convex object, e.g. a side view of a motorbike or a car with an extended radio aerial, is somewhat subjective.

Detections output by a method were assigned to ground truth objects satisfying the overlap criterion in order ranked by the (decreasing) confidence output. Lower-ranked detections of the same object as a higher-ranked detection were considered false positives. The consequence is that methods producing *multiple* detections of a single object would score poorly. All participants included algorithms in their methods to arbitrate between multiple detections.

5.2 Competitions and Participation

Four competitions were defined for the detection task, by the choice of training data: provided for the challenge, or the participant's own data; and the test set used: the "easier" `test1` images, or the "harder" `test2` images. Table 5 summarizes the competitions. For each competition, performance on each of the four object classes was evaluated. Participants were free to submit results for any or all of the object classes.

Table 5. Competitions for the *detection* task, defined by the choice of training data and test data.

No.	Task	Training data	Test data
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`

Table 6 lists the participation in competitions 5 and 6, which used the provided `train+val` image set for training. Five of the twelve participants entered results for these competitions. All five of these participants tackled the motorbike class, four the car class, and three the people class. Edinburgh submitted baseline results for all four classes. The concentration on the motorbike and car classes is expected as these are more typical "opaque" objects which have attracted most attention in the object recognition community; recognition of

Table 6. Participation in the *detection* task. Bullets indicate participation in the competition for a particular test set and object class.

	`test1`				`test2`			
	motorbikes	bicycles	people	cars	motorbikes	bicycles	people	cars
Aachen	–	–	–	–	–	–	–	–
Darmstadt	•	–	–	•	•	–	–	•
Edinburgh	•	•	•	•	•	•	•	•
FranceTelecom	•	–	–	•	•	–	–	•
HUT	–	–	–	–	–	–	–	–
INRIA-Dalal	•	–	•	•	•	–	•	•
INRIA-Dorko	•	–	•	–	–	–	–	–
INRIA-Jurie	–	–	–	–	–	–	–	–
INRIA-Zhang	–	–	–	–	–	–	–	–
METU	–	–	–	–	–	–	–	–
MPITuebingen	–	–	–	–	–	–	–	–
Southampton	–	–	–	–	–	–	–	–

more "wiry" objects (bicycles) or articulated objects (people) has been a recent development.

Only one participant, INRIA-Dalal, submitted results for competitions 7 and 8, in which training data other than that provided for the challenge could be used. This participant submitted results for the people class on both `test1` and `test2` image sets.

5.3 Overview of Detection Methods

Section 6 gives full details of the methods used by participants. The approaches used for the detection task can be broadly divided into three categories:

Configurations of Local Image Features. Two participants: Darmstadt and INRIA-Dorko used an approach based on local image features. These methods use interest point detectors and local image features represented as "visual words", as used by many of the methods in the classification task. In contrast to the classification task, the detection methods explicitly build a model of the spatial arrangement of the features; detection of the object then requires image features to match the model both in terms of appearance and spatial configuration. The two methods proposed differed in terms of the feature representation: patches of pixels/SIFT descriptors, clustering method for dictionary or "codebook" learning, and voting scheme for detection. Darmstadt used a Minimum Description Length (MDL) method to refine ambiguous detections and an SVM classifier to verify detections. INRIA-Dorko added a measure of discriminative power of each visual word to the voting scheme.

Window-based Classifiers. Two participants: FranceTelecom and INRIA-Dalal used "window-based" methods. In this approach, a fixed sized window is scanned over the image at all pixel positions and multiple scales; for each window, a classifier is applied to label the window as object or non-object, and positively labelled windows are grouped to give detections. FranceTelecom used a Convolutional Neural Network (CNN) classifier which applies a set of successive feature extraction (convolution) and down-sampling operations to the raw input image. INRIA-Dalal used a "histogram of oriented gradient" representation of the image window similar to computing SIFT descriptors around grid points within the window, and an SVM classifier.

Baseline Methods. Edinburgh proposed a set of "baseline" detection methods. Confidence in detections was computed either as the prior probability of a class from the training data, or using the classifier trained for the classification task. Several baseline methods for proposing bounding boxes were investigated including simply proposing the bounding box of the entire image, the mean bounding box from the training data, the bounding box of all strong interest points, or bounding boxes based on the "purity" of visual word representations of local features with respect to a class.

5.4 Discussion of Detection Methods

There have been two main approaches to object detection in the community: (i) window-based methods, which run a binary classifier over image windows, effectively turning the detection problem into a large number of whole-image classification problems; (ii) parts-based methods, which model objects as a collection of parts in terms of local appearance and spatial configuration. It is valuable that both these approaches were represented in the challenge. The methods proposed differ considerably in their representation of object appearance and geometric information. In the INRIA-Dalal method, a "holistic" representation of primitive local features (edges) is used; the position of features is encoded implicitly with respect to a fixed coordinate system. The FranceTelecom method might be understood as learning the approximate position of local object parts; the convolution operations can be viewed as part detection, and the sub-sampling steps introduce "slack" in the coordinate frame. The Darmstadt and INRIA-Dorko methods *explicitly* decompose the object appearance into local parts and their spatial configuration. It is particularly interesting to see how these methods compare across more rigid objects (cars/motorbikes), and those for which the shape of the object changes considerably (people).

6 Participants

Twelve participants took part in the challenge. We include here participants' own descriptions of the methods used.

6.1 Aachen

Participants: Thomas Deselaers, Daniel Keysers, Hermann Ney
Affiliation: RWTH Aachen, Aachen, Germany
E-mail: {deselaers,keysers,ney}@informatik.rwth-aachen.de
WWW: http://www-i6.informatik.rwth-aachen.de/

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

1. patch extraction
2. clustering
3. creation of histograms
4. discriminative training and classification

where the first three steps are feature extraction steps and the last is the actual classification step. This approach was first published in [12] and was extended and improved in [13].

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

The steps of the method are briefly outlined in the following paragraphs.

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

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

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

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

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

Training and Classification. Having obtained this representation by histograms of image patches, we define a decision rule for the classification of images. The approach based on maximum likelihood of the class-conditional distributions does not take into account the information of competing classes during training. We can use this information by maximizing the class posterior probability $\prod_{k=1}^{K} \prod_{n=1}^{N_k} p(k|X_{kn})$ instead. Assuming a Gaussian density with pooled covariances for the class-conditional distribution, this maximization is equivalent to maximizing the parameters of a log-linear or maximum entropy model

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

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

Conclusions. The method performs well for various tasks (e.g. Caltech {airplanes, faces, motorbikes}), was used in the ImageCLEF 2005 Automatic Annotation Task[14] where it performed very well, and also performed well in the PASCAL Visual Object Classes Challenge described in this chapter. An important advantage of this method is that it is possible to visualize those patches

[14] http://ir.shef.ac.uk/imageclef2005/

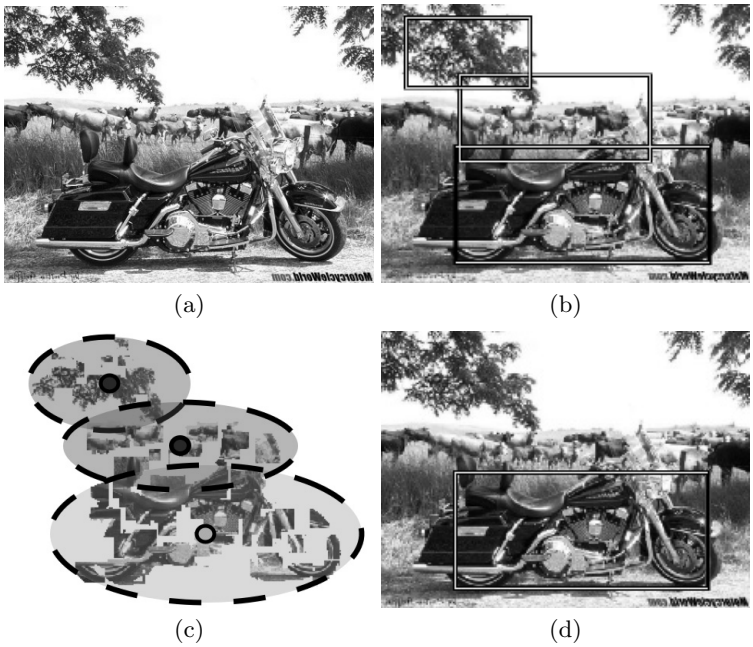


Fig. 6. *Darmstadt:* Illustration of the IRD approach. (a) input image; (b) detected hypothesis by the ISM model using a rather low threshold; (c) input to the SVM stage; (d) verified hypothesis.

which are discriminative for a certain class, e.g. in the case of faces it was learned that the most discriminative parts are the eyes.

6.2 Darmstadt

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

We submit results on the categories car and motorbike obtained with the *Implicit Shape Model (ISM)* [28] and the *Integrated Representative Discriminant (IRD)* approach [19]. The ISM in itself is an interesting model, as it has recently shown impressive results on challenging object class detections problems [30]. The IRD approach augments the representative ISM by an additional discriminant stage, which improves the precision of the detection system.

Local Feature Representation. We use local features as data representation. As scale-invariant interest point detector we use difference-of-Gaussians and as

region descriptor we use normalized raw pixel patches. Even though there exist more sophisticated descriptors, we want to point out that due to the rather high resolution of 25×25 pixels the representation is quite discriminant. The high dimensionality of the resulting features is taken care of by the quantization of the feature space via soft-matching to a codebook. More recently [35] [41] we have used more efficient feature representation for the task of object class detection.

Codebook. In both approaches, we use a codebook representation as a first generalization step, which is generated by an agglomerative clustering scheme. 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. This leaves us with 55 car images and 153 motorbike images for building the codebook and learning the model.

Learning and Evaluating the Model. The basic idea of the ISM is to represent the appearance of an object by a non-parametric, spatial feature occurrence distribution for each codebook. When using the model for detection, local feature are computed from the test image and afterwards matched to the codebook. Based on these matches, the spatial distributions stored in the ISM can be used to accumulate evidence for object hypothesis characterized by position in the image and size of the object. For a more detailed description - in particular how to achieve scale-invariance - we refer to [29].

MDL Hypothesis Verification Stage. As the ISM facilitates the use of segmentation masks for increased performance, we included the provided annotations in the training. Given this information, a pixel-level segmentation can be inferred on the test images. On the one hand this information can be fed back in the recognition loop for interleaved recognition and segmentation [28]. On the other hand, the problem of accepting a subset of ambiguous hypothesis in an image can be formulated as an optimization problem in a MDL framework based on the inferred figure and background probabilities[28]. For both methods submitted to the challenge we make use of the MDL stage.

SVM with Local Kernel of IRD Approach. The SVM validation stage is trained on detections and false alarms of the ISM on the whole training set for cars and motorbikes. We want to point out, that both systems work on the same data representation, so that the SVM makes full use of the information provided by the ISM. A hypothesis consists of an inferred position of the object centre in the image, an inferred object scale and a set of features that are consistent with this hypothesis. Based on this information, the SVM is used to eliminate false positives of the representative ISM model during detection. The whole process is illustrated in Figure 6.

Besides the fact, that it is appealing to combine representative and discriminant models from a machine learning point of view, we also profit from the explicit choices of the components: While part of the success of the ISM is a result of its capability for "across instances" learning, the resulting hypothesis can lack global consistency which result in superfluous object parts. By using an SVM with a kernel function of appearance *and* position we enforce a global consistency again. The benefit of enforcing global consistencies were studied in more detail in [30].

Experiments. 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 trade-off, we report two performance curves for the detection tasks. One for optimal equal error rate (EER) performance and one for optimized precision (labelled "ISMSVM_2" in the plots).

Notes on the Results. The models were exclusively trained on side-views. As the test data also includes multiple viewpoints, 100 % recall is not reachable given the used training scheme. Given that test-set 1 contains only side-views for the motorbikes and approximately 59% side-views for the cars and 39% and 12% for test-set 2 respectively, we detect nearly all side-views with a high level of precision.

6.3 Edinburgh

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

Experiments. 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 images[15] [37]. SIFT representations are then found for the image regions chosen by the interest point detector [32]. The SIFT representations for all the

[15] We used code from the Oxford Visual Geometry Group available at http://www.robots.ox.ac.uk/∼vgg/research/affine/.

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. [10].

All the systems were trained only on the provided training data (`train`), with parameters optimised using the provided validation data (`val`). The test data sets were only used in the final runs of the systems to obtain results for submission. All the detectors described below assume a *single* object of interest per image.

`Edinburgh_bof` **Classifier.** This classifier uses logistic regression[16], 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.

`Edinburgh_meanbb` **Detector.** This naïve 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.

`Edinburgh_wholeimage` **Detector.** This naïve approach is intended to act as a baseline result. All images in the test set are assigned the class probability as their confidence level. The object bounding box is simply set to the perimeter of the test image.

`Edinburgh_puritymeanbb` **Detector.** 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

[16] We used the Netlab logistic regression function, `glm`.

weight of each interest point's 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).

The size and shape of the bounding box for all detections was set to that of the class mean bounding box, scaled according to the size of the image. The class mean bounding box was calculated as for the `Edinburgh_meanbb` method.

Confidences are calculated by the bag-of-features classifier, as described for `Edinburgh_bof`, 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.

`Edinburgh_siftbb` **Detector.** This detector assigns the confidence levels calculated by the bag-of-features classifier, as described for `Edinburgh_bof`, while bounding boxes are predicted as the tight bounding box of the interest points found in the image by the Harris-Affine detector.

Discussion. Our entries consisted of one straightforward 'bag-of-features' approach to classification and four simple approaches to the detection task. In comparison to other entries tackling the classification task, the performance of our bag-of-features classifier was almost always behind that of the competitors. By the area under ROC curve measure (AUC), it achieved only 0.77, 0.72, 0.60 and 0.80 on the four object categories compared with 0.88, 0.82, 0.82 and 0.91 for the next highest competitor in each case. In all but the final category (cars), this meant our entry was the poorest performer.

Following discussions at the challenge workshop, we modified our approach in two small ways and performance improved considerably, to 0.89, 0.87, 0.81 and 0.85. The changes we made were to: 1) train our classifier on the `train+val` data set instead of only the `train` data set; and 2) normalise the bag-of-feature representation vectors. This first modification provided substantially more training data with which to refine the decision boundary of the classifier, leading to a small improvement in performance. The fact that the second modification led to such a significant performance increase suggests it is the *proportions* of the different visual words in the image that are useful for classification rather than their absolute number. This makes sense, as the images (and the objects within them) are commonly of different sizes and hence the number of features representing them varies.

Our approaches to the detection task were intended as simple baselines from which to judge the more complex approaches of the other competitors. Such baselines were widely acknowledged as useful by attendees at the workshop, and serve to highlight the real progress made in tackling this challenging task by the other entries.

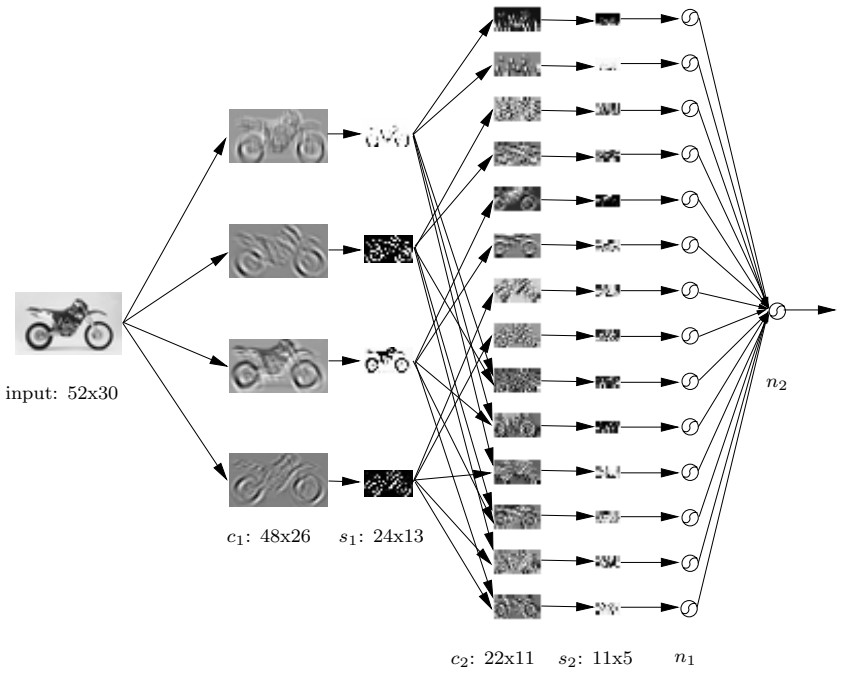


Fig. 7. *FranceTelecom:* Architecture of the Convolutional Object Finder (COF) system

6.4 FranceTelecom

Participants: Christophe Garcia, Stefan Duffner
Affiliation: France Télécom division R&D, Cesson Sévigné, France
E-mail: christophe.garcia@francetelecom.com
WWW: http://www.francetelecom.com/rd/

The proposed system, called Convolutional Object Finder (COF), is based on a convolutional neural network architecture inspired from our previous work on face detection [20]. It automatically synthesises simple problem-specific feature extractors and classifiers, from a training set of object and non-object patterns, without making any assumptions concerning the features to be extracted or the areas of the object pattern to be analysed. Once trained, for a given object, the COF acts like a fast pipeline of simple convolution and subsampling modules that treat the raw input image as a whole, at different scales, without requiring any local pre-processing of the input image (brightness correction, histogram equalisation, etc.).

The COF system consists of six layers, excepting the input plane (retina) that receives an image area of fixed size (52×30 pixels in the case of motorbikes) to be classified as *object* or *non-object* (see Fig.7). Layers c_1 through s_2 contain

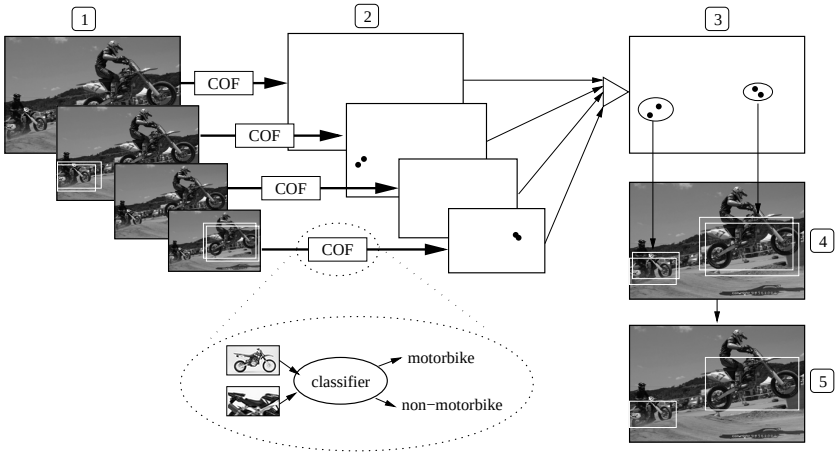


Fig. 8. *FranceTelecom:* Different steps of object localisation

a series of planes where successive convolutions and subsampling operations are performed.

These planes are called *feature maps* as they are in charge of extracting and combining a set of appropriate features. Layer n_1 contains a number of partially connected sigmoid neurons and layer n_2 contains the output unit of the network. These last two layers carry out the classification task using the features extracted in the previous layers.

The neural network is fully trained using a modified version of the backpropagation algorithm, by receiving object and non-object images with target answer $+1$ and -1 respectively. The positive training sets (object images) are augmented by virtual examples, generated by slightly rotating, translating and scaling the original object images. In order to reduce the number of false alarms, the set of negative (non-object) examples is iteratively constructed by a bootstrapping procedure. It uses a set of scenery images that do not contain the object to detect, in order to extract non-object image areas that produce a positive output value (false alarm) greater than a certain threshold. This threshold is initialised with a high value (e.g. 0.8) and is gradually decreased (until 0.0), throughout the iterative learning process, so that a rough class boundary is quickly found in the first iterations and refined later on.

In order to detect objects at different scales, the COF system is placed into a multi-scale framework as depicted in Fig. 8.

The input image is repeatedly subsampled by a factor of 1.2, resulting in a pyramid of images (step 1). Each image of the pyramid is then filtered by our convolutional network COF (step 2). After this processing step, object candidates (pixels with positive values in the result image) in each scale are mapped back to the input image scale (step 3). They are then grouped according to their proximity in image and scale spaces. Each group of object candidates is fused into a representative object whose centre and size are computed as the

centroids of the centres and sizes of the grouped objects, weighted by their individual network responses (step 4). After applying this grouping algorithm, the set of remaining representative object candidates serve as a basis for finer object localisation and false alarm dismissal (step 5). This finer localisation consists of a local search with smaller scale steps in a limited area around each object candidate. In order to remove false alarms, the sum of positive network outputs over the local search space is computed at each candidate position and candidate areas with a value below a certain threshold are rejected.

Experimental results show that the proposed system is very robust with respect to lighting, shape and pose variations as well as noise, occlusions and cluttered background. Moreover, processing is fast and a parallel implementation is straightforward. However, it should be noticed that detection results can be drastically enhanced if a large training set of thousands of object images is made available. As future extensions, we plan to enhance the COF system by allowing a variable aspect ratio for the retina image that will help to cope with highly variable 3D object shapes and poses.

6.5 HUT

Participants: Ville Viitaniemi, Jorma Laaksonen, Markus Koskela
Affiliation: Helsinki University of Technology, Helsinki, Finland
E-mail: {ville.viitaniemi,jorma.laaksonon,
markus.koskela}@hut.fi

Overview of the PicSOM system. For all the experiments we used a similar setup utilising our general purpose content-based image retrieval system named PicSOM [26]. 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 examples. The system operates in its interactive mode by the principles of *query by pictorial example* and *relevance feedback*. In these experiments, 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 basic principle of the PicSOM system is to use Self-Organizing Maps (SOMs), which are two-dimensional artificial neural networks, to organise and index the images of the collection. The SOM orders images so that visually similar images – in the sense of some low-level statistical feature – are mapped to the same or nearby map units in a two-dimensional grid. The PicSOM system inherently uses multiple features, creates a separate index for each of them and uses them all in parallel when selecting the retrieved images. In our current experiments, we used the system to give a qualification value for every image in the test set. That way we could order them in the order of descending similarity to the corresponding set of training images.

The visual features that were used to describe the images were chosen among the ones that were already 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. 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 the relative brightness of neighbouring pixels

Details of the Experimental Procedure. The PicSOM system was applied to the image classification task using the following procedure:

1. The training set images were projected to the parallel feature SOMs.
2. The distance of the projection of a given test image was locally compared with the nearby projections of positive and negative training images. This was achieved by kernel smoothing the SOM surface field defined by the positive and negative training impulses.
3. The results from the parallel feature SOMs were summed together.

In the time frame of the VOC challenge, we were not able to utilise the training set annotations beyond the presence/absence information, i.e. the bounding boxes and other additional annotations were not used.

System parameters were tuned using the validation set performance as an optimisation criterion. Feature selection was performed based on the performance in the validation set. As the performance measure we used the area under the ROC curve. All four target classes were processed separately and the optimisations led us to use four different sets of features. We used all four optimised feature sets for all four classes. This resulted in the total of 16 result sets submitted. Other parameters, such as the size of the SOMs, were not optimised. The final results were obtained by using only the union of the provided training and validation data sets in the training of the SOMs in the system.

The training, validation and testing set images were automatically segmented to a few parallel segmentations with predetermined numbers of segments. Visual features were extracted from both the segments and the whole images. Separate SOMs were trained for the segment features and the whole-image features. Figure 9 illustrates the use of the image segments and the parallel SOMs.

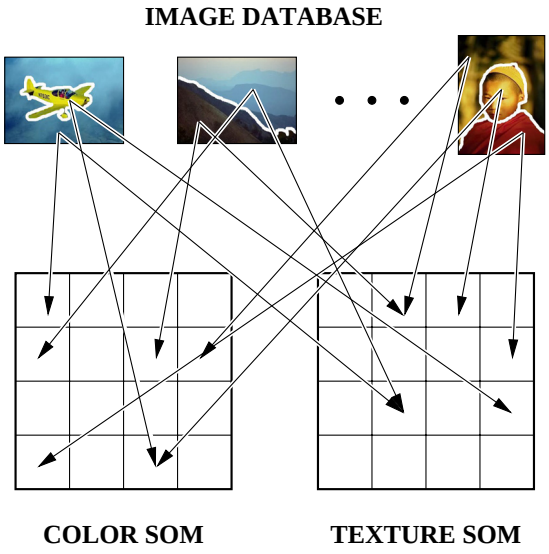
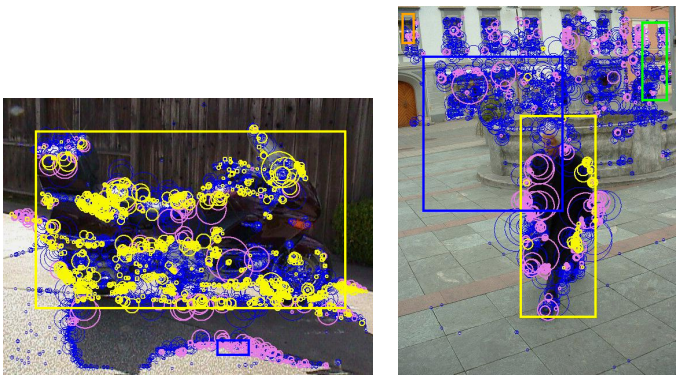


Fig. 9. *HUT:* An example of using two parallel SOM indices for segmented images int the PicSOM system. The colour and texture SOMs are trained with image segments and each segment is connected to its best-matching map unit on each SOM.



Motorbike People

Fig. 10. *INRIA-Dorko:* Example detections on test images for motorbike (left) and people (right). Blue (dark) points are eliminated due to feature selection, and yellow (bright) points vote for the best solution (yellow rectangle). Non-yellow rectangles indicate false detections with lower confidence.

6.6 INRIA-Dalal

Participants: Navneet Dalal, Bill Triggs
Affiliation: INRIA Rhône-Alpes, Montbonnot, France
E-mail: {navneet.dalal,bill.triggs}@inrialpes.fr

Introduction. INRIA participated in eight of the object detection challenges with its Histogram of Oriented Gradient (HOG) object detector: competitions 5 and 6 for classes Motorbike, Car, Person and competitions 7 and 8 for class Person.

In use, the detector scans the image with a detection window at multiple positions and scales, running an object/non-object classifier in each window. Local maxima of "object" score are found, and if the score is above threshold a detection is declared. The classifier is a linear SVM over our HOG feature set, trained using SVM-Light [22, 23].

The Histogram of Oriented Gradient feature set is described in detail in [11]. Here we focus on giving information not available in [11], but briefly, after some optional input normalization, we calculate image gradients at each pixel, use the gradient orientation to select an orientation channel and the gradient magnitude as a weight to vote into it, and accumulate these votes over small local regions called *cells*. Each cell thus contains a weighted gradient orientation histogram. The cells are gathered into somewhat larger spatial *blocks*, and the block's histograms are normalized as a group to provide local illumination invariance. The final descriptor is the concatenated vector of all channels of all cells of all blocks in the detection window. To reduce aliasing, the implementation includes careful spatial and angular interpolation and spatial windowing of the cells within each block. The blocks are usually chosen to overlap so (modulo windowing effects) each cell's histogram appears several times with different normalizations. The window is usually chosen somewhat (10-20%) larger than the object as including context helps recognition.

Data Preparation. For training and validation we use the size-normalized object boxes from the positive *train* and *val* sets. The corresponding negatives are sampled randomly from negative images. To allow for context we include an 8 or 16 pixel margin around the image window. Table 7 lists the window sizes and key parameters of each detector.

HOG Parameter Optimization. HOG descriptors have a number of parameters to set. This was done using the PASCAL *train* and *val* sets respectively for training and validation. For the window sizes given in table 7, the following settings turned out to be optimal for all of the object classes: taking the square root of image intensities before computing gradients; 20° orientation bins (9 in 180° or 18 in 360°); 2×2 blocks of 8×8 pixel cells; and an inter block

stride of 8 pixels (so each cell belongs to 4 blocks). Two settings changed from class to class. *(i)* Including the signs of gradients (*i.e.* using orientation range $0{-}360°$ rather than $0{-}180°$) is helpful for classes in which local contrasts typically have consistent signs (*e.g.* cars and motorcycles with their dark tyres on light rims), and harmful for less consistent classes (*e.g.* humans with their multi-coloured clothing). *(ii)* Regarding normalization, L2-Hys (L2-norm followed by Lowe-style clipping and renormalization [32]) and L1-Sqrt (L1-norm followed by square root, *i.e.*, $\mathbf{v} \rightarrow \sqrt{\mathbf{v}/(\|\mathbf{v}\|_1 + \epsilon)}$) typically have comparable performance, but for the motorbike class L1-Sqrt significantly outperforms L2-Hys. We suspect that this happens because L1-Sqrt provides more robustness against the rapid fine-detail gradient changes that are common in motorcycle wheels.

Table 7. The key parameters for each trained detector.

Class	Window Size	Avg. Size	Orientation Bins	Normalization Method	Margin (see §6.6)
Person	56×112	Height 80	9 ($0{-}180°$)	L2-Hys	12
Car	112×56	Height 40	18 ($0{-}360°$)	L2-Hys	8
Motorbike	144×80	Width 112	18 ($0{-}360°$)	L1-Sqrt	4

Multi-scale Detection Framework. To use the above window-based classifier for object detection, it is scanned across the image at multiple scales, typically firing several times in the vicinity of each object. We need to combine these overlapping classifier hits to produce a detection rule that fires exactly once for each observed object instance. We treat this as a maximum finding problem in the 3D position-scale space. More precisely, we convert the classifier score to a weight at each 3D point, and use a variable bandwidth Mean Shift algorithm [9] to locate the local maxima of the resulting 3D "probability density". Mean Shift requires positive weights, and it turns out that clipped SVM scores $\max(\text{score}, 0)$ work well. The (variable) bandwidth for each point is given by $(\sigma_x s, \sigma_y s, \sigma_s)$ where s is the detection scale and $\sigma_x, \sigma_y, \sigma_s$ are respectively the x, y and scale bandwidths. We use $\sigma_s = 30\%$ and set (σ_x, σ_y) to $(8, 16)$ for the Person class and $(16, 8)$ for the Motorbike and Car classes – *i.e.* proportional to the aspect ratio of the detection window, as in practice the multiple detections tend to be distributed in this way.

We perform one final step. The challenge rules consider detections to be false if they have less than 50% area overlap with the marked object box, and as our detection windows have been slightly enlarged to include some background context, we need to shrink them again. Different classes occupy different amounts of their bounding boxes on average, so we do this adaptively. For each class, we learn a final classifier on the combined *train+val* data set (with settings chosen by validation on *val* after training on *train*). Using this classifier on *train+val*, we calculate precision-recall curves for several different window shrinkage factors

and choose the factor that gives the best overall performance. Table 7 lists the chosen shrinkage margins in pixels relative to the detection window size. Note that this tuning is based on training data. For each challenge we performed just one run on test set, whose results were submitted to the challenge.

Additional Comments. We did not have time to optimize the window size of our motorbike classifier before the challenge, but afterwards we found that larger windows are preferable – 144×80 here, versus 112×56 in our original challenge submission. The performance of the new classifier is comparable to the two best results in the challenge.

6.7 INRIA-Dorko

Participants: Gyuri Dorkó, Cordelia Schmid
Affiliation: INRIA Rhône-Alpes, Montbonnot, France
E-mail: {gyuri.dorko,cordelia.schmid}@inrialpes.fr

Introduction. We have participated in the localization competition for people and motorbikes. Our method combines class-discriminative local features with an existing object localization technique to improve both its speed and performance. Our system learns the spatial distribution of the object positions for automatically created discriminative object-parts, and then, uses the generalized Hough-transform to predict object locations on unseen test images.

Feature Extraction. Images are described by sparse local features extracted with a scale-invariant interest point operator. We use a modified version of the multi-scale Harris detector [21]. Interest regions are described by the Scale Invariant Feature Transform (SIFT) [32] computed on a 4x4 grid and for 8 orientation bins, resulting in a 128 dimensional descriptor.

Training. We first learn a vocabulary of size 1200 from the scale-invariant features of the training set. We use expectation-maximization (EM) to estimate a Gaussian Mixture Model with a diagonal covariance matrix. Then, we assign a rank to each cluster based on its discriminative power as in [15]. Our criterion is derived from the likelihood score, and prefers rare but very discriminative object-parts. The rank for cluster C_i is defined as:

$$\tilde{P}^{+}(C_i) = \frac{\sum_{\mathbf{v_j} \in D^{+}} P(C_i|\mathbf{v_j})}{\sum_{\mathbf{v_j} \in D^{+}} P(C_i|\mathbf{v_j}) + \sum_{\mathbf{v_j} \in D^{-}} P(C_i|\mathbf{v_j})} \tag{3}$$

where D^{+} and D^{-} are the set of descriptors extracted from positive and negative images respectively and $P(C_i|\mathbf{v_j})$ is the probability of component C_i given

descriptor $\mathbf{v_j}$. We then learn the spatial distribution of the object positions and scales for each cluster. For each training image, we assign all descriptors inside the rectangle locating the object to its cluster (by MAP), and record the centre (x,y) and the scale (width and height) of the rectangle with respect to the assigned cluster. This step is equivalent to [29] with the difference that we collect the width and height separately and that we do not use the figure-ground segmentation of the object. The output of our training is a list of clusters with the following properties:

- the mean and variance representing the appearance distribution of the cluster,
- a probabilistic score for its discriminative power,
- and a spatial distribution of the object positions and scales.

Localization by Probabilistic Hough Voting. The localization procedure on a test image is similar to the initial hypothesis generation of Leibe *et al.* [29]. The difference is that we incorporate the discriminative capacity into the voting scheme: only the 100 most discriminative clusters participate in the voting, and the probabilistic score is integrated into the voting scheme. This allows better confidence estimations for the different hypotheses. Our algorithm is the following. The extracted scale-invariant features of the test image are assigned to the closest cluster by appearance (MAP). Then, the chosen clusters vote for possible object locations and scales (4D space). In practice we simplified the voting scheme from [29] by only allowing one cluster per descriptor to vote, and extended their formulation by weighting each vote with the discriminative score from (3). The predicted object locations and scales are found as maxima in the 4D voting space using the Mean-Shift[8] algorithm with a scale-adaptive balloon density estimator[9, 29]. The confidence level for each detection is determined by the sum of the votes around the object location in the voting space. Fig. 10 shows example detections on test images.

6.8 INRIA-Jurie

Participants: Frederic Jurie, Gyuri Dorkó, Diane Larlus, Bill Triggs
Affiliation: INRIA Rhône-Alpes, Montbonnot, France
E-mail: frederic.jurie@inrialpes.fr

We participated in the competition 1 for all four categories.
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 (see Fig. 11). 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 [10], then an SVM classifier is trained

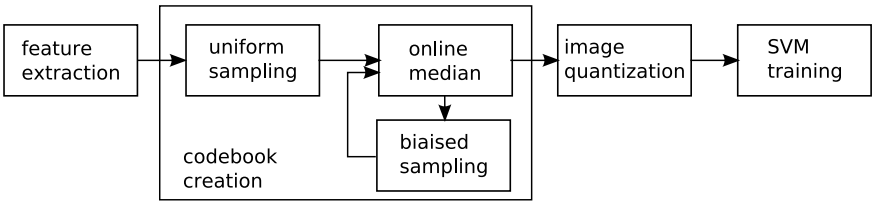


Fig. 11. *INRIA-Jurie:* Outline of the learning steps. See text for details.

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 20000 regions per image. Each region is then represented by a 128 dimensional SIFT descriptor [32], 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 highly populated; the large number of features per image leads to a total of several hundred thousand for the entire training set (`train+val`). Second, the dense feature set is extremely unbalanced as was shown in [24]. Therefore, to obtain a discrete set of labels on the descriptors we have designed a new clustering algorithm [27] 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 a large number of descriptors.

Our iterative approach discovers new clusters at each step by consecutively calling a sampling and a k-median algorithm (see Fig. 11) 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 centres. This is realized by introducing an *influence radius* to affect points close to already found centres. All affected descriptors are then excluded from any further sampling. Fig. 12 illustrates our sampling step. The influence radius ($r = 0.6$) and the total number of clusters ($k = 2000$) are parameters of our method.

The biased sampling is followed by the *online median* algorithm proposed by Mettu and Plaxton [34]. Their method is based on the *facility location* problem and chooses the centres one by one. At each iteration of our algorithm we discover 20 new centres by this algorithm.

We keep all the parameters of our codebook creation algorithm fixed and set by our earlier experience, *i.e.* they are not tuned for the PASCAL Challenge

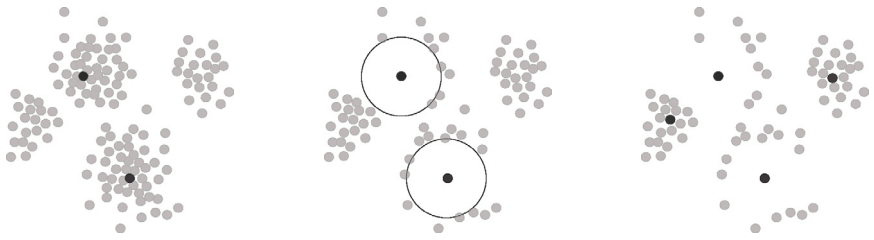


Fig. 12. *INRIA-Jurie:* Biased sampling. (a) assumes that we discovered 2 new centres in the previous step, which is marked by the two black points. (b) The influence radius determines an affectation ball around each centre. (c) All descriptors within these balls are removed and the remaining portion is then random sampled.

database. For the creation of the codebook we originally cropped the training images based on the provided bounding boxes, but later we discovered that our result remain the same using the full images. (ROC curves are reported with the cropped images.)

Image Quantization. Both learning and testing images are represented by the *bag of features* approach [10], 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 within the predefined influence radius. Our association discards descriptors that fall out of all affectation balls; they are considered as outliers. To measure the distance between SIFT features we used the Euclidean distance as in [32].

Classification. We used the implementation of [6] to train linear SVM classifiers on the normalized image histograms. In the first set of experiments (indicated by `dcb_p1` on our reports) we trained the SVMs on binary histograms, each bin indicating the presence or absence of the codebook elements. In the second set of experiments (indicated by `dcb_p2`), a standard vector normalisation is used.

6.9 INRIA-Zhang

Participants: Jianguo Zhang, Cordelia Schmid
Affiliation: INRIA Rhône-Alpes, Montbonnot, France
E-mail: {jianguo.zhang,cordelia.schmid}@inrialpes.fr

Abstract. Our approach represents images as distributions of features extracted from a sparse set of keypoint locations and learns a Support Vector Machine

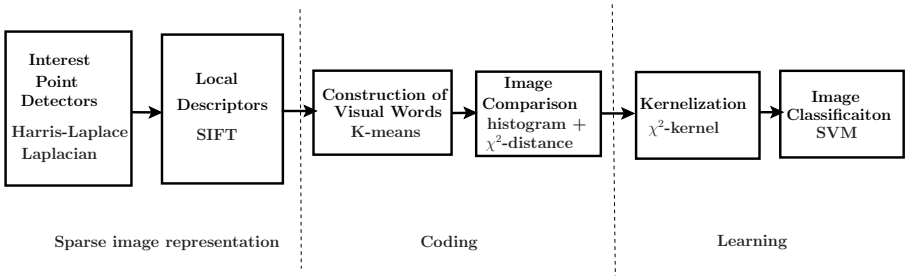


Fig. 13. *INRIA-Zhang:* Architecture of the approach.

classifier with a kernel based on an effective measure for comparing distributions. Results demonstrate that our approach is surprisingly effective for classification of object images under challenging real-world conditions, including significant intra-class variations and substantial background clutter.

Introduction. Fig. 13 illustrates the different steps of our approach. We first compute a sparse image representation by extracting a set of keypoint locations and describing each keypoint with a local descriptor. We then compare image distributions based on frequency histograms of visual words. Finally, images are classified with a χ^2-kernel and a Support Vector Machine (SVM).

A large-scale evaluation of our approach is presented in [44]. This evaluation shows that to achieve the best possible performance, it is necessary to use a combination of several detectors and descriptors together with a classifier that can make effective use of the complementary types of information contained in them. It also shows that using local features with the highest possible level of invariance usually does not yield the best performance. Thus, a practical recognition system should seek to incorporate multiple types of complementary features, as long as their local invariance properties do not exceed the level absolutely required for a given application. An investigation of the influence of background features on recognition performance shows the pitfalls of training on datasets with uncluttered or highly correlated backgrounds, since this yields disappointing results on test sets with more complex backgrounds.

Sparse Image Representation. We use two complementary scale-invariant detectors to extract salient image structures: *Harris-Laplace* [37] and *Laplacian* [31]. Harris-Laplace detects corner-like structures, whereas the Laplacian detects blob-like ones. Both detectors output circular regions at a characteristic scale.

SIFT features [32] are used to describe the scale normalized regions; it has been shown to outperform a set of existing descriptors [38]. SIFT computes the gradient orientation histogram for the support region. We use 8 orientation

planes. For each orientation, the gradient image is sampled over a 4×4 grid of locations, resulting in a 128 feature vector.

For the training images and test set 1 of the PASCAL challenge (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 for Harris-Laplace and 71 for the Laplacian. This illustrates the sparsity of our image representation.

Comparing Distributions of Local Features. We first construct a visual vocabulary from the local features and then represent each image as a histogram of visual words. The vocabulary for the PASCAL challenge is obtained as follows: 1) We randomly select 50000 descriptors from the training images of one class. 2) We cluster these features with k-means ($k = 250$). 3) We concatenate the cluster centres of the 4 classes to build the global vocabulary of 1000 words.

A histogram measures the frequency of each word in an image. Each feature in the image is assigned to the closest word. We use the χ^2 distance to compare histograms:

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

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

Kernel-based Classification. We use an extended Gaussian kernel [7]:

$$K(S_i, S_j) = \exp(-1/A \cdot D(S_i, S_j))$$

where $D(S_i, S_j)$ is the χ^2 distance and S_i, S_j are vocabulary histograms. The resulting χ^2 *kernel* is a Mercer kernel.

Each detector/descriptor pair can be considered as an independent channel. To combine different channels, we sum their distances, i.e., $D = \sum D_i, i = 1, ..., n$ where D_i is the similarity measure of channel i. The kernel parameter A is obtained by 5-fold cross validation on the training images.

For classification, we use *Support Vector Machines* [40]. For a two-class problem the decision function has the form $g(x) = \sum_i \alpha_i y_i K(x_i, x) - b$, where $K(x_i, x)$ is the value of a *kernel function* for the training sample x_i and the test sample x. The $y_i \in \{-1, +1\}$ and α_i are the class label and the learned weight of the training sample x_i. b is a learned threshold parameter. The training samples with $\alpha_i > 0$ are usually called *support vectors*.

We use the two-class setting for binary detection, i.e., classifying images as containing or not a given object class. If we have m classes ($m = 4$ for the PASCAL challenge), we construct a set of binary SVM classifiers $g_1, g_2, ..., g_m$, each trained to separate one class from the others. The SVM score is used as a confidence measure for a class (normalized to $[0, 1]$).

Conclusions. Our bag-of-keypoints method achieves excellent results for object category classification. However, successful category-level object recognition and localization is likely to require more sophisticated models that capture the 3D shape of real-world object categories as well as their appearance. In the development of such models and in the collection of new datasets, bag-of-keypoints methods can serve as effective baselines and calibration tools.

6.10 METU

Participants: Ilkay Ulusoy[1], Christopher M. Bishop[2]
Affiliation: [1]Middle Eastern Technical University, Ankara, Turkey
[2]Microsoft Research, Cambridge, UK
E-mail: ilkay@metu.edu.tr, cmbishop@microsoft.com

We follow several recent approaches [32, 37] and use an interest point detector to focus attention on a small number of local patches in each image. This is followed by invariant feature extraction from a neighbourhood around each interest point. Specifically we use DoG interest point detectors, and at each interest point we extract a 128 dimensional SIFT feature vector [32]. Following [3] we concatenate the SIFT features with additional colour features comprising average and standard deviation of (R, G, B), (L, a, b) and $(r = R/(R+G+B)$, $g = G/(R+G+B))$, which gives an overall 144 dimensional feature vector.

We use $\mathbf{t_n}$ to denote the image label vector for image n with independent components $t_{nk} \in \{0, 1\}$ in which $k = 1, \ldots K$ labels the class. Each class can be present or absent independently in an image. $\mathbf{X_n}$ denotes the observation for image n and this comprises a set of J_n feature vectors $\{\mathbf{x_{nj}}\}$ where $j = 1, \ldots, J_n$. Note that the number J_n of detected interest points will in general vary from image to image.

On a small-scale problem it is reasonable to segment and label the objects present in the training images. However, for large-scale object recognition involving thousands of categories this will not be feasible, and so instead it is necessary to employ training data which is at best 'weakly labelled'. Here we consider a training set in which each image is labelled only according to the presence or absence of each category of object.

Next we associate with each patch j in each image n a binary label $\tau_{njk} \in \{0, 1\}$ denoting the class k of the patch. These labels are mutually exclusive, so that $\sum_{k=1}^{K} \tau_{njk} = 1$. These components can be grouped together into vectors $\tau_{\mathbf{nj}}$. If the values of these labels were available during training (corresponding to strongly labelled images) then the development of recognition models would be greatly simplified. For weakly labelled data, however, the $\{\tau_{\mathbf{nj}}\}$ labels are hidden (latent) variables, which of course makes the training problem much harder.

Consider for a moment a particular image n (and omit the index n to keep the notation uncluttered). We build a parametric model $y_k(\mathbf{x_j}, \mathbf{w})$ for the probability that patch $\mathbf{x_j}$ belongs to class k. For example we might use a simple linear-

softmax model with outputs

$$y_k(\mathbf{x_j}, \mathbf{w}) = \frac{\exp(\mathbf{w_k^T x_j})}{\sum_\mathbf{l} \exp(\mathbf{w_l^T x_j})}$$

which satisfy $0 \leqslant y_k \leqslant 1$ and $\sum_k y_k = 1$. The probability of a patch label $\tau_\mathbf{j}$ to be class k is then given directly by the output y_k:

$$p(\tau_\mathbf{j}|\mathbf{x_j}) = \prod_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{y_k}(\mathbf{x_j}, \mathbf{w})^{\tau_\mathbf{jk}}$$

Next we assume that if one, or more, of the patches carries the label for a particular class, then the whole image will. Thus the conditional distribution of the image label, given the patch labels, is given by

$$p(\mathbf{t}|\tau) = \prod_{\mathbf{k=1}}^{\mathbf{K}} \left[\mathbf{1} - \prod_{\mathbf{j=1}}^{\mathbf{J}} [\mathbf{1} - \tau_\mathbf{jk}] \right]^{\mathbf{t_k}} \left[\prod_{\mathbf{j=1}}^{\mathbf{J}} [\mathbf{1} - \tau_\mathbf{jk}] \right]^{\mathbf{1-t_k}}$$

In order to obtain the conditional distribution $p(\mathbf{t}|\mathbf{X})$ we have to marginalize over the latent patch labels. Although there are exponentially many terms in this sum, it can be performed analytically for our model to give

$$\begin{aligned} p(\mathbf{t}|\mathbf{X}) &= \sum_\tau \left\{ p(\mathbf{t}|\tau) \prod_{\mathbf{j=1}}^{\mathbf{J}} \mathbf{p}(\tau_\mathbf{j}|\mathbf{x_j}) \right\} \\ &= \prod_{k=1}^{K} \left[1 - \prod_{j=1}^{J} [1 - y_k(\mathbf{x_j}, \mathbf{w})] \right]^{t_k} \left[\prod_{j=1}^{J} [1 - y_k(\mathbf{x_j}, \mathbf{w})] \right]^{1-t_k} \end{aligned} \tag{4}$$

Given a training set of N images, which are assumed to be independent, we can construct the likelihood function from the product of such distributions, one for each data point. Taking the negative logarithm then gives the following error function

$$E(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} \{ t_{nk} \ln [1 - Z_{nk}] + (1 - t_{nk}) \ln Z_{nk} \}$$

where we have defined

$$Z_{nk} = \prod_{j=1}^{J_n} [1 - y_k(\mathbf{x_{nj}}, \mathbf{w})]$$

The parameter vector $\mathbf{w}$ can be determined by minimizing this error (which corresponds to maximizing the likelihood function) using a standard optimization algorithm such as scaled conjugate gradients [4]. More generally the likelihood function could be used as the basis of a Bayesian treatment, although we do not consider this here.

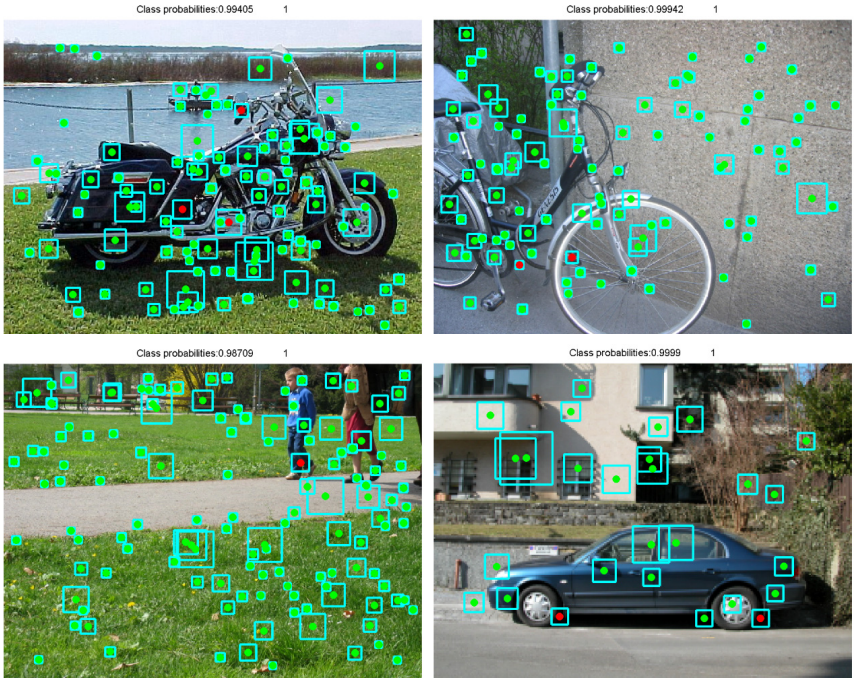


Fig. 14. *METU:* One patch labelling example for each class (motorbike, bike, people and car). Red and green dots denote foreground and background respectively. The patch labels are obtained by assigning each patch to the most probable class.

Once the optimal value $\mathbf{w}_{\mathrm{ML}}$ is found, the corresponding functions $y_k(\mathbf{x}, \mathbf{w}_{\mathrm{ML}})$ for $k = 1, \ldots, K$ will give the posterior class probabilities for a new patch feature vector $\mathbf{x}$. Thus the model has learned to label the patches even though the training data contained only image labels. Note, however, that as a consequence of the noisy 'OR' assumption, the model only needs to label one foreground patch correctly in order to predict the image label. It will therefore learn to pick out a small number of highly discriminative foreground patches, and will classify the remaining foreground patches, as well as those falling on the background, as 'background' meaning non-discriminative for the foreground class.

An example of patch labelling and image classification for each class is given in Figure 14.

6.11 MPITuebingen

Participants: Jan Eichhorn, Olivier Chapelle
Affiliation: Max Planck Institute for Biological Cybernetics, Tübingen, Germany
E-mail: {jan.eichhorn,olivier.chapelle}@tuebingen.mpg.de

Main Concepts For the absent/present object categorization task we used a Support Vector Classifier. Each image is converted to a collection of Local Image Descriptors (LIDs) and a kernel for sets is applied to this representation.

As LID we used the widely known SIFT-descriptors [32] but instead of the standard difference of Gaussians multiscale interest point detector we applied a basic Harris corner detector at one single scale. Each image was converted to a collection of LIDs where each LID contains coordinates, orientation and appearance of a particular salient region whose location was selected by the interest point detector (IPD). Note that, no data dependent post-processing of the LIDs was performed (as for example PCA or clustering in the appearance space of the training set).

For the successful use of Support Vector Classifiers it is necessary to define a kernel function appropriate for the underlying type of data representation. This function acts as a similarity measure (in our application for images) and should reflect basic assumptions about similarity in the categorization sense. For technical reasons it has to be a positive definite function.

To measure the similarity of two images, a possible strategy could be to find salient image regions of similar appearance (e.g. in SIFT-space) and thereby establishing a geometrical correspondence between the objects on the images (implicitly assuming that similar regions represent similar object parts).

In our method we avoid the complications of finding correspondences between images and neglect all geometry information at scales larger than the size of the extracted salient regions. In practice this means we ignore the coordinates of the LID and simply use its appearance part. Consequently the representation of a single image is reduced to a set of appearance vectors. On top of this representation we can now apply a kernel function for sets, the so called Bhattacharyya kernel [25]. Details of this kernel are described in the following Section. As a minor kernel we always used a standard Gaussian RBF-kernel $k_{\mathrm{RBF}}(x, x') = \exp(-\frac{\|x-x'\|}{2\sigma^2})$.

Maybe it is interesting to note that we observed a decreasing performance when using during training the segmentation mask of the objects that was provided with the datasets. This behaviour might indicate that the method can use information from the image background to infer the absence or presence of an object. In case of more realistic datasets and true multi-class categorization this effect should vanish.

Bhattacharyya Kernel [25]. The name of this kernel function arises from the fact that it is based on the Bhattacharyya affinity, a similarity measure that is defined for probability distributions:

$$k_{\text{bhatt}}(p, p') = \int \sqrt{p(x) \cdot p'(x)}\, \mathrm{d}x$$

To define a kernel function between two sets $\mathbf{L}$ and $\mathbf{L}'$, it was suggested in [25] to fit a Gaussian distribution to each of the sets: $\mathbf{L} \sim \mathcal{N}(\mu, \Sigma)$ and $\mathbf{L}' \sim \mathcal{N}(\mu', \Sigma')$. Then, the value of the *Bhattacharyya kernel* is the Bhattacharyya affinity of the corresponding Gaussians, which can be formulated as a closed expression

$$\begin{aligned} K_{\text{bhatt}}(\mathbf{L}, \mathbf{L}') =& |\Sigma|^{-\frac{1}{4}} \, |\Sigma'|^{-\frac{1}{4}} \, |\Sigma^\dagger|^{\frac{1}{2}} \\ & \exp\left[-\frac{1}{4} \left(\mu^\top \Sigma^{-1} \mu + \mu'^\top \Sigma'^{-1} \mu' \right) + \frac{1}{2} \mu^{\dagger\top} \Sigma^\dagger \mu^\dagger \right] \end{aligned}$$

where $\Sigma^\dagger = 2\left(\Sigma^{-1} + \Sigma'^{-1}\right)^{-1}$ and $\mu^\dagger = \frac{1}{2}\left(\Sigma^{-1}\mu + \Sigma'^{-1}\mu'\right)$.

Since the Gaussian approximation reflects only a limited part of the statistics of the empirical distribution, the authors further propose to map the set elements into a feature space induced by a minor kernel. The Bhattacharyya affinity can be computed in feature space by use of the kernel trick and doing so allows to capture more structure of the empirical distribution. However, in feature space the covariance matrices of each of the sets (Σ and Σ' respectively) are structurally rank-deficient and therefore it is necessary to involve a regularization step before computing the inverse:

$$\tilde{\Sigma}^{-1} = (\Sigma + \eta \cdot \mathrm{Tr}(\Sigma) \cdot I)^{-1}$$

Hereby a new parameter η is introduced, which adjusts the amount of regularization. The larger η is the more similar the two covariance matrices appear and the more the kernel depends only on the difference of the set means[17].

A more detailed analysis of other kernel functions for images represented by LIDs is under review for publication. A preliminary version can be found in a technical report [16].

6.12 Southampton

Participants: Jason D. R. Farquhar, Hongying Meng, Sandor Szedmak, John Shawe-Taylor
Affiliation: University of Southampton, Southampton, UK
E-mail: ss03v@ecs.soton.ac.uk

[17] If the covariance matrices are identical ($\Sigma = \Sigma'$) the Bhattacharyya kernel reduces to: $K_{\text{bhatt}}(\mathbf{L}, \mathbf{L}') = \exp\left(-\frac{1}{4}(\mu - \mu')^\top \Sigma^{-1} (\mu - \mu')\right)$

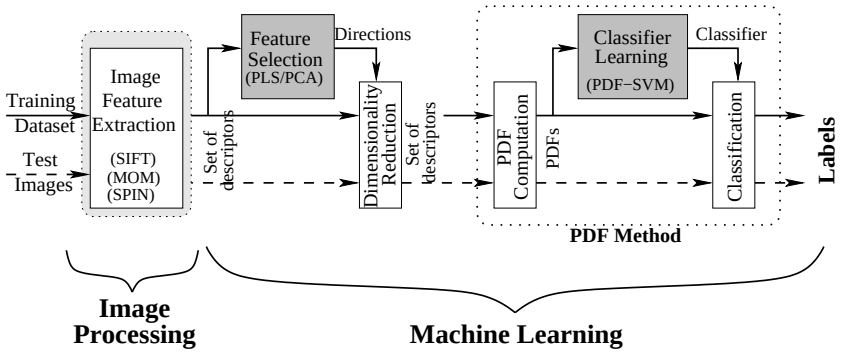


Fig. 15. *Southampton:* General classification schema

Introduction. Our method consists of two main phases, as shown in Figure 15, 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 innovations: 1) the Bhattacharyya kernel is used to measure the similarity of the sets of local features found in each image, and, 2) an extension of the well-known SVM, called SVM_2K, is used to combine different features and improve overall performance. Each of the main components of our approach is described next.

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). The types of the detectors used were, Multi-scale Harris-Affine, and Laplacian of Gaussians (LoG). To reduce the feature dimension and increase robustness to common image transformations (such as illumination or perspective) a local feature is generated for each patch using the *SIFT* descriptor. For more details see [32] or [36].

Dimensionality Reduction. As the dimension of the SIFTs is relatively high dimensionality reduction is used to improve the generalisation capability of the learning procedure and diminish the overall training time. The two types of dimensionality reduction tried are: Principal Component analysis (PCA), which finds directions of maximum variance in the input data, and Partial Least Squares Regression (PLS) [2], which finds directions of maximum covariance between input data and output labels.

PDF Computation. Image feature generation and dimensionality reduction output a set of local descriptors per image. As most machine learning algorithms cannot cope with variable length feature sets, previously histograms have been used to map these to a fixed length representation. An alternative approach is to model the set of features as a probability distribution (PDF) over feature

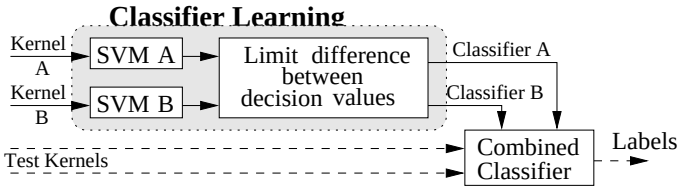


Fig. 16. *Southampton:* SVM_2K combines feature vectors arriving from distinct sources.

space and then define a kernel between PDFs, which can be used in any kernelised learning algorithm, such as the SVM. We assumed that the set of image features follow Gaussian distribution and then the Bhattacharyya kernel [25], $K(\Pr_1(x), \Pr_2(x)) = \int \sqrt{\Pr_1(x)}\sqrt{\Pr_2(x)}dx$, was used to measure similarity of them.

Classifier Learning. To date only maximum margin based classifiers have been used, specifically either a conventional SVM [42] or our modified multi-feature SVM, called SVM_2K. As shown in Figure 16, SVM_2K combines two distinct feature sources (or kernels) to maximise the output classifier performance. The details can be found in [33].

Experiments. Three learning methodologies within the framework outlined above were submitted which differed only in the types of interest point detector used (LoG or multi-scale Harris affine) and the classifier used (SVM or SVM_2K), with both LoG and Harris-Affine features used for SVM_2K. From initial experiments it was found that a 20 dimensional PLS reduction gave best performance so this was applied in all cases.

7 Results: Classification Task

7.1 Competition 1: `test1`

Table 8 lists the results of classification competition 1. In this competition, training was carried out using only the `train+val` image set, and testing performed on the `test1` image set. For each object class and submission, the EER and AUC measures are listed. Some participants submitted multiple results, and results for all submissions are shown. The ROC curves for the competition are shown in Figures 18–21, with each figure showing the results for a particular object class. In these figures, only the "best" result submitted by each participant is shown to aid clarity; the EER measure was used to choose the best result for each participant.

The INRIA-Jurie method performed consistently best in terms of both EER and AUC, achieving EER of 0.917–0.977 depending on the class. This method

Table 8. Results for competition 1: *classification*, train using the `train+val` image set and test on the `test1` image set. For each object class and submission, the EER and AUC measures are shown. Note that some participants submitted multiple results. Bold entries in each column denote the "best" methods for that object class according to EER or AUC.

	Motorbikes		**Bicycles**		**People**		**Cars**	
Submission	EER	AUC	EER	AUC	EER	AUC	EER	AUC
Aachen: ms-2048-histo	0.926	0.979	0.842	0.931	0.861	0.928	0.925	0.978
Aachen: n1st-1024	0.940	0.987	0.868	0.954	0.861	0.936	0.920	0.979
Darmstadt: ISM	0.829	0.919	–	–	–	–	0.548	0.578
Darmstadt: ISMSVM	0.856	0.882	–	–	–	–	0.644	0.717
Edinburgh: bof	0.722	0.765	0.689	0.724	0.571	0.597	0.793	0.798
HUT: final1	0.921	0.974	0.795	0.891	0.850	0.927	0.869	0.956
HUT: final2	0.917	0.970	0.816	0.895	0.833	0.931	0.908	0.968
HUT: final3	0.912	0.952	0.781	0.864	0.845	0.919	0.847	0.934
HUT: final4	0.898	0.960	0.767	0.880	0.857	0.921	0.909	0.971
INRIA-Jurie: dcb_p1	0.968	0.997	0.918	0.974	**0.917**	**0.979**	**0.961**	**0.992**
INRIA-Jurie: dcb_p2	**0.977**	**0.998**	**0.930**	0.981	0.901	0.965	0.938	0.987
INRIA-Zhang	0.964	0.996	**0.930**	**0.982**	**0.917**	0.972	0.937	0.983
METU	0.903	0.966	0.781	0.822	0.803	0.816	0.840	0.920
MPITuebingen	0.875	0.945	0.754	0.838	0.731	0.834	0.831	0.918
Southampton: develtest	0.972	0.994	0.895	0.961	0.881	0.943	0.913	0.972
Southampton: LoG	0.949	0.989	0.868	0.943	0.833	0.918	0.898	0.959
Southampton: mhar.aff	0.940	0.985	0.851	0.930	0.841	0.925	0.901	0.961

uses the "bag of words" representation with local descriptors extracted at points on a dense grid. Performance of the INRIA-Zhang method was very similar; this method also uses the bag of words representation, but uses interest point detection to extract a sparser set of local features. For three of the classes, the ROC curves for the two methods intersect several times, making it impossible to determine which method performs best overall; only for the "cars" class was the performance of the INRIA-Jurie method consistently better over the whole range of the ROC curve (Figure 21).

Performance of two of the other methods using distributions of local features: Aachen and Southampton, was also similar but typically slightly worse than the INRIA methods, though the Southampton method performed particularly well on the "motorbikes" class. The Aachen method uses a log-linear model for classification, and the Southampton method the Bhattacharyya kernel instead of the bag of words representation.

The MPITuebingen method, which is similar to the Southampton method in the use of the Bhattacharyya kernel had consistently lower performance; reasons might include differences in the method for extraction of local features. The Edinburgh method, which is very similar to the INRIA-Zhang method gave consistently worse results; Section 6.3 discusses the likely reasons for this.

Table 9. Results for competition 2: *classification*, train using the `train+val` image set and test on the `test2` image set. For each object class and submission, the EER and AUC measures are shown. Note that some participants submitted multiple results. Bold entries in each column denote the "best" methods for that object class according to EER or AUC.

	Motorbikes		**Bicycles**		**People**		**Cars**	
Submission	EER	AUC	EER	AUC	EER	AUC	EER	AUC
Aachen: ms-2048-histo	0.767	0.825	0.667	0.724	0.663	0.721	0.703	0.767
Aachen: n1st-1024	0.769	0.829	0.665	0.729	0.669	0.739	0.716	0.780
Darmstadt: ISM	0.663	0.706	–	–	–	–	0.551	0.572
Darmstadt: ISMSVM	0.683	0.716	–	–	–	–	0.658	0.683
Edinburgh: bof	0.698	0.710	0.575	0.606	0.519	0.552	0.633	0.655
HUT: final1	0.614	0.666	0.527	0.567	0.601	0.650	0.655	0.709
HUT: final2	0.624	0.693	0.604	0.647	0.614	0.661	0.676	0.740
HUT: final3	0.594	0.637	0.524	0.546	0.574	0.618	0.644	0.694
HUT: final4	0.635	0.675	0.616	0.645	0.587	0.630	0.692	0.744
INRIA-Zhang	**0.798**	**0.865**	**0.728**	**0.813**	**0.719**	**0.798**	**0.720**	**0.802**
MPITuebingen	0.698	0.765	0.616	0.654	0.591	0.655	0.677	0.717

The HUT method, which is based on segmented image regions, performed comparably to the methods based on local features for all but the "bicycles" class; the poorer performance on this class might be anticipated because of the difficulty of segmenting a bicycle from the background. The METU method, based on individual local descriptors, performed worse than the methods using the global distribution of local features except on the "motorbikes" class.

The "recognition by detection" method submitted by Darmstadt did not perform well on this competition. Darmstadt chose to train only using side views, and this will have limited the performance. Another possible reason is that there was correlation between the object class presence and the appearance of the background, which this method is unable to exploit.

7.2 Competition 2: `test2`

Table 9 lists the results of classification competition 2. In this competition, training was carried out using only the `train+val` image set, and testing performed on the `test2` image set. The ROC curves for the competition are shown in Figures 22–25, with each figure showing the results for a particular object class. Only the "best" result submitted by each participant is shown to aid clarity; the EER measure was used to choose the best result for each participant.

Fewer participants submitted results for competition 2 than competition 1. The best results were obtained by the INRIA-Zhang method both in terms of EER and AUC, and for all object classes; this method also performed close to best in competition 1.

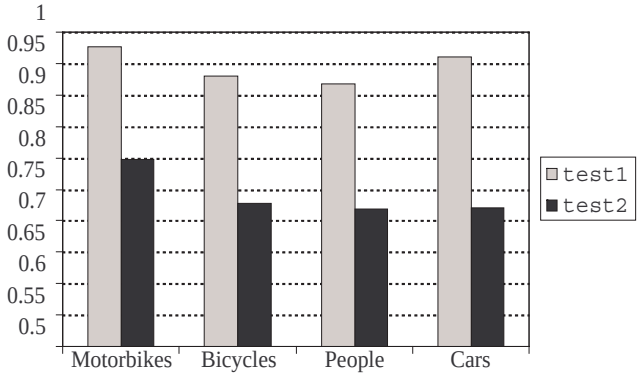


Fig. 17. Equal error rate (EER) results for *classification* competitions 1 and 2 by class and test set. The "best" (in terms of EER) result obtained for each class and each test set is shown. Note that results were much better for the `test1` set than for the `test2` set. There is perhaps surprisingly little difference in performance across classes.

The Aachen method performed similarly to the INRIA-Zhang method in competition 2, and better relative to the other methods than it did in competition 1; it may be that this method offers more generalization which is helpful on the more variable images in the `test2` image set, but not for `test1`.

Performances of the Edinburgh, HUT, and MPITuebingen methods were all similar but varied over the object classes. The poorer performance of the Edinburgh method on the `test1` images was not clear for the `test2` images.

In competition 2, the performance of the Darmstadt method was comparable to the others, whereas it performed poorly in competition 1. It may be that in the `test2` dataset there are fewer regularities in the image context of the object classes, so methods such as the Darmstadt one, which ignore the background, are more effective.

7.3 Comparison of Competitions 1 and 2

Figure 17 shows the best EER obtained for each object class in competition 1 (`test1`) and competition 2 (`test2`). The `test2` image set seems to be much more challenging; this was the intention in collecting this set of images. In terms of EER, performance on the `test2` images were worse than on the `test1` images, with EER in the range 0.720–0.798 for the best method, depending on the object class, compared to 0.917–0.977 for competition 1. Recall that this second test set was intended to provide a set of images with higher variability than those in the first image set; it seems that this intention has been met.

There is surprisingly little difference in performance of the best methods across object classes, using either the `test1` or `test2` image sets. While performance on the two object classes that the object recognition community might

typically consider 'easier': motorbikes and cars, was indeed better than for the other two classes on the `test1` image set, the differences to the other classes are small. One might expect recognition of bicycles to be much harder because of their "wiry" structure which makes segmentation from the background difficult, or means that local features will a contain significant area of background; humans might be considered difficult to recognize because of the high variability of shape (e.g. different poses) and appearance (clothes, etc.). It is not possible to offer a conclusive explanation of the results here; one possibility is unintended regularity in the background giving a strong cue to the object class. Because none of the methods used here except the Darmstadt method delineate the object(s) in the image which result in a positive classification, it is hard to tell which parts of the image are being used by the classifier.

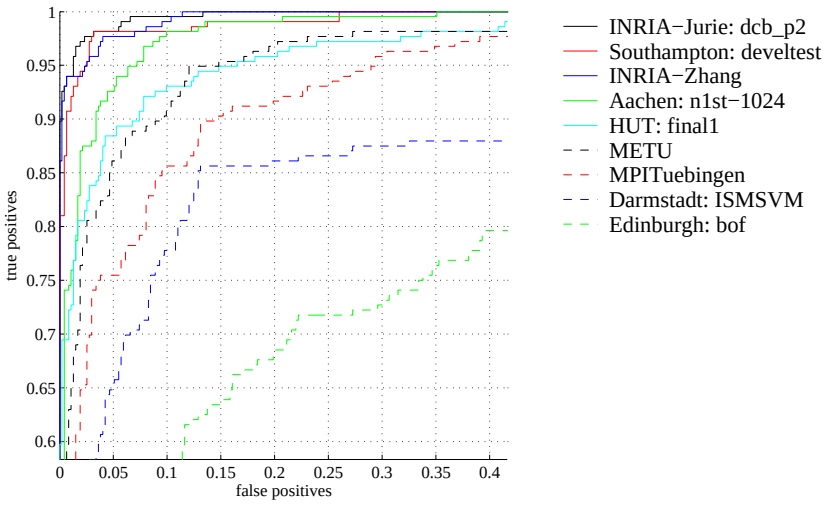


Fig. 18. ROC curves for *motorbikes* in competition 1: classification, train using the `train+val` image set and test on the `test1` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

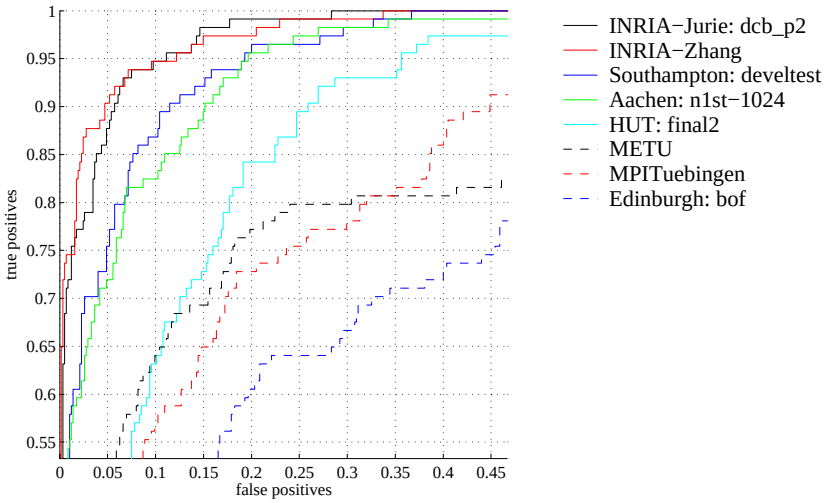


Fig. 19. ROC curves for *bicycles* in competition 1: classification, train using the `train+val` image set and test on the `test1` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

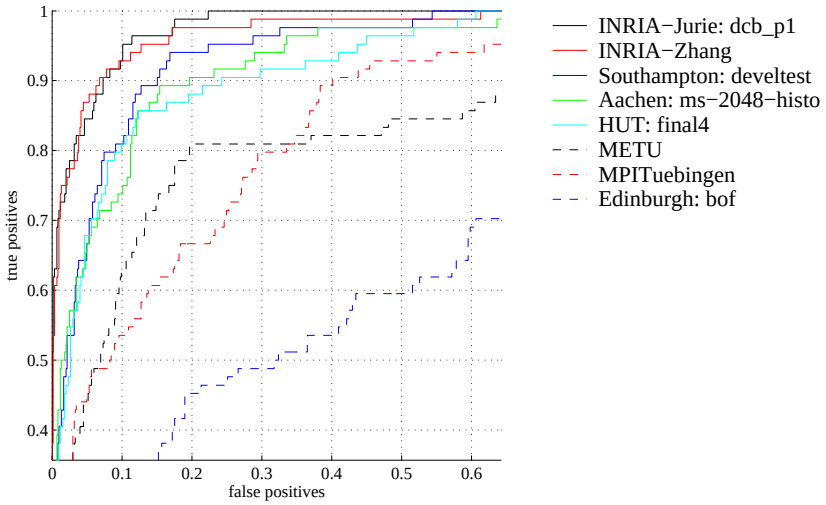


Fig. 20. ROC curves for *people* in competition 1: classification, train using the `train+val` image set and test on the `test1` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

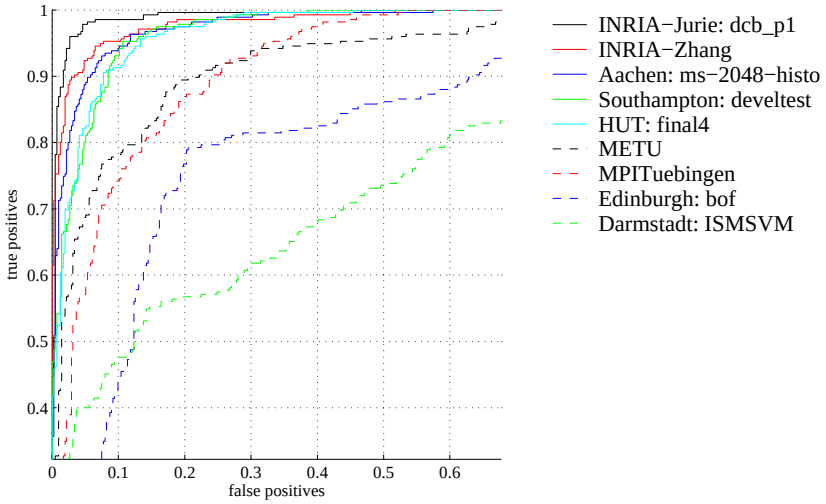


Fig. 21. ROC curves for *cars* in competition 1: classification, train using the `train+val` image set and test on the `test1` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

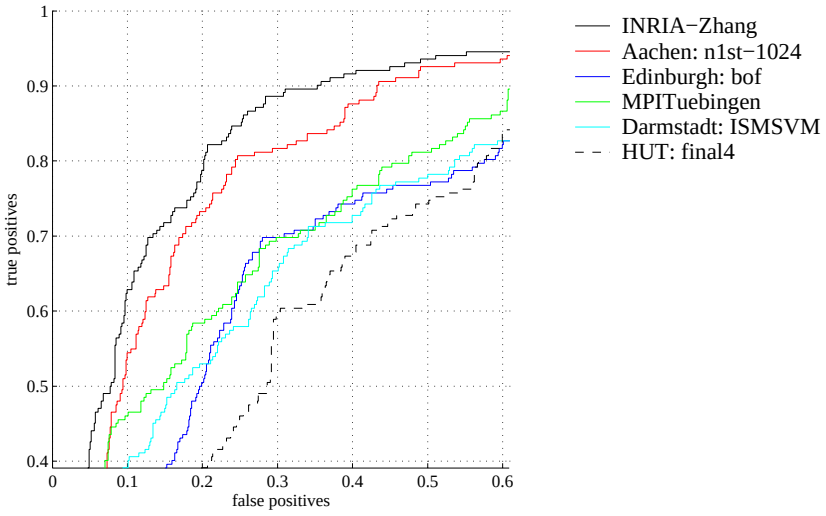


Fig. 22. ROC curves for *motorbikes* in competition 2: classification, train using the `train+val` image set and test on the `test2` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

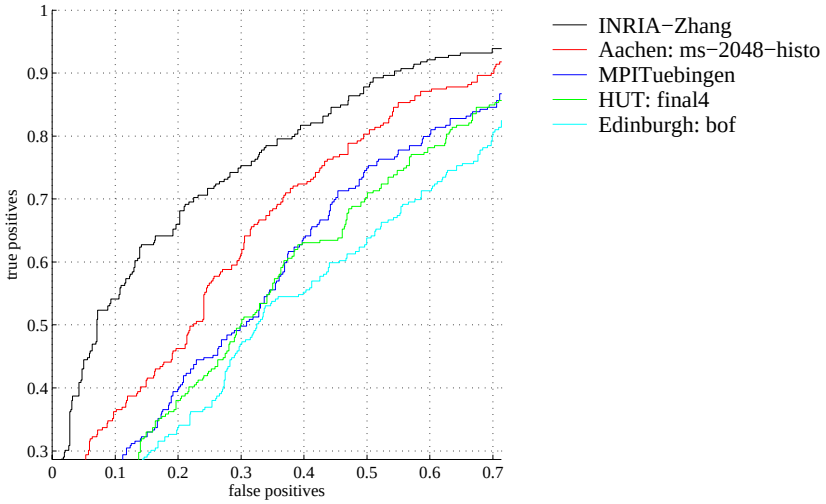


Fig. 23. ROC curves for *bicycles* in competition 2: classification, train using the `train+val` image set and test on the `test2` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

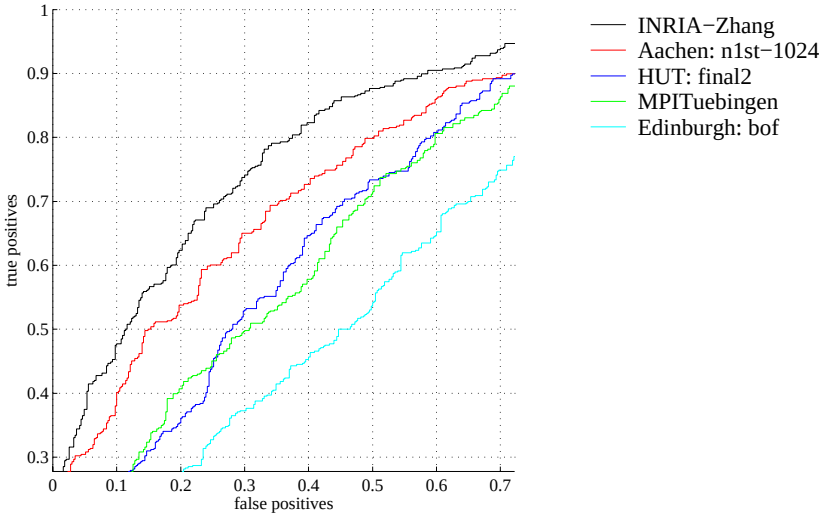


Fig. 24. ROC curves for *people* in competition 2: classification, train using the `train+val` image set and test on the `test2` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

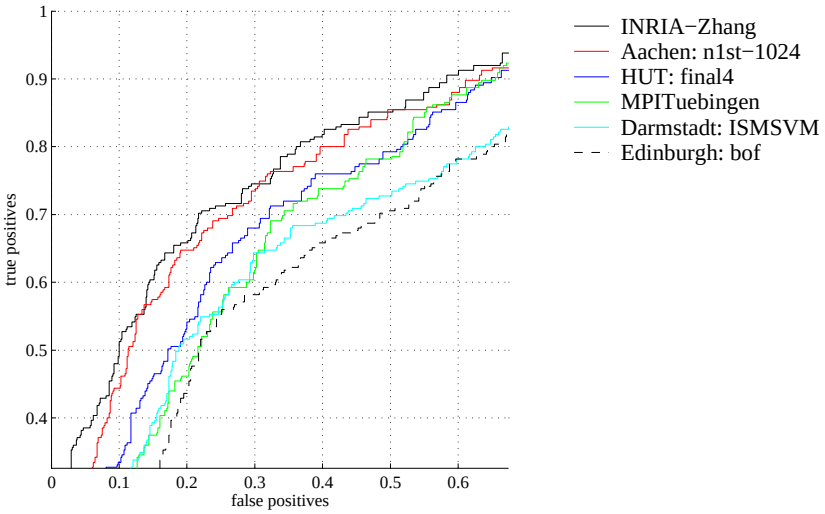


Fig. 25. ROC curves for *cars* in competition 2: classification, train using the `train+val` image set and test on the `test2` image set. The best result in terms of EER from each participant is shown, with curves ranked by decreasing EER. The axes cover a range equal to two times the maximum EER of the submitted results.

Table 10. Results for competition 5: *detection*, train using the `train+val` image set and test on the `test1` image set. For each object class and submission, the AP measure is shown. Note that some participants submitted multiple results. Bold entries in each column denote the "best" methods for that object class according to AP.

Submission	Motorbikes	Bicycles	People	Cars
Darmstadt: ISM	0.865	–	–	0.468
Darmstadt: ISMSVM	**0.886**	–	–	0.489
Darmstadt: ISMSVM_2	–	–	–	0.439
Edinburgh: meanbb	0.216	0.007	0.000	0.000
Edinburgh: puritymeanbb	0.470	0.015	0.000	0.000
Edinburgh: siftbb	0.453	0.098	0.002	0.000
Edinburgh: wholeimage	0.118	**0.119**	0.000	0.000
FranceTelecom	0.729	–	–	0.353
INRIA-Dalal	0.490	–	**0.013**	**0.613**
INRIA-Dorko	0.598	–	0.000	–

8 Results: Detection Task

8.1 Competition 5: `test1`

Table 10 lists the results of detection competition 5. In this competition, training was carried out using only the `train+val` image set, and testing performed on the `test1` image set. For each object class and submission, the AP measure is listed. Some participants submitted multiple results, and results for all submissions are shown. The precision/recall curves for the competition are shown in Figures 27–30, with each figure showing the results for a particular object class.

Performance of methods on the detection tasks varied much more greatly than for the classification task, and there were fewer submissions. For the "motorbikes" class, the Darmstadt method performed convincingly better than other methods, with an average precision of 0.886. The variant using an SVM verification stage (ISMSVM) was slightly better than that without it (ISM). The FranceTelecom method also performed well on this class, giving good precision across all recall levels, but was consistently outperformed by the Darmstadt method. The INRIA-Dorko method, which is a variant of the Darmstadt method, performed only slightly worse than the Darmstadt method at low recall, but precision dropped off sharply at recall above 0.5. The Darmstadt submission used segmentation masks for training, while the INRIA-Dorko method used only the bounding boxes, and this may account for the difference in results.

The INRIA-Dalal method performed significantly worse for motorbikes than the other methods. Section 6.6 reports improved results by modifying the window size used by the detector. In the challenge, performance was close to the better of the baseline methods provided by Edinburgh. These baseline methods used the bag of words classifier to assign confidence to detections and predicted a single bounding box either simply as the mean bounding box taken from the training

data, or as the bounding box of all Harris points in the image; the difference in performance between these two methods was small. The success of these simple methods can be attributed to the lack of variability in the `test1` data: many of the motorbikes appear in the centre of the image against a fairly uniform background.

For the "bicycles" class, the only results submitted were for Edinburgh's baseline methods. The method predicting the bounding box as the bounding box of all Harris points did best, suggesting that uniform background may have been the reason.

For the "people" class, INRIA-Dalal, INRIA-Dorko and Edinburgh submitted results. The INRIA-Dalal method performed best but AP was very low at 0.013. The INRIA-Dorko method and baselines all gave almost zero average precision. The poor results on this task may be attributed to the small size of the training set relative to the large variability in appearance of people.

For the "cars" class, the INRIA-Dalal method achieved the highest AP of 0.304. For recall below 0.5, the Darmstadt method also performed well, with the ISMSVM_2 run giving greater precision than the INRIA-Dalal method; precision dropped off sharply at higher levels of recall. Darmstadt chose to train only on *side* views of cars and this explains the drop off in precision as the method fails to find cars from other views.

The FranceTelecom method did not perform as well as the INRIA-Dalal or Darmstadt methods, but was consistently much better than any of the Edinburgh baselines. The failure of the baselines suggests that the car images exhibited much less regularity than the motorbike images.

8.2 Competition 6: `test2`

Table 11 lists the results of detection competition 6. In this competition, training was carried out using only the `train+val` image set, and testing performed on the `test2` image set. The precision/recall curves for the competition are shown in Figures 31–34, with each figure showing the results for a particular object class.

Overall performance on the `test2` image set was much worse than on the `test1` images. The best results were obtained for motorbikes, and for this class AP dropped from 0.886 on `test1` to 0.341 on `test2`.

The relative performance of the methods was largely unchanged from that observed in competition 5. For motorbikes, the Darmstadt method performed best, and for cars the INRIA-Dalal method. For the "cars" class, the INRIA-Dalal method performed convincingly better than the Darmstadt method, which achieved high precision but lower recall in competition 5. The reason for this may be that the `test2` images contain an even lower proportion of side views of cars than in the `test1` data.

The FranceTelecom method also gave results well above the baselines for the "motorbikes" class, but results for the "cars" class were poor, with precision dropping off at very low recall.

Table 11. Results for competition 6: *detection*, train using the `train+val` image set and test on the `test2` image set. For each object class and submission, the AP measure is shown. Note that some participants submitted multiple results. Bold entries in each column denote the "best" methods for that object class according to AP.

Submission	Motorbikes	Bicycles	People	Cars
Darmstadt: ISM	0.292	–	–	0.083
Darmstadt: ISMSVM	0.300	–	–	0.181
Darmstadt: ISMSVM_2	**0.341**	–	–	–
Edinburgh: meanbb	0.055	0.000	0.000	0.000
Edinburgh: puritymeanbb	0.116	0.004	0.000	0.000
Edinburgh: siftbb	0.088	**0.113**	0.000	0.028
Edinburgh: wholeimage	0.020	0.006	0.000	0.005
FranceTelecom	0.289	–	–	0.106
INRIA-Dalal	0.124	–	**0.021**	**0.304**

For the "people" class, only INRIA-Dalal and Edinburgh submitted results. The precision/recall curve of the INRIA-Dalal method was consistently above any of the baseline methods, but AP was very low at 0.021; this is probably due to the limited training data.

For all classes except people the Edinburgh baselines did surprisingly well, though consistently worse than the other methods. In particular, the method proposing the bounding box of all Harris points gave good results. This suggests that there may still be a significant bias toward objects appearing on a uniform background in the `test2` images.

8.3 Comparison of Competitions 5 and 6

Figure 26 shows the best AP obtained for each object class in competition 5 (`test1`) and competition 6 (`test2`). For the "motorbikes" and "cars" classes, for which results significantly better than the baselines were achieved, results on `test1` were much better than on `test2` suggesting that the second test set is indeed much more challenging, as was the intention.

Performance across the object classes varied greatly on both test sets. Note however that for bicycles only results for "baseline" methods were submitted, and for people results for only two methods were submitted for `test1`, and only one method for `test2`.

For the `test1` images, performance for motorbikes was better than that for cars, which is interesting since one might expect cars to be easier to recognize because of their more convex structure. The reason may be due to less variation in the pose of motorbikes (mostly side views) relative to cars in the `test1` images. Results on the two classes for the `test2` images were about equal, suggesting that there is less bias in the second test set.

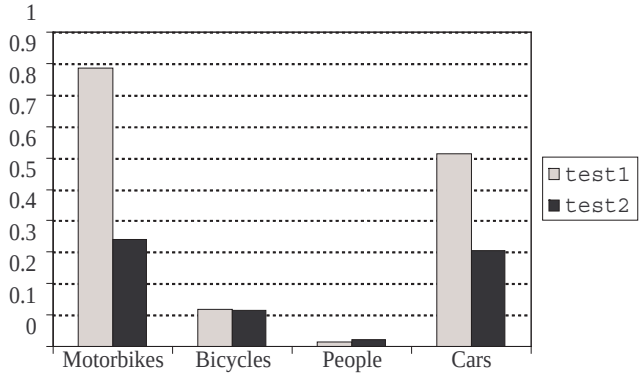


Fig. 26. Average precision (AP) results for *detection* competitions 5 and 6 by class and test set. The "best" (in terms of AP) result obtained for each class and each test set is shown. For the motorbike and car classes the results were much better for the `test1` set than for the `test2` set. There is a large difference in performance across classes; however note that few groups submitted results for bicycles and people.

8.4 Competitions 7 and 8

Competitions 7 and 8 allowed participants to use any training data other than the test data provided for the challenge. Only one participant submitted results: INRIA-Dalal tackled the "people" class on both test sets. Figure 35 shows precision/recall curves for these results. Average precision was 0.410 for the `test1` images, and 0.438 for the `test2` images. These results are strikingly different than those obtained using the same method but only the provided training data: AP of 0.013 for `test1` and 0.021 for `test2`. This suggests that, certainly for this method, the size of the training set provided for the "people" class was inadequate.

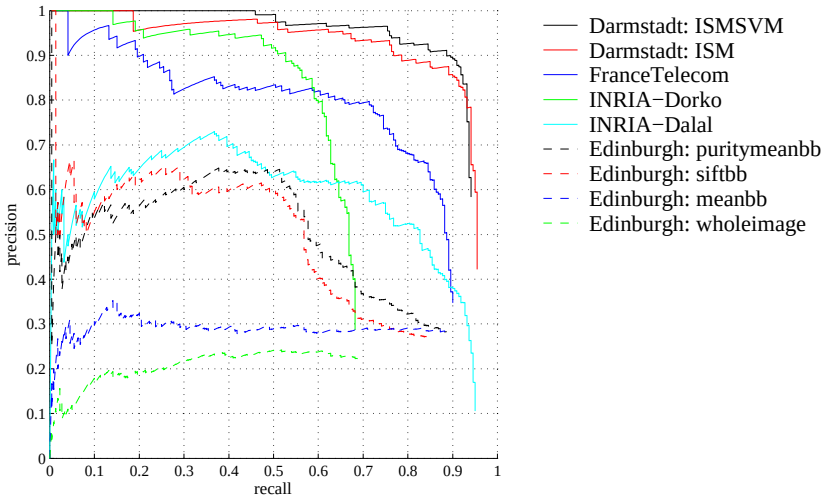


Fig. 27. PR curves for *motorbikes* in competition 5: detection, train using the `train+val` image set and test on the `test1` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.

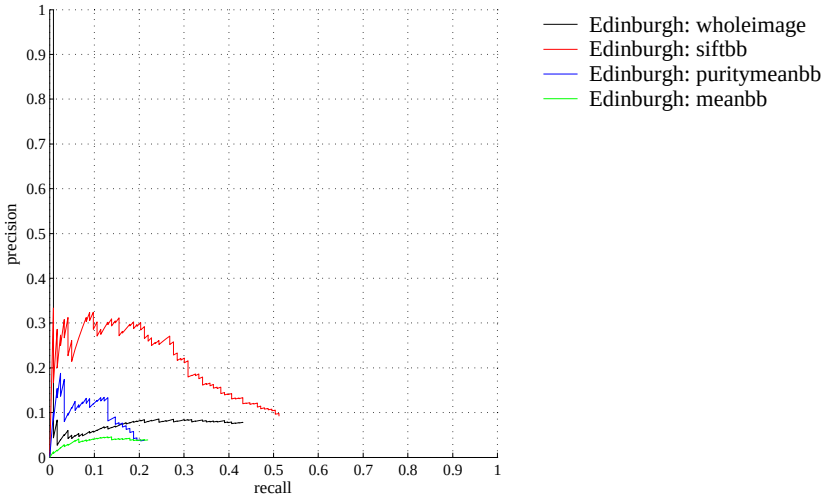


Fig. 28. PR curves for *bicycles* in competition 5: detection, train using the `train+val` image set and test on the `test1` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.

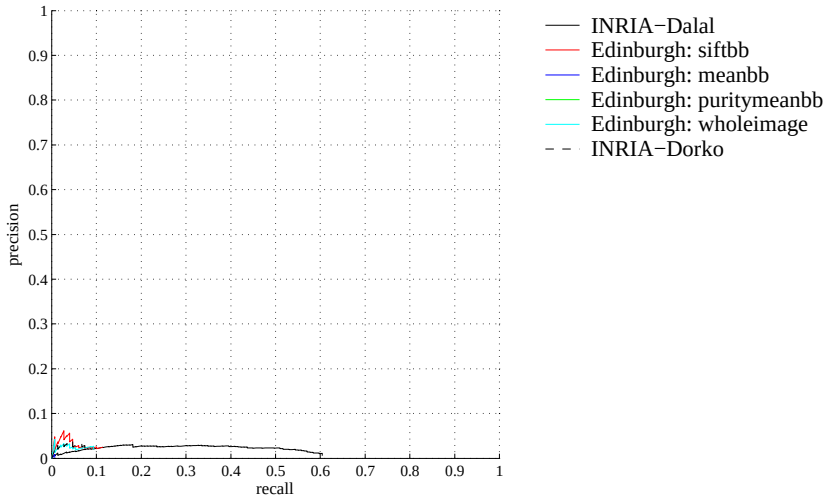


Fig. 29. PR curves for *people* in competition 5: detection, train using the `train+val` image set and test on the `test1` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.

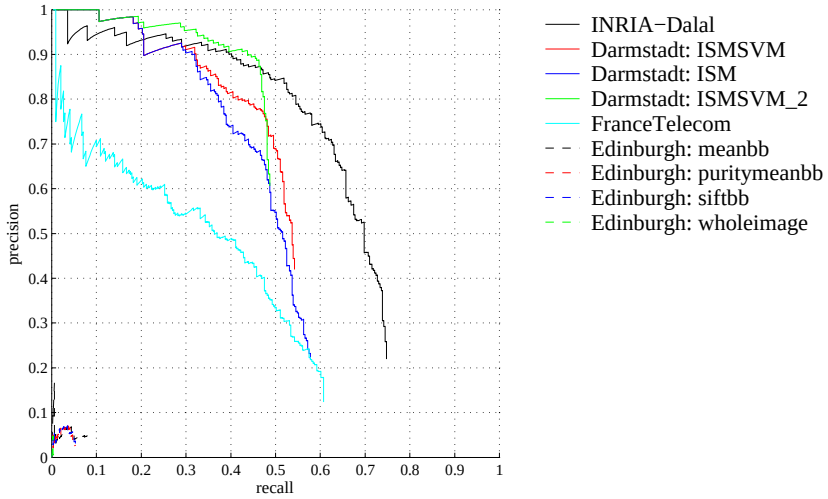


Fig. 30. PR curves for *cars* in competition 5: detection, train using the `train+val` image set and test on the `test1` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.

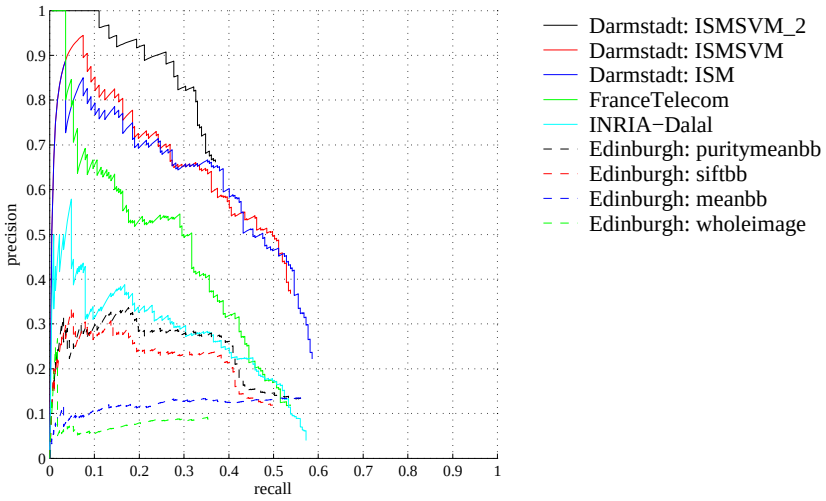


Fig. 31. PR curves for *motorbikes* in competition 6: detection, train using the `train+val` image set and test on the `test2` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.

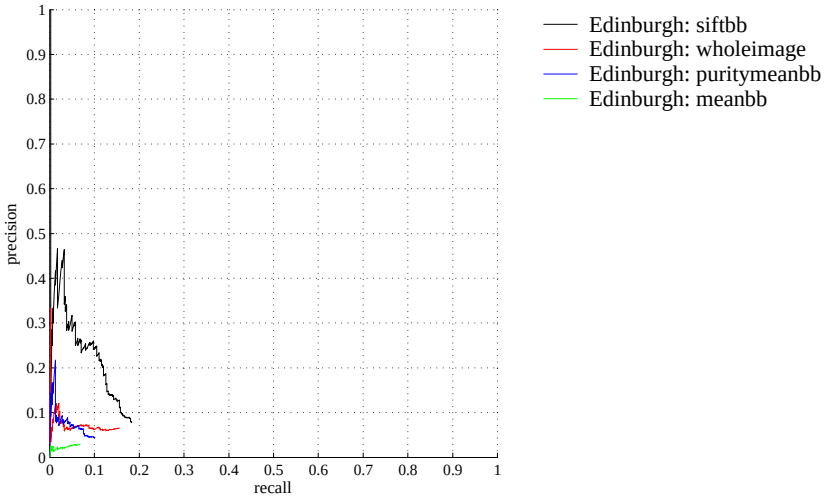


Fig. 32. PR curves for *bicycles* in competition 6: detection, train using the `train+val` image set and test on the `test2` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.

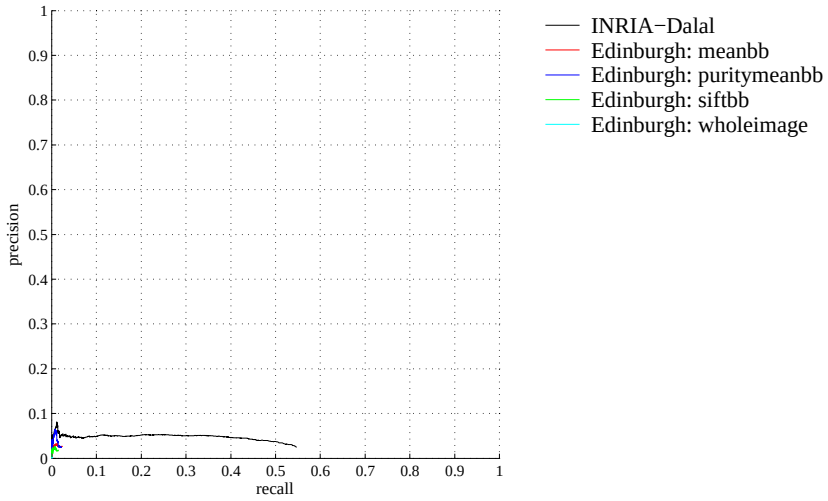


Fig. 33. PR curves for *people* in competition 6: detection, train using the `train+val` image set and test on the `test2` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.

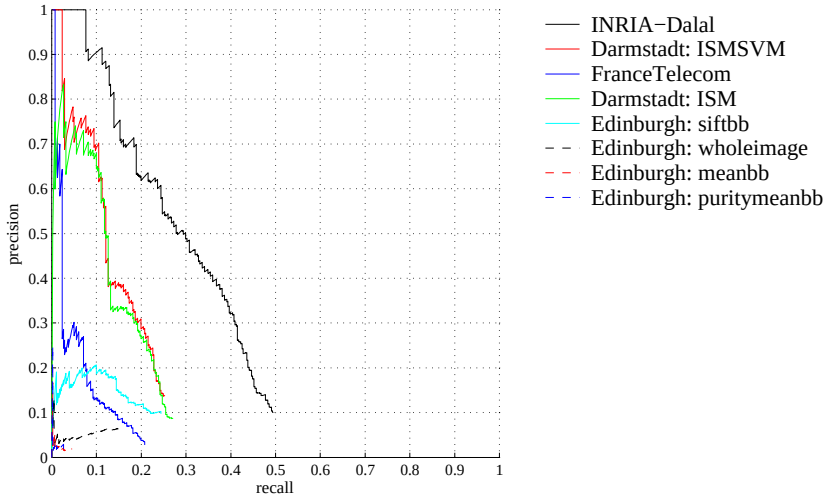
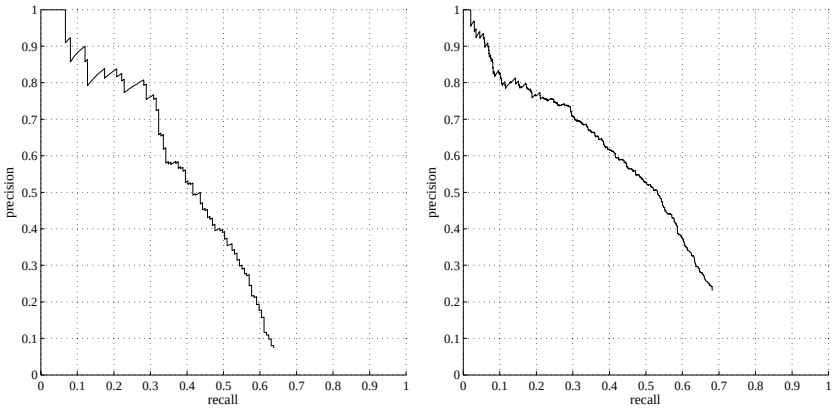


Fig. 34. PR curves for *cars* in competition 6: detection, train using the `train+val` image set and test on the `test2` image set. All results submitted by each participant are shown, with curves ranked by decreasing AP.



(a) `test1` (b) `test2`

Fig. 35. PR curves for *people* in competitions 7 and 8: detection, train using any data other than the provided test sets. Results shown are for the sole submission, from INRIA-Dalal.

9 Discussion

The challenge proved a worthwhile endeavour, with participation from twelve groups representing nine institutions. A range of methods for object classification and detection were evaluated providing a valuable snapshot of the state of the art in these tasks. The experience gained in the challenge, and discussion at the challenge workshop, resulted in a number of issues for consideration in future challenges.

Errors in the Data. Several participants commented on errors in the provided data. Undoubtedly there remained some errors in the first data set which arose from incomplete labelling of the original image databases from which the images were taken. Future challenges should improve the quality of this data. In terms of evaluation, all participants used the same data so any errors should not have caused bias toward a particular method. A key aspect of machine learning is the ability of a learning method to cope with some proportion of errors in the training data. It might be interesting for future challenges to consider data sets with known and varying proportion of errors to test methods' robustness to such errors.

A related issue is the difficulty of establishing ground truth, particularly for the detection task. For many images it is hard for a human observer to judge whether a particular object is really recognizable, or to segment individual objects, for example a number of bicycles in a bike rack. A unique aspect of the

challenge was that the images were collected without reference to a particular method, whereas many databases will have been collected and annotated with a particular approach e.g. window-based or parts-based in mind. Future challenges might employ multiple annotations of the same images to allow some consensus to be reached, or increasing the size of the datasets might reduce the effect of such ambiguity on evaluation results. The results of existing methods might also be used to judge the "difficulty" of each image.

Limited Training Data. One participant commented that the training data provided for the person detection task was insufficient. However, there is a move in the object recognition community toward use of small training sets, as little as tens of images for some object classes, so there is some value in testing results with small training sets. Future challenges might consider providing larger training sets.

Difficulty Level of the Data. One participant commented that the `train+val` data was too "easy" with respect to the test data. Images were assigned randomly to the `train`, `val`, and `test1` image sets, so the training data should have been unbiased with respect to `test1`. It is quite possible that for current methods, the `train+val` data was not sufficient to learn a method successful on `test2` images. This is more a comment of current methods than the data itself, for example most current methods are "view-based" and require training on different views of an object; other methods might not have such requirements.

Releasing Test Data. In the challenge, the test data with associated ground truth was released to participants. Code to compute the ROC and PR curves was given to participants and the computed curves were returned to the organizers. This protocol was followed to minimize the burden on both participants and organizers, however, because the participants had access to the ground truth of the test sets, there was a risk that participants might optimize their methods on the test sets.

It was suggested that for future challenges the test data and/or ground truth not be released to participants. This gives two alternatives: (i) release images but not ground truth. One problem here is that participants may informally generate their own ground truth by "eye-balling" their results (this is much less of a problem in most machine learning contests, where it is hard for humans to generate predictions based on the input features); (ii) release no test data. This would require that participants submit binaries or source code to the organizers who would run it on the test data. This option was not taken for the challenge because of anticipated problems in running participants' code developed on different operating systems, with different shared libraries, etc. Submitting source code e.g. MATLAB code would also raise issues of confidentiality.

Evaluation Methods. Some participants were concerned that the evaluation measures (EER, AUC, AP) were not defined before results were submitted. In

future challenges it might be productive to specify the evaluation measures, though this does run the risk of optimizing a method with respect to a particular measure. It might be useful to further divide the datasets to obtain a more informative picture of what each method is doing, for example detecting small vs. large objects, or particular views.

It was also suggested that evaluation of discrimination between classes carried out more directly (e.g. in the forced-choice scenario), rather than in a set of binary classification tasks would be informative. Because of the use of images containing objects from multiple classes, this requires defining new evaluation measures; one possibility is to measure classification accuracy as a function of a "refusal to predict" threshold.

Increasing the Number of Classes. Future challenges might increase the number of classes beyond the four used here. This would be useful to establish how well methods scale to a large number of classes. Other work has looked at discrimination of 101 classes [17] but only in the case that each image contains a single object (using the "forced choice" scenario). New data sets must be acquired to support evaluation in the more realistic case of multiple objects in an image. A number of researchers are collecting image databases which could contribute to this.

Measuring State-of-the-Art Performance. The challenge encouraged participants to submit results based on their own (unlimited) training data, but only one such submission was received. This was disappointing because it prevented judgement of just how well these classification and detection tasks can be achieved by current methods with no constraints on training data or other resources. Future challenges should provide more motivation for participants to submit results from methods built using unlimited resources.

Acknowledgements

We are very grateful to those who provided images and their annotations; these include: Bastian Leibe & Bernt Schiele (TU-Darmstadt), 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 database collection and web pages `http://www.pascal-network.org/challenges/VOC/` were assembled by Manik Varma (University of Oxford). We are also grateful to Steve Gunn (University of Southampton) for enabling creation of the challenge web pages, 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.

References

1. S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1475–1490, 2004.
2. M. Barker and W. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.
3. K. Barnard, P. Duygulu, D. Forsyth, N. Freitas, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.
4. C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
5. E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, pages 109–124, 2002.
6. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
7. O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, Oct. 1999.
8. D. Comaniciu and P. Meer. Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*, 2:22–30, 1999.
9. D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proceedings of the 8th IEEE International Conference on Computer Vision, Vancouver, Canada*, volume 1, pages 438–445, July 2001.
10. G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV2004 Workshop on Statistical Learning in Computer Vision*, pages 59–74, 2004.
11. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 886–893, San Diego, CA, USA, June 2005.
12. T. Deselaers, D. Keysers, and H. Ney. Discriminative training for object recognition using image patches. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 157–162, San Diego, CA, USA, June 2005.
13. T. Deselaers, D. Keysers, and H. Ney. Improving a discriminative approach to object recognition using image patches. In *DAGM 2005*, volume 3663 of *LNCS*, pages 326–333, Vienna, Austria, August/September 2005.
14. G. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Proceedings of the 9th IEEE International Conference on Computer Vision, Nice, France*, pages 634–640, Oct. 2003.
15. G. Dorkó and C. Schmid. Object class recognition using discriminative local features. Technical report, INRIA, Feb. 2005.
16. J. Eichhorn and O. Chapelle. Object categorization with SVM: kernels for local features. Technical report, Max Planck Institute for Biological Cybernetics, July 2004.

17. 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 *Proceedings of the Workshop on Generative-Model Based Vision, Washington, DC, USA*, June 2004.
18. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, June 2003.
19. M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminant models for object category detection. In *Proceedings of the 10th IEEE International Conference on Computer Vision, Beijing, China*, Oct. 2005.
20. C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, Nov. 2004.
21. C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.
22. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany*, pages 137–142. Springer Verlag, Heidelberg, Germany, 1998.
23. T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. The MIT Press, Cambridge, MA, USA, 1999.
24. F. Jurie and W. Triggs. Creating efficient codebooks for visual recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vision, Beijing, China*, 2005.
25. R. Kondor and T. Jebara. A kernel between sets of vectors. In *Proceedings of the 20th International Conference on Machine Learning, Washingon, DC, USA*, 2003.
26. J. Laaksonen, M. Koskela, and E. 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.
27. D. Larlus. Creation de vocabulaires visuels efficaces pour la categorisation d'images. Master's thesis, Image Vision Robotic, INPG and UJF, june 2005.
28. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV2004 Workshop on Statistical Learning in Computer Vision*, pages 17–32, Prague, Czech Republic, May 2004.
29. B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *Proceedings of the 26th DAGM Annual Pattern Recognition Symposium*, Tuebingen, Germany, Aug. 2004.
30. B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 2005.
31. T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.
32. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
33. H. Meng, J. Shawe-Taylor, S. Szedmak, and J. R. D. Farquhar. Support vector machine to synthesise kernels. In *Proceedings of the Sheffield Machine Learning Workshop*, Sheffield, UK, 2004.
34. 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.

35. K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vision, Beijing, China*, Oct. 2005.
36. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume 2, pages 257–263, June 2003.
37. K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004.
38. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
39. A. Opelt, A. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, volume 2, pages 71–84, 2004.
40. B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, Cambridge, MA, USA, 2002.
41. E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proceedings of the 16th British Machine Vision Conference*, Oxford, UK, 2005.
42. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
43. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 18–32, 2000.
44. J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: An in-depth study. Technical report, INRIA, 2005.