The PASCAL Visual Object Classes Challenge: A Retrospective

International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

The Pascal Visual Object Classes Challenge – a Retrospective

**Mark Everingham, S. M. Ali Eslami, Luc Van Gool,
Christopher K. I. Williams, John Winn, Andrew Zisserman**

Received: date / Accepted: date

Abstract The Pascal Visual Object Classes (VOC) challenge consists of two components: (i) a publicly available *dataset* of images together with ground truth annotation and standardised evaluation software; and (ii) an annual *competition* and workshop. There are five challenges: classification, detection, segmentation, action classification, and person layout. In this paper we provide a review of the challenge from 2008–2012.

The paper is intended for two audiences: *algorithm designers*, researchers who want to see what the state of the art is, as measured by performance on the VOC datasets, along with the limitations and weak points of the current generation of algorithms; and, *challenge designers*, who want to see what we as organisers have learnt from the process and our recommendations for the organisation of future challenges.

To analyse the performance of submitted algorithms on the VOC datasets we introduce a number of novel evaluation methods: a bootstrapping method for determining whether differences in the performance of two algorithms are significant or not; a normalised average precision so that performance can be compared across classes with different proportions of positive instances; a clustering method for visualising the performance across multiple algorithms so that the hard and easy images can be identified; and the use of a joint classifier over the submitted algorithms in order to measure their complementarity and combined performance. We also analyse the community's progress through time using the methods of Hoiem et al (2012) to identify the types of occurring errors.

We conclude the paper with an appraisal of the aspects of the challenge that worked well, and those that could be improved in future challenges.

Mark Everingham, who died in 2012, was the key member of the VOC project. His contribution was crucial and substantial. For these reasons he is included as the posthumous first author of this paper. An appreciation of his life and work can be found in Zisserman et al (2012).

Mark Everingham
University of Leeds, UK

S. M. Ali Eslami (✉)
Microsoft Research, Cambridge, UK
(The majority of this work was performed whilst at the University of Edinburgh)
E-mail: alie@microsoft.com

Luc Van Gool
KU Leuven, Belgium and ETH, Switzerland

Christopher K. I. Williams
University of Edinburgh, UK

John Winn
Microsoft Research, Cambridge, UK

Andrew Zisserman
University of Oxford, UK

1 Introduction

The Pascal[1] Visual Object Classes (VOC) Challenge has been an annual event since 2006. The challenge consists of two components: (i) a publicly available *dataset* of images obtained from the Flickr web site (2013), together with ground truth annotation and standardised evaluation software; and (ii) an annual *competition* and workshop. There are three principal challenges: *classification* – "does the image contain any instances of a particular object class?" (where object classes include cars, people, dogs, etc.), *detection* – "where are the instances

[1] Pascal stands for pattern analysis, statistical modelling and computational learning. It was an EU Network of Excellence funded project under the IST Programme of the European Union.

of a particular object class in the image (if any)?", and *segmentation* – "to which class does each pixel belong?". In addition, there are two subsidiary challenges ('tasters'): *action classification* – "what action is being performed by an indicated person in this image?" (where actions include jumping, phoning, riding a bike, etc.) and *person layout* – "where are the head, hands and feet of people in this image?". The challenges were issued with deadlines each year, and a workshop held to compare and discuss that year's results and methods.

The challenges up to and including the year 2007 were described in our paper Everingham et al (2010). The purpose of this paper is not just to continue the story from 2008 until the final run of the challenge in 2012, although we will cover that to some extent. Instead we aim to inform two audiences: first, *algorithm designers*, those researchers who want to see what the state of the art is, as measured by performance on the VOC datasets, and the limitations and weak points of the current generation of algorithms; second, *challenge designers*, who want to see what we as organisers have learnt from the process and our recommendations for the organisation of future challenges.

1.1 Paper layout

This paper is organised as follows: we start with a review of the challenges in Section 2, describing in brief the competitions, datasets, annotation procedure, and evaluation criteria of the 2012 challenge, and what was changed over the 2008–2012 lifespan of the challenges. The parts on annotation procedures and changes to the challenges are intended for challenge organisers.

Section 3 provides an overview of the results for the 2012 challenge and, thereby, a snapshot of the state of the art. We then use these 2012 results for several additional and novel analyses, going further than those given at the challenge workshops and in our previous publication on the challenge (Everingham et al, 2010). At the end of Section 3 we consider the question of how the performance of algorithms can be fairly compared when all that is available is their prediction on the test set, and propose a method for doing this. This is aimed at challenge organisers.

Section 4 takes stock and tries to answer broader questions about where our field is at in terms of the classification and detection problems that can or cannot be solved. First, inspired by Hoiem et al (2012), we propose evaluation measures that normalise against the proportion of positive instances in a class (a problem when comparing average precision *across* classes). It is shown that some classes – like 'person' – still pose larger problems to modern methods than may have been believed. Second, we describe a clustering method for visualising the performance across multiple algorithms submitted during the lifespan of the challenges, so that the characteristics of hard and easy images can be identified.

Section 5 investigates the level of complementarity of the different methods. It focusses on classification, for which a 'super-method' is designed by combining the 2012 submitted methods. It turns out that quite some performance can be gained over any one existing method with such a combination, without any of those methods playing a dominant role in the super-method. Even the combination of only pairs of classifiers can bring a substantial improvement and we make suggestions for such pairs that would be especially promising. We also comment on the construction of super-methods for detection and segmentation.

In Section 6 we turn to progress through time. From the evaluation server, we have available to us the results of all algorithms for the challenges from 2009 to 2012, and we analyse these using the methods of Hoiem et al (2012) to identify the types of errors occurring across time. Although important progress has been made, it has often not been as monotonic as one might expect. This underlines the fact that novel, promising ideas may require some consolidation time and benchmark scores must not be used to discard such novelties. Also, the diversity among the scores has increased as time has progressed.

Section 7 summarises our conclusions, both about what we believe to have done well and about caveats. This section also makes suggestions that we hope will be useful for future challenge organisers.

2 Challenge Review

This section reviews the challenges, datasets, annotation and evaluation procedures over the 2009–2012 cycles of the challenge. It gives a bare bones summary of the challenges and then concentrates on changes since the 2008 release. Our companion paper (Everingham et al, 2010) describes in detail the motivation, annotations, and evaluation measures of the VOC challenges, and these details are not repeated here. Sec. 2.3 on the annotation procedure is intended principally for challenge organisers.

2.1 Challenge tasks

This section gives a short overview of the three principal challenge tasks on *classification*, *detection*, and *segmentation*, and of the two subsidiary tasks ('tasters')

Vehicles	Household	Animals	Other
Aeroplane	Bottle	Bird	Person
Bicycle	Chair	Cat	
Boat	Dining table	Cow	
Bus	Potted plant	Dog	
Car	Sofa	Horse	
Motorbike	TV/monitor	Sheep	
Train			

Table 1: The VOC classes. The classes can be considered in a notional taxonomy.

on *action classification* and *person layout*. The evaluation of each of these challenges is described in detail in Sec. 2.4.

2.1.1 Classification

For each of twenty object classes predict the presence/absence of at least one object of that class in a test image. The twenty objects classes are listed in Table 1. Participants are required to provide a real-valued confidence of the object's presence for each test image so that a precision-recall curve can be drawn. Participants may choose to tackle all, or any subset of object classes, for example 'cars only' or 'motorbikes and cars'.

Two competitions are defined according to the choice of training data: (i) taken from the VOC training/validation data provided, or (ii) from any source excluding the VOC test data. In the first competition, any annotation provided in the VOC training/validation data may be used for training, for example bounding boxes or particular views e.g. 'frontal' or 'left'. Participants are *not* permitted to perform additional manual annotation of either training or test data. In the second competition, any source of training data may be used *except* the provided test images.

2.1.2 Detection

For each of the twenty classes, predict the bounding boxes of each object of that class in a test image (if any), with associated real-valued confidence. Participants may choose to tackle all, or any subset of object classes. Two competitions are defined in a similar manner to the classification challenge.

It is clear that the additional requirement to locate the instances in an image makes detection a more demanding task than classification. Guessing the right answer is far more difficult to achieve. It is also true that detection can support more applications than mere classification, e.g. obstacle avoidance, tracking, etc. During the course of the Pascal VOC challenge it had even been suggested that only detection matters and classification is hardly relevant. However, this view is rather extreme. Even in cases where detection is the end goal, classification may be an appropriate initial step to guide resources towards images that hold good promise of containing the target class. This is similar to how an 'objectness' analysis (e.g. Alexe et al, 2010) can guide a detector's attention to specific locations within an image. Classification could also be used to put regression methods for counting into action, which have been shown to perform well without any detection (Lempitsky and Zisserman, 2010).

2.1.3 Segmentation

For each test image, predict the object class of each pixel, or give it 'background' status if the object does not belong to one of the twenty specified classes. There are no confidence values associated with this prediction. Two competitions are defined in a similar manner to the classification and detection challenges.

Segmentation clearly is more challenging than detection and its solution tends to be more time consuming. Detection can therefore be the task of choice in cases where such fine-grained image analysis is not required by the application. However, several applications do need a more detailed knowledge about object outline or shape, such as robot grasping or image retargeting. Even if segmentation is the goal, detection can provide a good initialization (e.g. Leibe et al, 2004).

2.1.4 Action classification

This taster was introduced in 2010. The motivation was that the world is dynamic and snapshots of it still convey substantial information about these dynamics. Several of the actions were chosen to involve object classes that were also part of the classification and detection challenges (like a person riding a horse, or a person riding a bike). The actions themselves were all geared towards people.

In 2010 the challenge was: for each of ten action classes predict if a specified person (indicated by a bounding box) in a test image is performing the corresponding action. The output is a real-valued confidence that the action is being performed so that a precision-recall curve can be drawn. The action classes are 'jumping', 'phoning', 'playing instrument', 'reading', 'riding bike', 'riding horse', 'running', 'taking photo', 'using computer', 'walking', and participants may choose to tackle all, or any subset of action classes, for example

'walking only' or 'walking and running'. Note, the action classes are not exclusive, for example a person can be both 'riding a bicycle' and 'phoning'. In 2011 an 'other' class was introduced (for actions different from the ten already specified). This increased the difficulty of the challenge. The output is still a real-valued confidence for each of the ten actions. As with other parts of the challenge, the training could be either based on the official PASCAL VOC training data, or on external data.

It was necessary for us to specify the person of interest in the image as there may be several people performing different actions. In 2012 the person of interest was specified by both a bounding box and a point on the torso, and a separate competition defined for each. The motivation for this additional point annotation was that the aspect ratio of the bounding box might provide some information on the action being performed, and this was almost entirely removed if only a point was provided. For example, the aspect ratio of the box could help distinguish walking and running from other action classes (this was a criticism raised during the 2011 PASCAL VOC workshop).

2.1.5 Person layout

For each person in a test image (their bounding box is provided) predict the presence or absence of parts (head, hands and feet), and the bounding boxes of those parts. The prediction of a person layout should be output with an associated real-valued confidence of the layout so that a precision-recall curve can be generated for each person. The success of the layout prediction depends both on: (i) a correct prediction of parts present/absent (e.g. are the hands visible or occluded); (ii) a correct prediction of bounding boxes for the visible parts. Two competitions are defined in a similar manner to the classification challenge.

2.2 Datasets

For the purposes of the challenge, the data is divided into two main subsets: training/validation data (`trainval`), and test data (`test`). For participants' convenience, the `trainval` data is further divided into suggested training (`train`) and validation (`val`) sets, however participants are free to use any data in the `trainval` set for training and/or validation.

There is complete annotation for the twenty classes: i.e. all images are annotated with bounding boxes for every instance of the twenty classes for the classification and detection challenges. In addition to a bounding box for each object, attributes such as: 'orientation', 'occluded', 'truncated', 'difficult'; are specified. The full list of attributes and their definitions is given in Everingham et al (2010). Fig. 1 shows samples from each of the challenges including annotations. Note, the annotations on the `test` set are not publicly released.

Statistics for the number of object instances and images in the training and validation datasets for the classification, detection, segmentation and layout challenges is given in Table 3, and for the action classification challenge in Table 4. Note, we do not release the exact numbers of object instances in the `test` set, but both the number of instances per class and number of images are approximately balanced with those in the `trainval` set.

The number of images and instances in all the tasks was increased up to 2011. From 2011 to 2012 the number of images in the classification, detection and person layout tasks was not increased, and only those for segmentation and action classification were augmented.

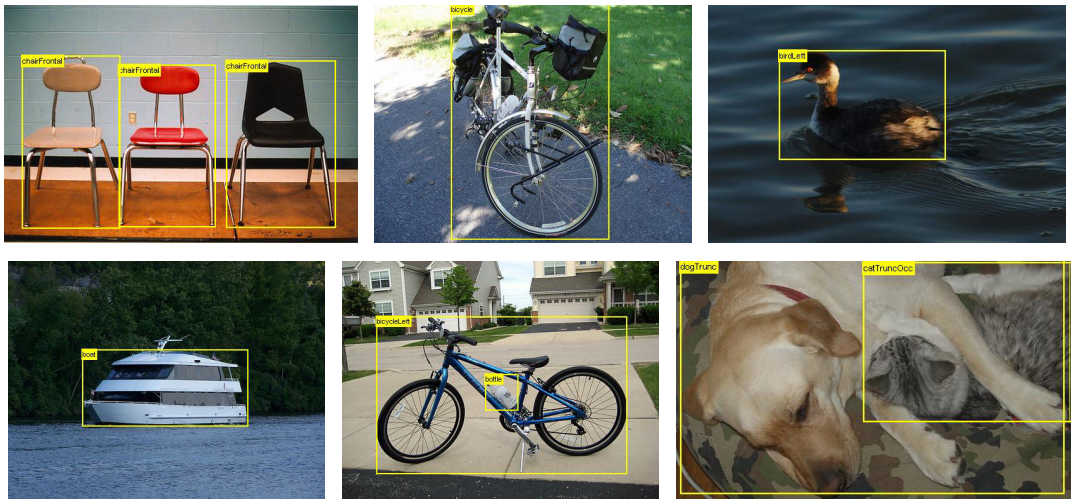
From 2009 onwards the data for all tasks consists of the previous years' images augmented with new images. Before this, in 2008 and earlier, an entirely new dataset was released each year for the classification/detection tasks. Augmenting allows the number of images to grow each year and, more importantly, means that test results can be compared with the previous years' images. Thus, for example, performance of all methods from 2009–2012, can be evaluated on the 2009 test set (although the methods may have used a different number of training images).

2.3 Annotation procedure

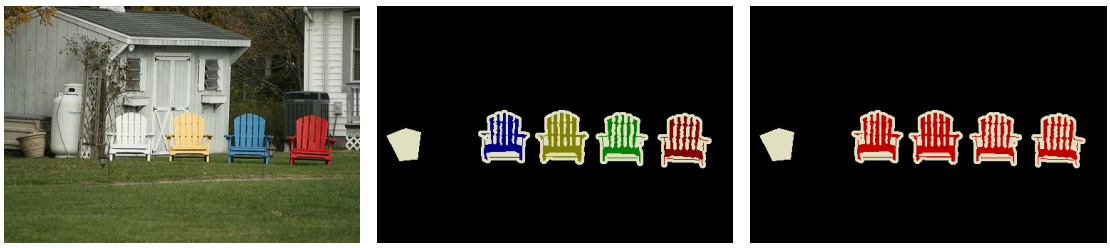
The procedure of collecting the data and annotating it with ground truth is described in our companion paper (Everingham et al, 2010). However, the annotation process has evolved since that time and we outline here the main changes in the collection and annotation procedure. Note, for challenge organisers, one essential factor in obtaining consistent annotations is to have guidelines available in advance of the annotation process. The ones used for VOC are available at the PASCAL VOC annotation guidelines web page (2012).

2.3.1 Use of Mechanical Turk for initial class labelling of the images

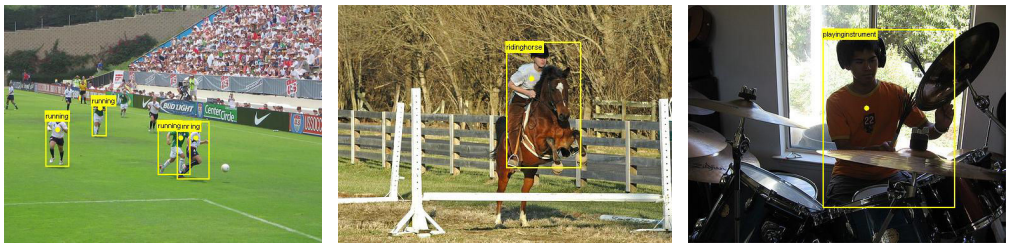
We aimed to collect a balanced set of images with a certain minimum number of instances of each class. This required finding sufficient images of the rarer classes, such as 'bus' and 'dining table'. In previous years this had been achieved by getting the annotators to focus on such classes towards the end of the annotation period,



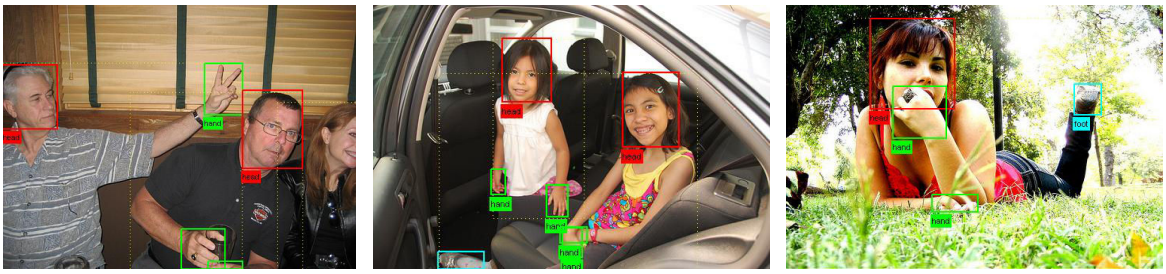
(a) Classification and detection



(b) Segmentation



(c) Action classification



(d) Person layout

Fig. 1: Sample images from the Pascal dataset. (a) Each image has an annotation file giving a bounding box and object class label for each object in one of the twenty classes present in the image. Note that multiple objects from multiple classes may be present in the same image. Annotation was performed according to a set of guidelines distributed to all annotators. (b) A subset of images are also annotated with pixel-wise segmentation of each object present, to support the segmentation competition. Segmentations are annotated at the object and object class level. (c) Images for the action classification task are disjoint from those of the classification, detection and segmentation tasks. They have been partially annotated with people, bounding boxes, reference points and their actions. Annotation was performed according to a set of guidelines distributed to all annotators. (d) Images for the person layout taster, where the test set is disjoint from the main tasks, have been additionally annotated with parts of the people (head, hands and feet).

which often meant having to skip through large numbers of images before finding examples of the desired class.

Our initial hope was to use Mechanical Turk (MT) for most or all of the annotation. We were not able to obtain MT bounding box annotations of sufficiently high quality to achieve this. However, the labels of whether a class was present or not were high enough quality to allow a balanced image set to be selected, prior to annotation by our trained annotators. This saved substantial time during the annotation period at relatively low cost.

2.3.2 Interleaved annotation and checking

Previously, significant effort had been spent in checking and correcting the annotations after the main annotation period. For segmentation annotation this was a very time consuming task. It was also common to discover patterns of errors by new annotators who had not yet become fully familiarised with the annotation guidelines.

To help new annotators to self-correct and to reduce the amount of post-hoc correction needed, the main annotation period was changed to incorporate both annotation and checking running in parallel. After each image was annotated, it was passed to a different annotator for checking. Examples of common errors were pasted onto a common noticeboard (with the annotator remaining anonymous) so that the entire group could understand and avoid such errors in future. In this way, errors were picked up and corrected earlier and post-hoc checking was substantially reduced. Over time, the checking part of the annotation effort grew to take around 50% of the annotators' time.

2.3.3 Increased time spent on segmentation

As the datasets for the classification and detection tasks became large, more emphasis was placed on increasing the size of the segmentation dataset. Segmentation requires substantially more annotation effort than detection – it can easily take ten times as long to segment an object than to draw a bounding box around it. Over time, the annotation period was adapted until around 50% of the time was spent on segmentation annotation and checking.

Once post-hoc segmentation correction was completed, each annotator was sent a report detailing the number and kind of errors that they made, so they could avoid such errors in future years.

2.3.4 Co-located annotators

In the earlier years of the challenge (up to 2008) annotations were carried out with all annotators located at the same site. This enabled efficient discussion, e.g. of challenging or unusual cases, and was very flexible in allowing changes of priorities and training. In later years the annotators could remain in their own labs and a web interface was developed for annotating using a standard client-server architecture. However, the annotation event took place simultaneously (over a period of three to five days) so that even if the annotators were not co-located they could still discuss in real-time using Skype messaging. In addition, some experienced annotators were always included at each remote site to train novices.

2.4 Submission and evaluation

2.4.1 Submission of results

The running of the challenge consisted of two phases: At the start of the challenge, participants were issued a development kit comprising training/validation images with annotation, and MATLAB[2] software to access the annotation (stored in an XML format compatible with LabelMe, Russell et al, 2008), to compute the evaluation measures, and including simple baseline implementations for each competition. In the second phase, *unannotated* test images were distributed. Participants were then required to run their methods on the test data and submit results to an evaluation server. The test data was available for approximately three months before submission of results.

2.4.2 Evaluation of results

In addition to withholding the test data annotation, it was also required that participants submit only a single entry per method. We encouraged participants to observe the best practice guidelines, given at the PASCAL VOC best practice guidelines web page (2012), that parameters should be tuned on the validation set and algorithms run only once on the test set, so that the organisers were not asked to choose the best result for them. To add to this encouragement, the evaluation server restricted the number of times a participant could submit results for earlier years (since performance could be partially gauged as earlier years' images are a subset of the current year's).

[2] MATLAB ® is a registered trademark of MathWorks, Inc.

2.4.3 Classification and detection

Both the classification and detection tasks were evaluated as a set of 20 independent two-class tasks: e.g. for classification "is there a car in the image?", and for detection "where are the cars in the image (if any)?". A separate 'score' is computed for each of the classes. For the classification task, participants submitted results in the form of a confidence level for each image and for each class, with larger values indicating greater confidence that the image contains the object of interest. For the detection task, participants submitted a bounding box for each detection, with a confidence level for each bounding box. The provision of a confidence level allows results to be ranked such that the trade-off between false positives and false negatives can be evaluated, without defining arbitrary costs on each type of classification error.

In the case of classification, the correctness of a class prediction depends only on whether an image contains an instance of that class or not. However, for detection a decision must be made on whether a prediction is correct or not. To this end, detections were assigned to ground truth objects and judged to be true or false positives by measuring bounding box overlap. To be considered a correct detection, the area of overlap a_o between the predicted bounding box B_p and ground truth bounding box B_{gt} must exceed 50% by the formula:

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

where $B_p \cap B_{gt}$ denotes the intersection of the predicted and ground truth bounding boxes and $B_p \cup B_{gt}$ their union.

Detections output by a method were assigned to ground truth object annotations satisfying the overlap criterion in order ranked by the (decreasing) confidence output. Ground truth objects with no matching detection are false negatives. Multiple detections of the same object in an image were considered false detections, e.g. 5 detections of a single object counted as 1 correct detection and 4 false detections – it was the responsibility of the participant's system to filter multiple detections from its output.

For a given task and class, the precision-recall curve is computed from a method's ranked output. Up until 2009 interpolated average precision (Salton and Mcgill, 1986) was used to evaluate both classification and detection. However, from 2010 onwards the method of computing AP changed to use all data points rather than TREC-style sampling (which only sampled the monotonically decreasing curve at a fixed set of uniformly-spaced recall values $0, 0.1, 0.2, ..., 1$). The intention in interpolating the precision-recall curve was to reduce the impact of the 'wiggles' in the precision-recall curve, caused by small variations in the ranking of examples. However, the downside of this interpolation was that the evaluation was too crude to discriminate between the methods at low AP.

2.4.4 Segmentation

The segmentation challenge was assessed per class on the intersection of the inferred segmentation and the ground truth, divided by the union (commonly referred to as the 'intersection over union' metric):

$$\text{seg. accuracy} = \frac{\text{true pos.}}{\text{true pos.} + \text{false pos.} + \text{false neg.}} \quad (2)$$

Pixels marked 'void' in the ground truth (i.e. those around the border of an object that are marked as neither an object class or background) are excluded from this measure. Note, we did not evaluate at the individual object level, even though the data had annotation that would have allowed this. Hence, the precision of the segmentation between overlapping objects of the same class was not assessed.

2.4.5 Action classification

The task is assessed in a similar manner to classification. For each action class a score for that class should be given for the person performing the action (indicated by a bounding box or a point), so that the test data can be ranked. The average precision is then computed for each class.

2.4.6 Person layout

At test time the method must output the bounding boxes of the parts (head, hands and feet) that are visible, together with a single real-valued confidence of the layout so that a precision/recall curve can be drawn.

From VOC 2010 onwards, person layout was evaluated by how well each part *individually* could be predicted: for each of the part types (head, hands and feet) a precision/recall curve was computed, using the confidence supplied with the person layout to determine the ranking. A prediction of a part was considered true or false according to the overlap test, as used in the detection challenge, i.e. for a true prediction the bounding box of the part overlaps the ground truth by at least 50%. For each part type, the average precision was used as the quantitative measure.

This method of evaluation was introduced following criticism of an earlier evaluation used in 2008, that was

	jumping	phoning	playing	reading	ridingbike	ridinghorse	running	takingphoto	using	walking	other	total
Img	405	444	459	463	400	411	310	414	395	386	799	4588
Obj	495	457	619	530	578	534	561	456	476	597	1043	6278

Table 4: Statistics of the action classification VOC2012 dataset. For the `trainval` dataset, the number of images containing at least one person performing a given action, and the corresponding number of objects are shown.

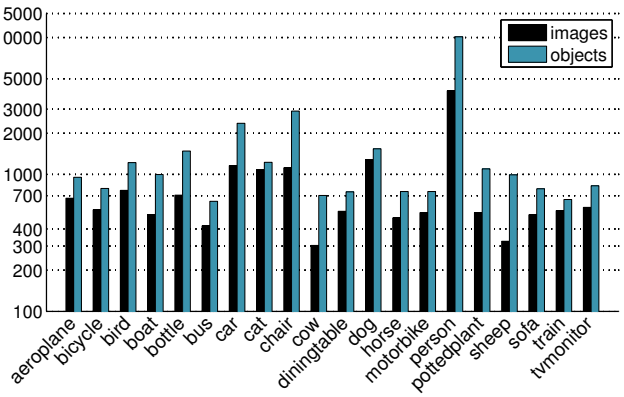


Fig. 2: Summary of the main VOC2012 dataset. Training and validation images only. Histogram by class of the number of objects and images containing at least one object of the corresponding class. Note the log scale on the vertical axis. Best viewed in colour.

considered too strict and demanding (given the state of the art in layout detection algorithms at that time). In VOC 2008, the layout was still assessed by computing a precision-recall curve, but rather than assessing parts individually the *entire* layout was assessed. To be considered a true positive, each layout estimate had to satisfy two criteria: (i) the set and number of predicted parts matches ground truth exactly e.g. {head, hand, hand} or {head, hand, foot}; and (ii) the predicted bounding box of each part overlaps ground truth by at least 50%. These criteria were relaxed from VOC 2010 on, though this task never became as popular as the others.

3 VOC 2012 Results and Rankings

In this section we review the results of the VOC 2012 challenge to give a snapshot of the state-of-the-art in the final year of the challenge. Secs. 3.1, 3.2, 3.3 and 3.4 describe the top performing methods for the classification, detection, segmentation and action classification challenges in 2012 respectively (there were no entries for complete person layout so we do not include that here). Having done that, in Sec. 3.5, we then propose a method to assess whether differences in AP between the methods are significant or not based on bootstrap sampling – this is important as it enables one to tell if the method proposed by the 'runner up' should actually be considered as equivalent to that of the 'winner'.

The VOC 2012 participants (and our codenames for them) are listed in Table 2. Where possible we have identified publications describing these methods in the right hand column of the table; in addition short descriptions were provided by the participants and are available at the Pascal VOC 2012 challenge results webpage (2012).

The number of images and objects in the VOC 2012 training and validation sets are shown as a histogram for the classification and detection challenges in Figure 2. The numbers are tabulated in Table 3 for classification, detection and segmentation, and in Table 4 for the action classification challenge. There were 850 annotated objects instances in 609 images for the person layout challenge.

3.1 Classification

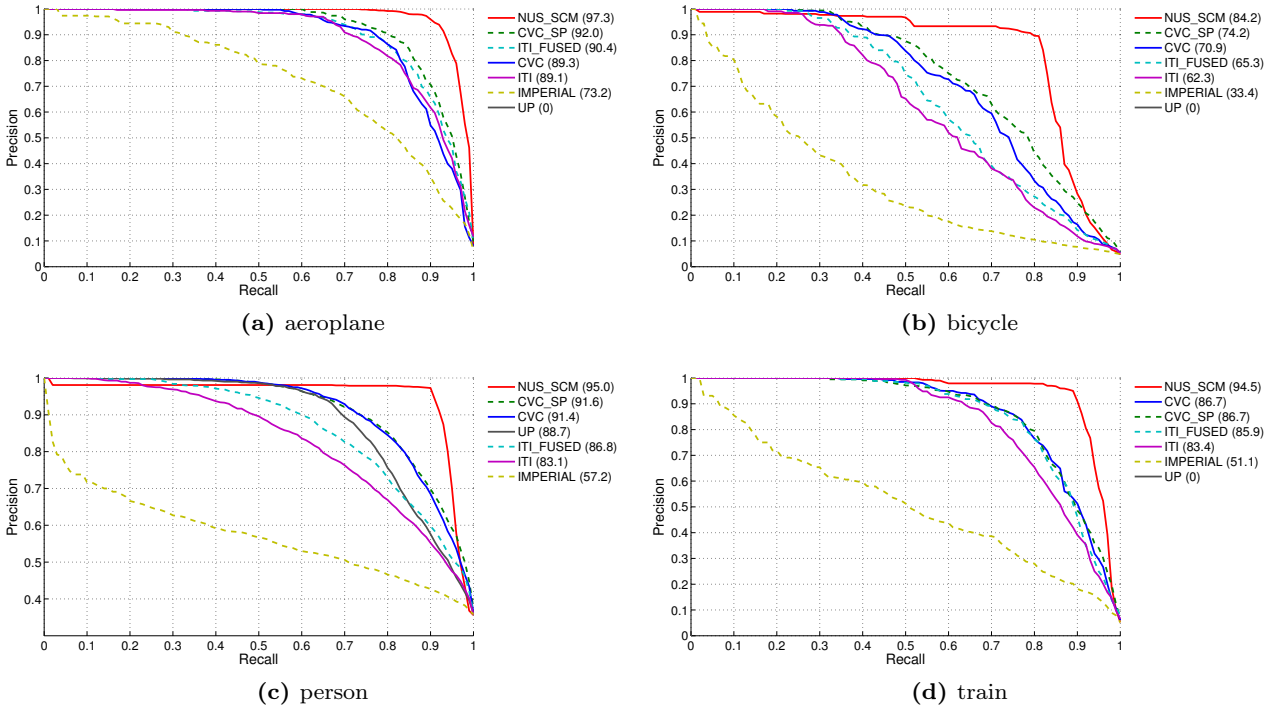
Figure 5 and Table 5 give summaries of the results of the classification challenge for both competition 1 (using supplied data only) and competition 2 (which also allowed external data to be used). Figure 3 shows precision-recall curves for a sample of the classes. The winning method for competition 1 is NUS_SCM. Its performance exceeded all other methods (including those in competition 2) for all classes in 2012, and also improved on the 2011 winning entries in all but one class ('pottedplant'). The NUS_SCM method started from a fairly standard pipeline of a bag-of-visual-words (BOW) representation and spatial pyramid matching (SPM), followed by a support vector machine (SVM) classifier (see Sec. 6.1 for more details). To this they added the identification and use of sub-categories (e.g. identifying different types of chair), and a refinement of SPM based on the output of sliding window detection confidence maps.

3.2 Detection

Figure 6 and Table 6 give the results of the detection challenge for competition 3 (using supplied data only); there were no entries for competition 4. Figure 4 shows precision-recall curves for a sample of the classes. The winning method was UVA_HYBRID (see Van de

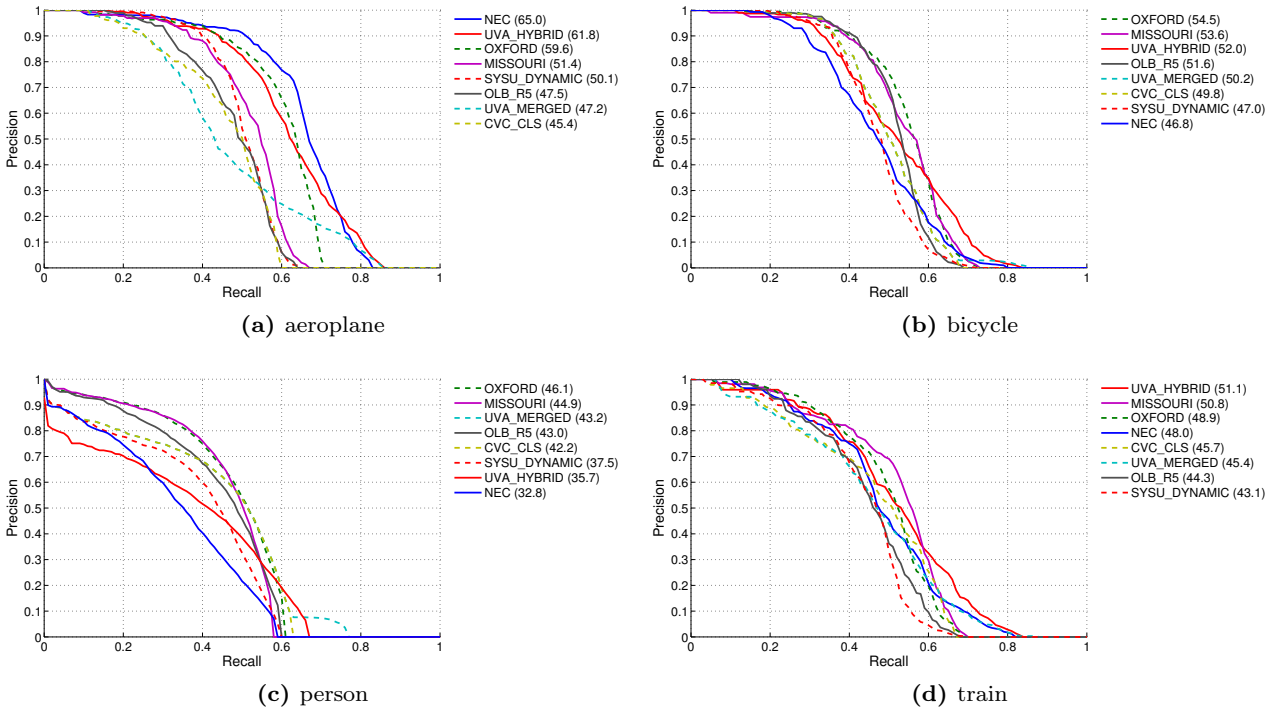
Codename	Cls	Det	Seg	Act	Institutions	Contributors	References
BONN_CSI	·	·	●	·	University of Bonn, Georgia Institute of Technology, University of Coimbra	Joao Carreira, Fuxin Li, Guy Lebanon, Cristian Sminchisescu	Li et al (2013)
BONN_JOINT	·	·	●	·	University of Bonn, Georgia Institute of Technology, University of Coimbra, Vienna University of Technology	Joao Carreira, Adrian Ion, Fuxin Li, Cristian Sminchisescu	Ion et al (2011a,b)
BONN_LINEAR	·	·	●	·	University of Bonn, University of Coimbra	Joao Carreira, Rui Caseiro, Jorge Batista, Cristian Sminchisescu	Carreira et al (2012)
CVC	●	·	·	·	Computer Vision Barcelona	Fahad Khan, Camp Davesa, Joost van de Weijer, Rao Muhammad Anwer, Albert Gordo, Pep Gonfaus, Ramon Baldrich, Antonio Lopez	Khan et al (2012a)
CVC_CLS	·	●	·	·	Computer Vision Barcelona	Albert Gordo, Camp Davesa, Fahad Khan, Pep Gonfaus, Joost van de Weijer, Rao Muhammad Anwer, Ramon Baldrich, Jordi Gonzalez, Ernest Valveny	Khan et al (2012a,b)
CVC_SP	●	·	·	·	Computer Vision Barcelona, University of Amsterdam, University of Trento	Fahad Khan, Jan van Gemert, Camp Davesa, Jasper Uijlings , Albert Gordo, Sezer Karaoglu, Koen van de Sande, Pep Gonfaus, Rao Muhammad Anwer, Joost van de Weijer, Cees Snoek, Ramon Baldrich, Nicu Sebe, Theo Gevers	Khan et al (2012a,b); Karaoglu et al (2012); Van Gemert (2011)
HU	·	·	·	●	Hacettepe University, Bilkent University	Cagdas Bas, Fadime Sener, Nazli Ikizler-Cinbis	Sener et al (2012)
IMPERIAL	●	·	·	·	Imperial College London	Ioannis Alexiou, Anil A. Bharath	Alexiou and Bharath (2012)
ITI, ITI_ENTROPY, ITI_FUSED	●	·	·	·	ITI-CERTH, University of Surrey, Queen Mary University of London	Elisavet Chatzilari, Spiros Nikolopoulos, Yiannis Kompatsiaris, Joseph Kittler	-
MISSOURI	·	●	·	·	University of Missouri Columbia	Guang Chen, Miao Sun, Xutao Lv, Yan Li, Tony Han	-
NEC	·	●	·	·	NEC Laboratories America, Stanford University	Olga Russakovsky, Xiaoyu Wang, Shenghuo Zhu, Li Fei-Fei, Yuanqing Lin	Russakovsky et al (2012)
NUS_SCM	●	·	·	·	National University of Singapore, Panasonic Singapore Laboratories, Sun Yat-sen University	Dong Jian, Chen Qiang, Song Zheng, Pan Yan, Xia Wei, Yan Shuicheng, Hua Yang, Huang Zhongyang, Shen Shengmei	Song et al (2011); Chen et al (2012)
NUS_SP	·	·	●	·	National University of Singapore, Panasonic Singapore Laboratories	Wei Xia, Csaba Domokos, Jian Dong, Shuicheng Yan, Loong Fah Cheong, Zhongyang Huang, Shengmei Shen	Xia et al (2012)
OLB_R5	·	●	·	·	Orange Labs Beijing, France Telecom	Zhao Feng	-
OXFORD	·	●	·	·	University of Oxford	Ross Girshick, Andrea Vedaldi, Karen Simonyan	-
OXFORD_ACT	·	·	·	●	University of Oxford	Minh Hoai, Lubor Ladicky, Andrew Zisserman	Hoai et al (2012)
STANFORD	·	·	·	●	Stanford University, MIT	Aditya Khosla, Rui Zhang, Bangpeng Yao, and Li Fei-Fei	Khosla et al (2011)
SYSU_DYNAMIC	·	●	●	·	Sun Yat-Sen University	Xiaolong Wang, Liang Lin, Lichao Huang, Xinhui Zhang, Zechao Yang	Wang et al (2013)
SZU	·	·	·	●	Shenzhen University	Shiqi Yu, Shengyin Wu, Wensheng Chen	-
UP	●	·	·	·	University of Padova	Loris Nanni	Nanni and Lumini (2013)
UVA_HYBRID	·	●	·	·	University of Amsterdam	Koen van de Sande, Jasper Uijlings, Cees Snoek, Arnold Smeulders	Van de Sande et al (2011); Uijlings et al (2013)
UVA_MERGED	·	●	·	·	University of Amsterdam	Sezer Karaoglu, Fahad Khan, Koen van de Sande, Jan van Gemert, Rao Muhammad Anwer, Jasper Uijlings, Camp Davesa, Joost van de Weijer, Theo Gevers, Cees Snoek	Khan et al (2012a); Uijlings et al (2013)
UVA_NBNN	·	·	●	·	University of Amsterdam	Carsten van Weelden, Maarten van der Velden, Jan van Gemert	-

Table 2: Participation in the 2012 challenge. Each method is assigned an abbreviation used in the text, and identified as a classification (Cls), detection (Det), segmentation (Seg), or action classification (Act) method. The contributors to each method are listed with references to publications describing the method, where available.



(a) aeroplane **(b)** bicycle **(c)** person **(d)** train

Fig. 3: Classification results. Precision-recall curves are shown for a representative sample of classes. The legend indicates the AP score (%) obtained by the corresponding method.



(a) aeroplane **(b)** bicycle **(c)** person **(d)** train

Fig. 4: Detection results. Methods trained on VOC2012 data. Precision-recall curves are shown for a representative sample of classes. The legend indicates the AP score (%) obtained by the corresponding method.

		aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	total
Main	**Img**	670	552	765	508	706	421	1161	1080	1119	303	538	1286	482	526	4087	527	325	507	544	575	11540
	Obj	954	790	1221	999	1482	637	2364	1227	2906	702	747	1541	750	751	10129	1099	994	786	656	826	31561
Seg	**Img**	178	144	208	150	183	152	255	250	271	135	157	249	147	157	888	167	120	183	167	157	2913
	Obj	218	197	277	232	357	237	457	286	550	284	168	299	204	204	1738	322	308	209	189	198	6934

Table 3: Statistics of the main and segmentation VOC2012 datasets. Showing the number of images in the `trainval` dataset containing at least one object of a given class, and the corresponding number of object instances. Note that because images may contain objects of several classes, the totals shown in the Img rows are not simply the sum of that row.

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
CVC	89.3	70.9	69.8	73.9	51.3	84.8	79.6	72.9	63.8	59.4	64.1	64.7	75.5	79.1	91.4	42.7	63.2	61.9	86.7	73.8
CVC_SP	92.0	74.2	73.0	77.5	54.3	85.2	81.9	76.4	65.2	63.2	68.5	68.9	78.2	81.0	91.6	55.9	69.4	65.4	86.7	77.4
IMPERIAL	73.2	33.4	31.0	44.7	17.0	57.7	34.4	45.9	41.2	18.1	30.2	34.3	23.1	39.3	57.2	11.9	23.1	25.3	51.1	36.2
ITI	89.1	62.3	60.0	68.1	33.4	79.8	66.9	70.3	57.4	51.0	55.0	59.3	68.6	74.5	83.1	25.6	57.2	53.8	83.4	64.9
ITI_FUSED	90.4	65.3	65.8	72.3	37.6	80.6	70.5	72.4	60.3	55.1	61.4	63.6	72.4	77.4	86.8	37.7	61.1	57.2	85.9	68.7
NUS_SCM	**97.3**	**84.2**	**80.8**	**85.3**	**60.8**	**89.9**	**86.8**	**89.3**	**75.4**	**77.8**	**75.1**	**83.0**	**87.5**	**90.1**	**95.0**	**57.8**	**79.2**	**73.4**	**94.5**	**80.7**
UP	–	–	–	–	–	–	–	–	–	–	–	–	–	–	88.7	–	–	–	–	–
METHODS BELOW ALSO TRAINED ON EXTERNAL DATA																				
ITI_ENTROPY	**88.1**	**63.0**	**61.9**	**68.6**	**34.9**	**79.6**	**67.4**	**70.5**	**57.5**	**52.0**	**55.3**	**60.1**	**68.7**	**74.3**	**83.2**	**26.4**	**57.6**	**53.4**	**83.0**	**64.0**

Table 5: Classification results. For each object class and submission, the AP score (%) is shown. Gold entries in each column denote the maximum AP for the corresponding class, and silver entries denote the results ranked in second place. Competition 1 results are in the top part of the table, and competition 2 in the lower part.

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
CVC_CLS	45.4	49.8	15.7	16.0	26.3	54.6	44.8	35.1	16.8	31.3	23.6	26.0	45.6	49.6	42.2	14.5	30.5	28.5	45.7	40.0
MISSOURI	51.4	53.6	18.3	15.6	31.6	56.5	47.1	38.6	**19.5**	31.9	22.1	25.0	50.3	51.9	44.9	11.9	37.7	30.6	50.8	39.2
NEC	**65.0**	46.8	**25.0**	24.6	16.0	51.0	44.9	51.5	13.0	26.6	31.0	40.2	39.7	51.5	32.8	12.6	35.7	33.5	48.0	44.8
OLB_R5	47.5	51.6	14.2	12.6	27.3	51.8	44.2	25.3	17.8	30.2	18.1	16.9	46.9	50.9	43.0	09.5	31.2	23.6	44.3	22.1
SYSU_DYNAMIC	50.1	47.0	07.9	03.8	24.8	47.2	42.8	31.1	17.5	24.2	10.0	21.3	43.5	46.4	37.5	07.9	26.4	21.5	43.1	36.7
OXFORD	59.6	**54.5**	21.9	21.6	**32.1**	52.5	**49.3**	40.8	19.1	**35.1**	28.9	37.2	**50.9**	49.9	**46.1**	15.6	39.3	35.6	48.9	42.8
UVA_HYBRID	61.8	52.0	24.6	**24.8**	20.2	**57.1**	44.5	**53.6**	17.4	33.0	**38.2**	**42.8**	48.8	**59.4**	35.7	**22.8**	**40.3**	**39.5**	**51.1**	**49.5**
UVA_MERGED	47.2	50.2	18.3	21.4	25.2	53.3	46.3	46.3	17.5	27.8	30.3	35.0	41.6	52.1	43.2	18.0	35.2	31.1	45.4	44.4

Table 6: Detection results. Methods trained on VOC2012 data. For each object class and submission, the AP score (%) is shown. Gold entries in each column denote the maximum AP for the corresponding class, and silver entries denote the results ranked in second place.

Sande et al, 2011) which used multiple segmentations to hypothesise bounding boxes bottom up, thus avoiding an expensive sliding window search (with potentially more false positives). These candidate bounding boxes were then classified using a BOW feature representation, SPM, and a SVM using the histogram intersection kernel.

The method from Oxford won on six classes. This used a local implementation of the deformable parts model (DPM; Felzenszwalb et al, 2010) sliding window detector to propose candidate regions. The top 100 candidates were then re-scored using a homogeneous kernel map (χ^2) SVM combining the DPM's scores, two descriptors computed on the regions, together with two context models (the context scoring of Felzenszwalb et al, and context from image classification scores).

	mean	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
BONN_CSI	45.4	85.0	59.3	27.9	43.9	39.8	41.4	52.2	**61.5**	**56.4**	13.6	44.5	26.1	42.8	51.7	**57.9**	51.3	29.8	45.7	28.8	49.9	43.3
BONN_JOINT	47.0	**85.1**	**65.4**	29.3	**51.3**	33.4	44.2	59.8	60.3	52.5	13.6	**53.6**	**32.6**	40.3	**57.6**	57.3	49.0	33.5	53.5	29.2	47.6	37.6
BONN_LINEAR	44.8	83.9	60.0	27.3	46.4	40.0	41.7	57.6	59.0	50.4	10.0	41.6	22.3	43.0	51.7	56.8	50.1	33.7	43.7	29.5	47.5	44.7
NUS_SP	**47.3**	82.8	52.9	**31.0**	39.8	**44.5**	**58.9**	**60.8**	52.5	49.0	**22.6**	38.1	27.5	**47.4**	52.4	46.8	**51.9**	**35.7**	**55.2**	**40.8**	**54.2**	**47.8**
UVA_NBNN	11.3	63.2	10.5	02.3	03.0	03.0	01.0	30.2	14.9	15.0	00.2	06.1	02.3	05.1	12.1	15.3	23.4	00.5	08.9	03.5	10.7	05.3
METHODS BELOW ALSO TRAINED ON EXTERNAL DATA																						
BONN_CSI	46.8	85.0	63.6	26.8	45.6	**41.7**	47.1	54.3	58.6	55.1	**14.5**	49.0	30.9	**46.1**	52.6	**58.2**	**53.4**	32.0	44.5	**34.6**	45.3	43.1
BONN_JOINT	**47.5**	**85.2**	63.4	**27.3**	**56.1**	37.7	**47.2**	57.9	**59.3**	55.0	11.5	**50.8**	30.5	45.0	**58.4**	57.4	48.6	**34.6**	**53.3**	32.4	47.6	39.2
BONN_LINEAR	46.7	84.7	**63.9**	23.8	44.6	40.3	45.5	**59.6**	58.7	**57.1**	11.7	45.9	**34.9**	43.0	54.9	58.0	51.5	34.6	44.1	29.9	**50.5**	**44.5**

Table 7: Segmentation results. For each object class and submission, the AP score (%) is shown. Gold entries in each column denote the maximum AP for the corresponding class, and silver entries denote the results ranked in second place. Competition 5 results are in the top part of the table, and competition 6 in the lower part.

3.3 Segmentation

Table 7 gives the results of the segmentation challenge for competition 5 (using supplied data only) and competition 6 (which also allowed external data to be used). The winning method for competition 5 was NUS_SP which used an object detector to identify object bounding boxes and then determined a segmentation for each bounding box using a superpixel-based MRF (see Xia et al, 2012). This method achieved a mean AP 4% higher than the winner of the previous year, which suggests that segmentation methods continue to improve, although some of the increase may be due to the additional training data available in 2012.

The second placed method in competition 5 and the winning entry in competition 6 was BONN_JOINT which created multiple segmentations of each image and then sampled from a distribution over tilings constructed from these segments (see Ion et al, 2011a). Parameter learning was achieved using the method of Ion et al (2011b). The additional training data used in competition 6 was a set of ground truth annotations provided by the Berkeley vision group. This data proved to be valuable in that it increased the mean AP of this method by about 0.5%.

3.4 Action classification

Table 8 gives the results of the action classification challenge. This consisted of competitions 9 (using only supplied data) and competition 10 (which also allowed external data to be used).

The winning method STANFORD for competition 9 is mostly described in the paper by Khosla et al (2011), and that of OXFORD_ACT for competition 10 in Hoai et al (2012). What these high-scoring approaches seem

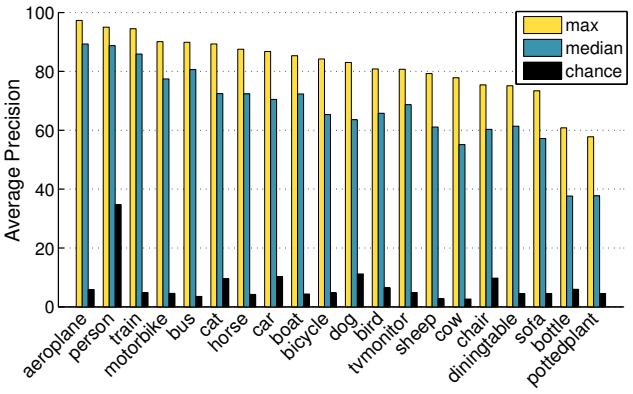


Fig. 5: Summary of the 2012 classification results by class (competition 1). For each class three values are shown: the maximum AP obtained by any method (max), the median AP over all methods (median) and the AP obtained by a random ranking of the images (chance).

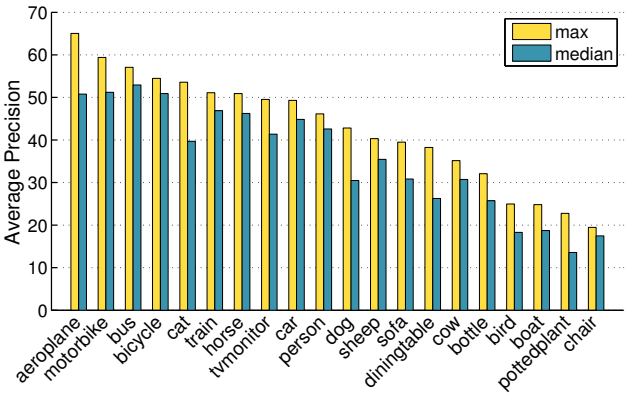


Fig. 6: Summary of the 2012 detection results by class. For each class two values are shown: the maximum AP obtained by any method (max) and the median AP over all methods (median).

	jumping	phoning	playinginstrument	reading	ridingbike	ridinghorse	running	takingphoto	usingcomputer	walking
STANFORD	**75.7**	44.8	**66.6**	**44.4**	**93.2**	**94.2**	87.6	**38.4**	**70.6**	**75.6**
SZU	73.8	**45.0**	62.8	41.4	93.0	93.4	**87.8**	35.0	64.7	73.5
METHODS BELOW ALSO TRAINED ON EXTERNAL DATA										
HU	59.4	39.6	56.5	34.4	75.6	80.2	74.3	27.6	55.2	56.6
OXF_ACT	**77.0**	**50.4**	**65.3**	**39.5**	**94.1**	**95.9**	**87.7**	**42.7**	**68.6**	**74.5**

Table 8: Action classification results. For each object class and submission, the AP score (%) is shown. Gold entries in each column denote the maximum AP for the corresponding class, and silver entries denote the results ranked in second place. Competition 9 results are in the top half of the table, and competition 10 in the lower half.

to have in common is a top-down analysis. STANFORD first focuses on larger, more globally enclosing boxes, and then homes in on smaller ones that capture details relevant to the action. OXFORD_ACT followed an approach that was quite different in the specifics, but also started with global regions, then homed in on telling details like the hands. A difference was that for the OXFORD_ACT method the regions were pre-selected by the designers, while the STANFORD method itself determines which portions of the given bounding box are relevant at the different stages of processing. The methods also shared their use of a wide variety of features. The OXFORD_ACT method was in competition 10 as it used additional data to train detectors for the upper body and musical instruments.

3.5 Bootstrapping AP and rank

In the challenge different methods will produce different scores on each class and competition. For classification, detection, and action classification this will be the AP score, while for segmentation it is the segmentation accuracy (see Eq. 2). This is a single number that summarises a method's performance on a whole dataset — how should we assess if differences between methods are significant? A simple approach to this question is via the bootstrap (see e.g. Wasserman, 2004, Ch. 8), where the data points are sampled *with replacement* from the original n test points to produce bootstrap replicates. We first came across this idea in the blog comment by O'Connor (2010), although bootstrapping of ROC curves has been discussed by many authors, e.g. Hall et al (2004); Bertail et al (2009).

We can use the bootstrap in a number of different ways: to simply judge the variability for a given method, to compare the relative strength of two methods, or to look at rank ranges in order to get an overall sense of all methods in a competition.

For a single method we can obtain a bootstrap estimate of the confidence interval for a method's score by running a large number of bootstrap replicates, sorting the resulting scores, and then returning the $\alpha/2$ and $1-\alpha/2$ quantiles, where for example $\alpha = 0.05$ would yield a 95% confidence interval. (This is the percentile interval method described in Wasserman, 2004, Sec. 8.3.)

To compare two methods A and B, we first compute the difference in score for each method on each bootstrap sample. We then use the percentile bootstrap to estimate a confidence interval, with a null hypothesis that A is equivalent to B (at the $1-\alpha$ level). This is rejected if zero is not contained in the confidence interval, leading to the conclusion that method A is statistically significantly better than method B, or vice versa, depending on the result. This procedure is more informative than the unpaired confidence intervals in determining whether two methods are significantly different; for example a variation in the hardness of the bootstrap replicates may give rise to overlapping score intervals, even if method A always beats method B.

Thirdly, in the challenge we can also determine the rank of each method on each bootstrap replicate, and thus a confidence interval for the rank of a method (using $\alpha/2$ and $1-\alpha/2$ quantiles as above). This can provide a useful summary of the relative strength of the methods without the need for pairwise comparisons. Note that rank ranges depend on all entrants in a competition, while the individual confidence interval is a property of a single method.

These bootstrap ideas are illustrated in detail for four classes in Tables 9 and 10, for the classification and detection competitions respectively. Summary results for all classes highlighting methods that are not significantly different from the leading one are shown in Table 11 (for classification) and Table 12 (for detection).

4 What We Can and Cannot Do Today

In this section we examine the results of the VOC classification and detection competitions in more detail to answer the following questions:

– which classes are current methods doing better or worse on?

	aeroplane					bottle					person					potted plant				
	AP range			Rank	RR	AP range			Rank	RR	AP range			Rank	RR	AP range			Rank	RR
	0.025	0.5	0.975			0.025	0.5	0.975			0.025	0.5	0.975			0.025	0.5	0.975		
CVC	87.2	89.4	91.1	4	4-5	47.4	51.4	55.0	3	3	90.7	91.4	92.0	3	2-3	38.3	42.9	47.2	3	3
CVC_SP	90.2	92.1	93.5	2	2	50.6	54.4	58.0	2	2	91.0	91.6	92.2	2	2-3	**51.6**	**56.1**	**60.3**	**2**	**1-2**
IMPERIAL	70.0	73.3	76.4	6	6	15.2	17.3	19.7	6	6	55.6	57.3	59.1	7	7	10.3	12.1	14.2	6	6
ITI	86.8	89.1	90.8	5	4-5	30.2	33.6	37.2	5	5	82.1	83.2	84.2	6	6	22.3	26.0	29.8	5	5
ITI_FUSED	88.3	90.5	92.0	3	3	34.2	37.9	41.7	4	4	85.9	86.8	87.6	5	5	33.4	37.8	42.3	4	4
NUS_SCM	**96.4**	**97.3**	**98.1**	**1**	**1**	**57.3**	**61.1**	**64.9**	**1**	**1**	**94.4**	**95.1**	**95.6**	**1**	**1**	**53.1**	**58.0**	**62.6**	**1**	**1-2**
UP	–	–	–	7	7	–	–	–	7	7	88.0	88.7	89.5	4	4	–	–	–	7	7

Table 9: Bootstrapped classification results on 4 classes. Here $\alpha = 0.05$, RR denotes the rank range, and the leading methods that are not statistically significantly different from each other are highlighted in gold.

	bicycle					bus					horse					potted plant				
	AP range			Rank	RR	AP range			Rank	RR	AP range			Rank	RR	AP range			Rank	RR
	0.025	0.5	0.975			0.025	0.5	0.975			0.025	0.5	0.975			0.025	0.5	0.975		
CVC_CLS	47.0	49.8	52.5	6	6	51.4	54.6	57.9	3	3	42.6	45.7	48.7	5	4-6	12.9	14.7	16.4	4	3-5
MISSOURI	**50.9**	**53.7**	**56.6**	**2**	**1-3**	**53.4**	**56.5**	**60.2**	**2**	**1-2**	**47.0**	**50.4**	**53.5**	**2**	**1-3**	10.1	12.0	13.8	6	5-6
NEC	43.9	46.8	49.9	8	7-8	47.8	50.9	54.5	7	4-7	36.6	39.8	43.2	8	7-8	10.9	12.8	14.8	5	4-6
OLB_R5	48.9	51.7	54.5	4	3-5	48.6	51.8	54.9	6	4-7	43.7	47.0	50.4	4	3-5	07.7	09.6	11.3	7	7
SYSU_DYNAMIC	44.0	47.0	49.9	7	7-8	43.9	47.2	50.6	8	8	40.1	43.7	47.6	6	5-7	06.5	07.9	09.5	8	8
OXFORD	**51.7**	**54.5**	**57.3**	**1**	**1-2**	49.5	52.6	55.8	5	4-7	**47.7**	**51.0**	**54.3**	**1**	**1-3**	13.5	15.7	17.7	3	3-4
UVA_HYBRID	49.1	52.0	54.7	3	2-5	**54.0**	**57.1**	**60.2**	**1**	**1-2**	**45.7**	**49.0**	**52.0**	**3**	**1-4**	**20.1**	**22.8**	**25.6**	**1**	**1**
UVA_MERGED	47.4	50.2	52.9	5	3-5	50.2	53.4	56.7	4	4-7	38.5	41.7	44.9	7	6-8	15.8	18.2	20.5	2	2

Table 10: Bootstrapped detection results on 4 classes. Here $\alpha = 0.05$, RR denotes the rank range, and the leading methods that are not statistically significantly different from each other are highlighted in gold.

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
CVC	89.4	70.8	69.8	73.9	51.4	84.9	79.7	72.9	63.9	59.6	64.3	64.8	75.8	79.1	91.4	42.9	63.5	62.1	86.7	73.8
CVC_SP	92.1	74.2	73.1	77.5	54.4	85.2	81.9	76.4	65.3	63.6	68.7	69.0	78.3	80.9	91.6	**56.1**	69.6	65.5	86.7	77.4
IMPERIAL	73.3	33.6	31.1	45.0	17.3	57.8	34.7	46.0	41.3	18.7	30.7	34.6	23.3	39.5	57.3	12.1	23.7	25.6	51.4	36.5
ITI	89.1	62.4	60.1	68.2	33.6	79.8	67.0	70.3	57.5	51.3	55.3	59.4	68.7	74.5	83.2	26.0	57.4	54.1	83.4	64.9
ITI_FUSED	90.5	65.4	65.9	72.3	37.9	80.7	70.6	72.5	60.4	55.4	61.7	63.6	72.5	77.4	86.8	37.8	61.2	57.3	85.8	68.8
NUS_SCM	**97.3**	**84.3**	**80.9**	**85.4**	**61.1**	**90.0**	**86.9**	**89.4**	**75.5**	**78.2**	**75.4**	**83.2**	**87.6**	**90.2**	**95.1**	**58.0**	**79.6**	**73.8**	**94.5**	**80.9**
UP	–	–	–	–	–	–	–	–	–	–	–	–	–	–	88.7	–	–	–	–	–

Table 11: Bootstrapped classification results on all classes. The leading methods that are not statistically significantly different from each other are highlighted in gold.

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
CVC_CLS	45.4	49.8	15.7	16.1	26.3	54.6	44.8	35.1	16.8	31.4	23.6	26.0	45.7	49.7	42.3	14.7	30.5	28.5	45.8	39.9
MISSOURI	51.4	**53.7**	18.3	15.6	**31.7**	**56.5**	47.1	38.7	**19.5**	32.0	22.1	25.1	**50.4**	51.9	44.9	12.0	37.8	30.8	**50.9**	39.4
NEC	**65.0**	46.8	**25.1**	**24.7**	16.1	50.9	44.9	51.6	13.0	26.7	31.0	40.2	39.8	51.6	32.8	12.8	35.8	33.7	48.0	44.7
OLB_R5	47.5	51.7	14.2	12.6	27.4	51.8	44.2	25.5	17.8	30.3	18.2	17.0	47.0	50.9	43.0	09.6	31.3	23.7	44.3	22.1
SYSU_DYNAMIC	50.0	47.0	07.9	03.8	24.9	47.2	42.7	31.3	17.5	24.4	10.1	21.4	43.7	46.4	37.5	07.9	26.4	21.6	43.2	36.5
OXFORD	59.5	**54.5**	21.9	21.7	**32.1**	52.6	**49.3**	40.8	**19.1**	**35.3**	28.9	37.2	**51.0**	49.9	**46.1**	15.7	**39.4**	35.7	**49.0**	42.8
UVA_HYBRID	61.6	52.0	**24.6**	**24.9**	20.2	**57.1**	44.5	**53.7**	17.4	**33.1**	**38.1**	**42.9**	**49.0**	**59.5**	35.8	**22.8**	**40.3**	**39.7**	**51.1**	**49.4**
UVA_MERGED	47.2	50.2	18.4	21.5	25.2	53.4	46.3	46.3	17.5	27.9	30.1	35.1	41.7	52.1	43.2	18.2	35.1	31.2	45.5	44.3

Table 12: Bootstrapped detection results on all classes. The leading methods that are not statistically significantly different from each other are highlighted in gold.

	mean	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor
BONN_CSI	**45.3**	**84.9**	59.5	**27.9**	44.1	**39.7**	41.6	**52.5**	**61.6**	**56.2**	13.4	44.4	**25.9**	**42.7**	**51.6**	**58.2**	**51.4**	**29.4**	**45.7**	28.7	**49.8**	**43.6**
BONN_JOINT	**46.9**	**85.1**	**65.9**	**29.3**	**51.7**	**33.3**	43.8	**60.1**	**60.5**	**52.2**	13.5	**54.0**	**32.6**	**40.3**	**57.7**	**57.0**	**49.0**	**33.1**	**53.6**	29.1	**47.3**	**37.6**
BONN_LINEAR	**44.8**	83.9	**60.3**	**27.3**	**46.5**	**39.9**	42.0	**57.5**	**59.2**	**50.2**	09.9	41.5	**21.7**	**42.9**	**51.8**	**57.1**	**50.1**	**33.4**	**44.0**	29.1	**47.8**	**44.8**
NUS_SP	**47.2**	82.8	**52.9**	**31.0**	40.1	**44.4**	**58.6**	**61.0**	52.4	**49.0**	**22.6**	37.9	**27.2**	**47.4**	**52.6**	**47.1**	**51.9**	**35.3**	**54.9**	**40.7**	**54.1**	**47.7**
UVA_NBNN	11.2	63.2	10.4	02.3	02.9	02.9	00.9	30.2	14.7	14.9	00.2	06.0	02.2	05.0	12.2	15.2	23.4	00.5	08.8	03.4	10.7	05.2
METHODS BELOW ALSO TRAINED ON EXTERNAL DATA																						
BONN_CSI	**46.7**	**85.0**	**64.0**	**26.7**	45.9	**42.0**	**47.1**	**54.3**	**58.8**	**55.1**	**14.4**	**48.9**	**30.6**	**46.1**	**52.7**	**58.4**	**53.4**	**31.7**	44.4	**34.5**	45.5	**42.6**
BONN_JOINT	**47.5**	**85.2**	**63.8**	**27.0**	**56.3**	**37.8**	**46.8**	**58.2**	**59.4**	**54.9**	**11.4**	**50.9**	**30.4**	**45.0**	**58.6**	**57.4**	48.6	**34.8**	**53.3**	**32.2**	**47.8**	**38.7**
BONN_LINEAR	**46.7**	**84.7**	**63.9**	**23.8**	44.8	**40.5**	**44.9**	**59.9**	**58.8**	**56.9**	**11.5**	**45.8**	**34.9**	**43.0**	**55.0**	**58.3**	**51.5**	**34.7**	44.2	29.7	**50.5**	**44.1**

Table 13: Bootstrapped segmentation results on all classes. The leading methods that are not statistically significantly different from each other are highlighted in gold.

– are there groups of images that are handled particularly well or particularly badly, and can these be characterised?

We continue this examination in Sec. 6, where we analyse in detail the types of errors occurring for each class over time.

4.1 Comparing across classes

The standard VOC 2012 classification and detection results given earlier and reproduced in the top plots of Fig. 7 and Fig. 8 show the best average precision achieved in descending order across classes. However, this does not necessarily mean that we are doing better on classes earlier in the ordering than those later in the ordering. Looking at the 'chance' results we see that a random ranking does better on some classes than others. For the person class, for example, it is easier to get a higher AP score simply because a much higher proportion of images contain people, than is the case for other classes. To overcome this bias, we need to consider a different comparison metric.

4.1.1 Comparing classification across classes

To correct for the varying proportion of positive instances in different classes, we define a *normalised* precision measure that takes this into account. This normalised measure will allow us to compare classification accuracy across classes meaningfully. It is inspired by the normalised AP measure introduced by Hoiem et al (2012) for detection.

For reference, the standard definition of precision is:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \quad (3)$$

$$= \frac{\text{TPR} \times n_{pos}}{\text{TPR} \times n_{pos} + \text{FPR} \times n_{neg}} \quad (4)$$

where TPR is the true positive rate, FPR is the false positive rate, n_{pos} is the number of positive examples, and n_{neg} the number of negative examples. As already mentioned, it is difficult to compare precisions across classes where these numbers differ because precision depends on the proportion of positive and negative images. To perform such a comparison we instead define a *normalised* precision measure:

$$\text{norm. precision} = \frac{\text{TPR} \times \bar{n}_{pos}}{\text{TPR} \times \bar{n}_{pos} + \text{FPR} \times \bar{n}_{neg}} \quad (5)$$

where $\bar{n}_{pos}$ and $\bar{n}_{neg}$ are the average number of positive and negative examples across all classes. Thus for a particular classifier threshold, computing the normalised precision measure simply involves calculating the TPR (as in Eq. 4) and using its value in Eq. 5. A normalised average precision measure for classification can be computed by averaging normalised precisions computed at a range of recalls.

The bottom plot of Fig. 7 gives the VOC 2012 classification results using this normalised measure. The first thing to note is that the 'chance' results (obtained by setting $TPR = FPR$) are now the same for all classes, showing that the normalisation has equalised the accuracy for a random classifier. In addition, the normalised results also reveal aspects of the accuracy across classes that are not clear from the original results (top plot). The biggest change is that the 'person' class drops from 2nd to 13th position in the ranking, indicating that this class is substantially harder to identify than might have been understood from the unnormalised results alone.

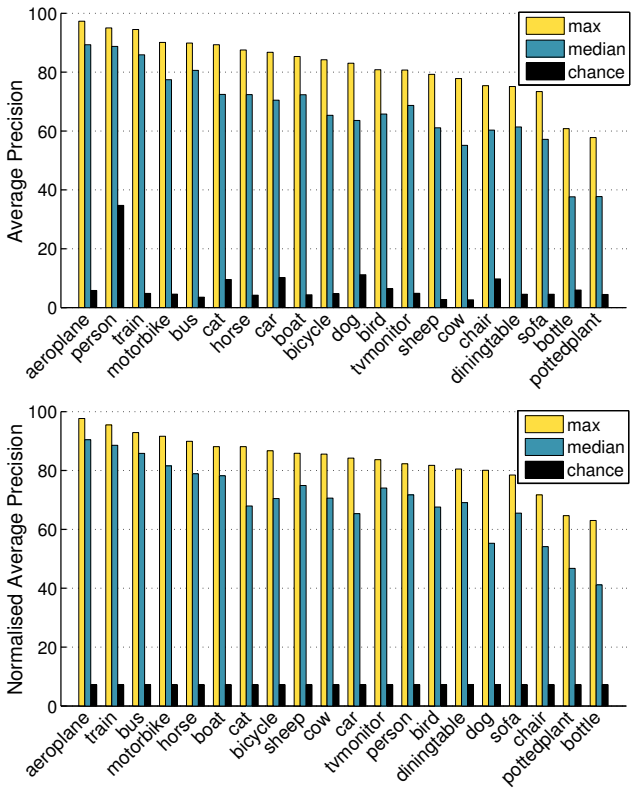


Fig. 7: Effect of normalised AP on 2012 classification results by class. *Above:* Standard AP results. *Below:* Normalised AP results. For each class three values are shown: the maximum AP obtained by any method (max), the median AP over all methods (median) and the AP obtained by a random ranking of the images (chance).

This is not the only difference: the 'chair' class now joins the 'bottle' and 'potted plant' classes in underperforming the general trend. These three classes seem to be substantially harder to recognise than the others. Other classes with large changes in rank are: 'dog' (down 5 places to 16), 'sheep' (up 5 places to 9) and 'cow' (also up 5 places to 10). However, these large changes in rank are less significant since they occur in the centre of the list where the AP figures are similar (and so small changes in AP can lead to large changes in rank).

4.1.2 Comparing detection across classes

In the detection case, the number of negatives is difficult to define, since it is hard to say the number of places where an object could be in an image, but isn't. Instead, we assume that the number of possible places is very large compared to the variation in the number of positives across classes. This allows us to assume that the number of negatives is approximately equal across classes, even when they have different numbers of positives. The result is the same as the normalised AP measure introduced in Hoiem et al (2012):

$$\text{norm. prec. det.} = \frac{\text{TPR} \times \bar{n}_{pos}}{\text{TPR} \times \bar{n}_{pos} + \text{FP}}, \tag{6}$$

where FP is the number of false positives in the class. A normalised average precision measure for detection can be computed by averaging normalised precisions computed at a range of recalls.

The detection results using this normalised measure are shown in the bottom plot of Fig. 8. In general the impact of the normalisation is less for detection than for classification, with most classes hardly changing in rank. The biggest change is once again for the 'person' class, which drops from 10th to third from last. This again suggests that the normal way of reporting results may underestimate the difficulty of detecting people in images – the state-of-the-art accuracy for detecting people is in fact only slightly better than that of detecting plants and chairs, despite all the special research effort that has gone into this case. The other two classes whose ranking drops after normalisation are 'cat' and 'dog', in both cases by three places in the ranking, reflecting the higher rate of occurrence of cats and dogs in the test set. However, this is a relatively small change in rankings compared to that of the 'person' class and should not cause us to substantially re-evaluate the difficulty of these two classes.

Finally, it is of interest to examine the difference in class ranking between classification and detection. The most dramatic difference is for the 'boat' class which is one of the better performing classes for classification but one of the worst for detection. This suggests that it is much easier to detect an image that contains a boat than to find the boat in the image. A plausible explanation for this is that the presence of an expanse of water is a good indicator of the presence of a boat. So a classifier can use this cue to infer that a boat is present, whereas it is not as helpful for a detector in precisely locating the boat.

4.2 Identifying easy and hard groups of images

In this section, we aim to identify easy and hard groups of images using the pooled results of the classification challenge for all submissions since 2009, a total of 73 submissions. The idea is to cluster both images and methods simultaneously (bi-clustering) such that methods that are clustered together tend to perform either well or badly on all the images in each image cluster.

To be more precise, the following steps were followed for each class:

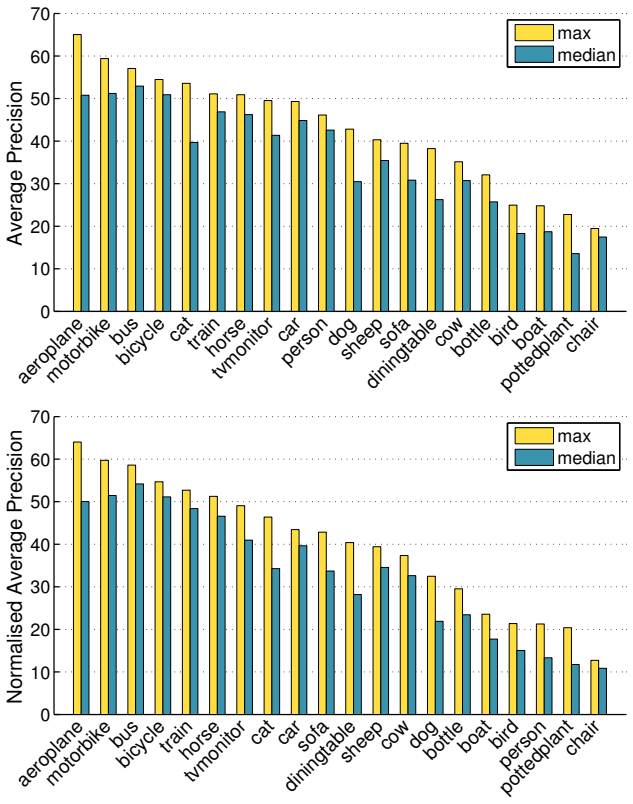


Fig. 8: Effect of normalised AP on 2012 detection results by class. *Above:* Standard AP results. *Below:* Normalised AP results. For each class two values are shown: the maximum AP obtained by any method (max) and the median AP over all methods (median).

1. Using the submitted results for each submission the test images were classified as containing or not containing the class. Since the submitted format included a continuous score for each image, these were converted into binary by choosing a threshold that gave equal numbers of false positive and false negatives.
2. Taking only the positive images (that is, images containing objects of the class) these results were formed into a binary $73 \times N$ matrix, where N is the number of positive images of that class.
3. The columns of the matrix were re-ordered so as to minimise an objective function whose main component was the sum of Hamming distances between adjacent rows/columns. In addition the objective function also contained a longer-range term that computed the Hamming distance between columns that were 20 apart, downweighted by a factor of 0.05. The minimisation was achieved using 40,000 iterations of the following greedy iterative algorithm:
   (a) Select a column or block of columns, favouring blocks that have high Hamming distance to the immediate neighbouring columns.
   (b) Move the selected columns to the location in the matrix that minimises the objective function. The block of columns can also be flipped if that further minimises the objective.
4. Apply the same algorithm to the rows of the matrix for 10,000 iterations.
5. Manually analyse the resulting matrix to identify block structures representing groups of images that are jointly either correctly or incorrectly handled by different groups of methods.

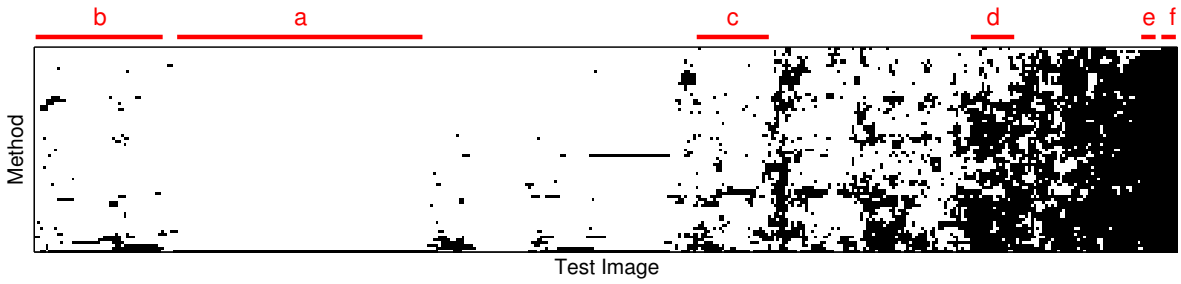
We also performed similar analysis for the detection challenge (a total of 57 submissions since 2009), except in this setting we identify easy and hard groups of object *instances* (as opposed to images). Here, the binary $57 \times N$ matrix represents whether the test instances were identified in each method's top $2N$ most confident detections, where N is the number of positive instances of that class. We do this to include the methods' lower confidence detections in our analysis.

Figs. 9–11 illustrate the resultant matrices for the three classes: aeroplane, horse and bicycle for classification, and Figs. 12–14 illustrate the resultant matrices for the three classes: bus, cat and tvmonitor for detection. For each class, six identified groups are shown, with six randomly selected images per group used to illustrate their nature.

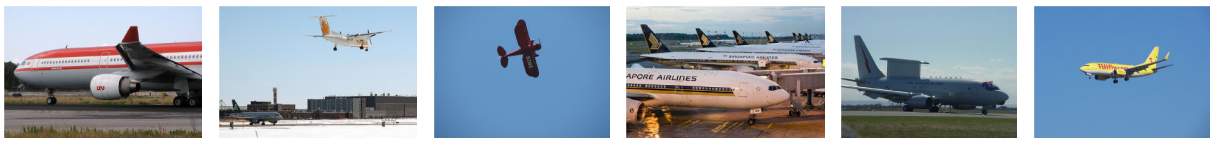
In each case the groups of images are ordered to show a steady increase in difficulty. This increase can be characterised by reduction in object size, increased truncation or occlusion, increased variation in illumination or viewpoint, increased background clutter. In each figure, the final group is of images which none of the current methods work on – it would be interesting to focus analysis and research on this group, with the aim of teasing out properties of these images which may inspire improvements to current methods.

5 Super Methods

In this section we investigate whether methods can be combined in order to obtain an algorithm with superior performance. This is a way of probing the diversity and complementarity of the submitted methods. We will use the classification task as an example, and ask the question: "can the output of the submitted methods be combined in order to build a *'super-classifier'* whose performance exceeds that of the individual methods?". This question is answered in Sec. 5.1 by constructing a super-classifier from the VOC 2012 submitted methods. In



(a) Images identified correctly by 99% of methods on average. Each image contains a large, unoccluded aeroplane seen from the side with little or no background clutter.



(b) Images identified correctly by 94% of methods on average. Slightly harder images where the aeroplane is still large but may be truncated or seen from a more unusual viewpoint. The backgrounds are still largely uncluttered.

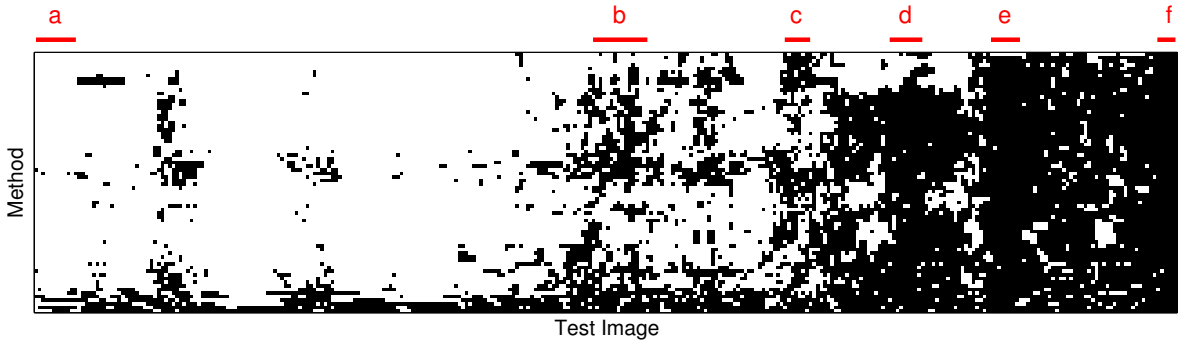
(c) Images identified correctly by 88% of methods on average. Still harder images with more cluttered backgrounds, smaller or more truncated aeroplanes and more varied lighting.

(d) Images identified correctly by 35% of methods on average. These may have unusual illumination, shadows and more extreme viewpoint variation or truncation.

(e) Images identified correctly by only the best method. The aeroplane is often small or highly truncated.

(f) Images identified correctly by none of the methods. Whilst certainly challenging, it is not immediately clear why these images are so much more challenging than the images in groups (d) and (e).

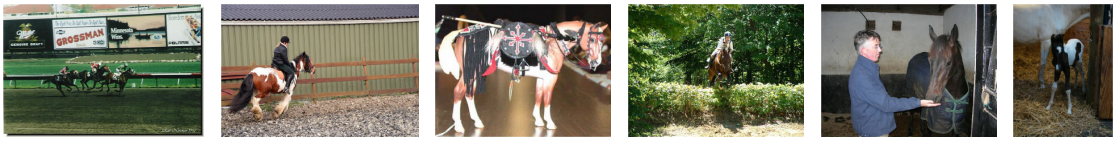
Fig. 9: Analysis of the matrix of submissions for classification: aeroplane. Each row corresponds to one of the 73 classification methods submitted since 2009. The columns correspond to the test images that contain instances of the class, with black indicating a missed classification. Six different groups of test images have been highlighted in red. (a–f) A selection of images from the different groups. The groups are ordered by increasing difficulty.



(a) Images identified correctly by 95% of methods on average. Each image contains a large, unoccluded horse with its legs clearly visible and with little background clutter.

(b) Images identified correctly by 52% of methods on average. Slightly harder images where the horse is still large but may be truncated. The backgrounds are still largely uncluttered.

(c) Images identified correctly by 42% of methods on average. Still harder images with more cluttered backgrounds, smaller or more truncated objects and varied poses.

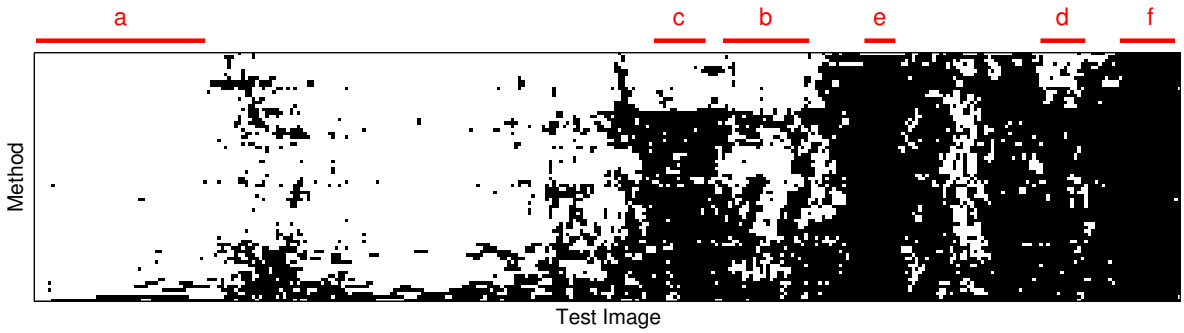


(d) Images identified correctly by 13% of methods on average. These images sometimes have unusual illumination and horses appear in more varied poses.

(e) Images identified correctly by only the best method. The horse is often small, highly truncated or in a highly unusual pose.

(f) Images identified correctly by none of the methods, typified by small horse sizes and large amounts of occlusion and truncation.

Fig. 10: Analysis of the matrix of submissions for classification: horse. Each row corresponds to one of the 73 classification methods submitted since 2009. The columns correspond to the test images that contain instances of the class, with black indicating a missed classification. Six different groups of test images have been highlighted in red. (a–f) A selection of images from the different groups. The groups are ordered by increasing difficulty.



(a) Images identified correctly by 97% of methods on average. Each image contains a large, unoccluded bicycle seen from the side with little background clutter.

(b) Images identified correctly by 52% of methods on average. Slightly harder images where the bicycle is smaller and may be seen from the rear or front.

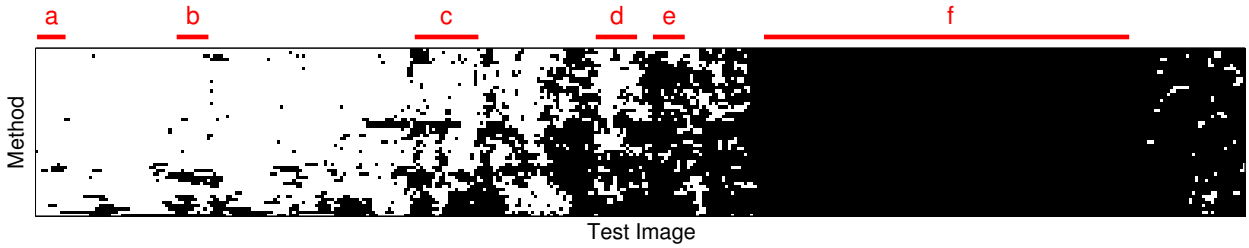
(c) Images identified correctly by 27% of methods on average. Still harder images with more cluttered backgrounds and smaller bicycles.

(d) Images identified correctly by 19% of methods on average. These images sometime have very small or truncated bicycles or they are in unusual poses.

(e) Images identified correctly by only the best method. The bicycle is often very small or highly truncated.

(f) Images identified correctly by none of the methods.

Fig. 11: Analysis of the matrix of submissions for classification: bicycle. Each row corresponds to one of the 73 classification methods submitted since 2009. The columns correspond to the test images that contain instances of the class, with black indicating a missed classification. Six different groups of test images have been highlighted in red. (a–f) A selection of images from the different groups. The groups are ordered by increasing difficulty.



(a) Instances identified correctly by 90% of methods on average. The bus is large and in a canonical pose.

(b) Instances identified correctly by 83% of methods on average. Slightly harder images where the bus is still large but in more varied poses.

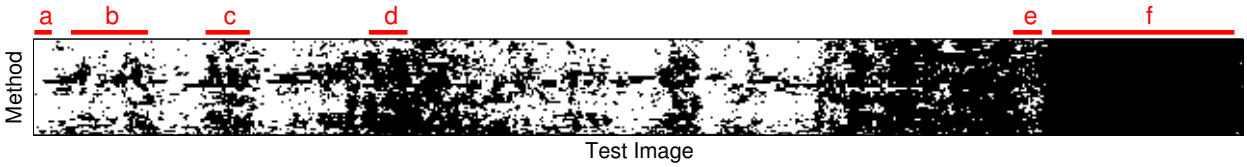
(c) Instances identified correctly by 47% of methods on average. Harder images with more cluttered backgrounds, and smaller buses that are sometimes truncated.

(d) Instances identified correctly by 30% of methods on average. Instances are smaller still and often occluded.

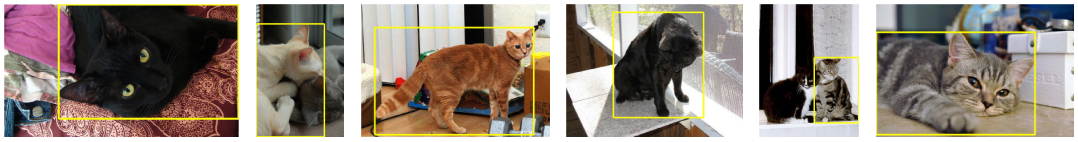
(e) Instances identified correctly by 10% of methods on average. The buses are highly truncated and occluded.

(f) Instances identified correctly by none of the methods.

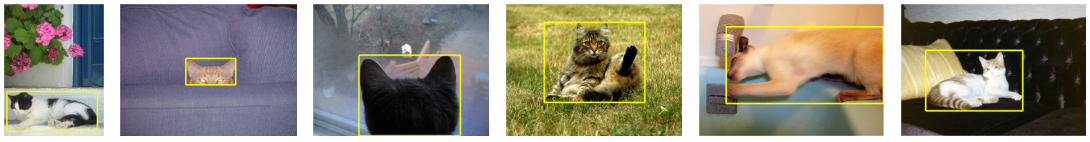
Fig. 12: Analysis of the matrix of submissions for detection: bus. Each row corresponds to one of the detection methods submitted since 2009. The columns correspond to the instances of objects of the class, with black indicating a missed detection. Six different groups of test images have been highlighted in red. (a-f) A selection of images from the different groups. The groups are ordered by increasing difficulty.



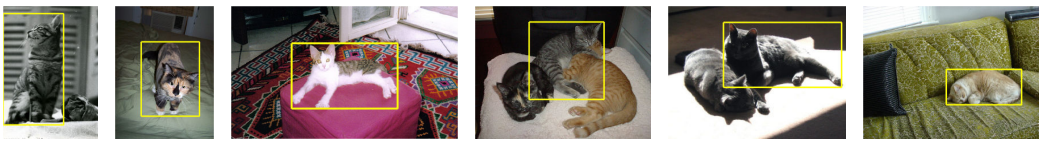
(a) Instances identified correctly by 96% of methods on average. In each instance the cat's face is clearly visible.



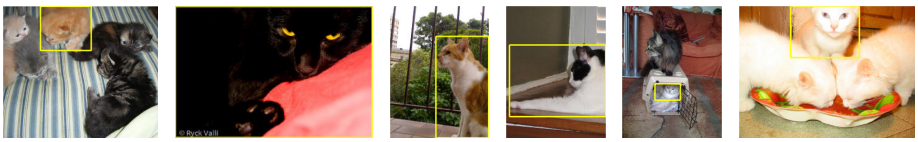
(b) Instances identified correctly by 69% of methods on average. Slightly harder images where the cat is still large but it appears in more varied poses.



(c) Instances identified correctly by 42% of methods on average. Harder images where the cat is sometimes truncated or occluded.



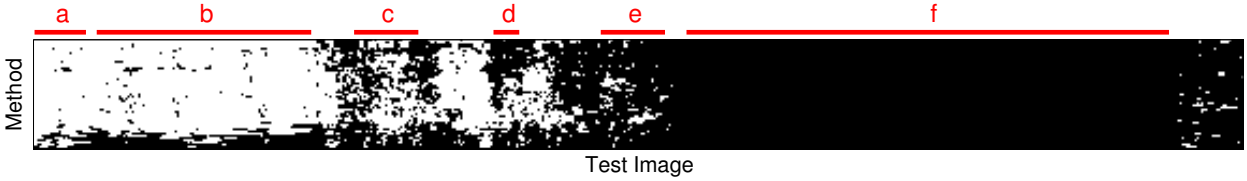
(d) Instances identified correctly by 20% of methods on average. These images have more cluttered backgrounds and often contain multiple cats in the same image.

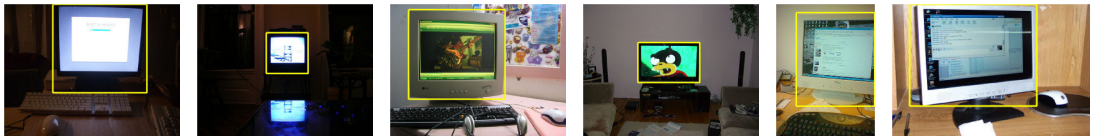


(e) Instances identified correctly by 18% of methods on average. The cats are smaller, rarely facing the camera and at times highly truncated.

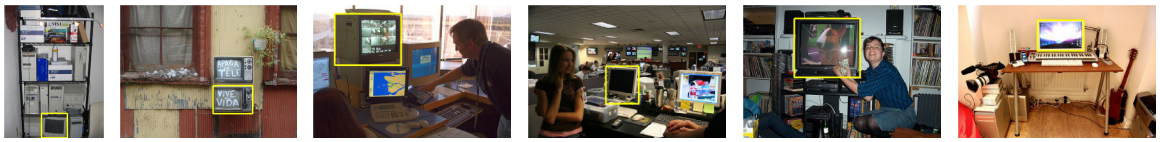
(f) Instances identified correctly by none of the methods.

Fig. 13: Analysis of the matrix of submissions for detection: cat. Each row corresponds to one of the detection methods submitted since 2009. The columns correspond to the instances of objects of the class, with black indicating a missed detection. Six different groups of test images have been highlighted in red. (a-f) A selection of images from the different groups. The groups are ordered by increasing difficulty.





(a) Instances identified correctly by 96% of methods on average. The tv or monitor screen is large, facing the camera and often brightly lit.



(b) Instances identified correctly by 85% of methods on average. Slightly harder images where the screen is slightly smaller and the background is slightly more cluttered.

(c) Instances identified correctly by 65% of methods on average. Harder images with more screens in each image that are sometimes truncated.

(d) Instances identified correctly by 49% of methods on average. The instances are typically highly truncated.

(e) Instances identified correctly by 14% of methods on average. The screens sometimes face away from the camera.

(f) Instances identified correctly by none of the methods.

Fig. 14: Analysis of the matrix of submissions for detection: tvmonitor. Each row corresponds to one of the detection methods submitted since 2009. The columns correspond to the instances of objects of the class, with black indicating a missed detection. Six different groups of test images have been highlighted in red. (a-f) A selection of images from the different groups. The groups are ordered by increasing difficulty.

Sec. 5.2 we examine the learnt parameters of the super-classifier to determine how the methods are weighted, and if particular methods are used for all classes. Finally, in Sec. 5.3, we use this combination scheme to identify *pairs* of methods that tend to provide complementary information for a given classification task. The result of this indicates which sites should collaborate to best improve their methods.

5.1 Is it possible to improve on the best submission by using a combination of submissions?

The method we investigate for the super-classifier is a linear classifier for each of the VOC classes, where the feature vector consists of the real-valued scores supplied by each submitted method. We have these scores available for all of the VOC test images, though not for the VOC training or validation images. For this reason the investigation is carried out on the VOC test data, for which we have the ground truth labels.

5.1.1 Training, test data, and evaluation

The 10991 images and labels that form the VOC 2012 `test` dataset are used both as training data for the super-classifier, and as test data to evaluate the super-classifier's performance.

The images are separated into two sets of approximately equal size in a stratified manner (i.e. the number of positive and negative training images in each set are roughly equal). The super-classifier is trained on one set and its performance is tested on the other set. The experiments are then repeated by switching the train and test datasets, and the method is evaluated as the average AP across the two folds. To ensure a fair comparison, the same two-fold evaluation is also computed for the individual methods. Despite the stratification the difference between the AP computed by averaging the two folds and that computed on all the test data can be as high as 2.40% AP; see Fig. 15 for the precision-recall curves of a single method on the 'boat' class.

5.1.2 Data preparation and classifier training

The feature vector $\mathbf{x}_i$ that is used to predict the presence or absence of a VOC class c in the ith image consists of the M real-valued scores x_{im} submitted by each of the methods for that image, i.e. it is an M dimensional vector. The scores x_{im} are linearly scaled to ensure that the range of values spanned by each particular method m in the training data is between -1 and 1. The same linear scaling is also applied to the feature vectors of the test data.

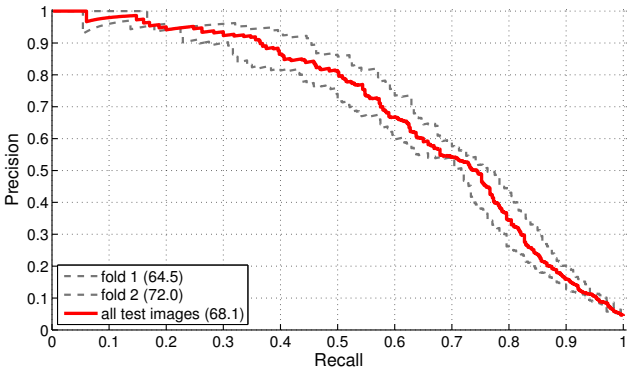


Fig. 15: Fold variance for super-classifier training/test data. Precision-recall curve for a single method on the 'boat' class. The two grey curves are the method's performances when evaluated on either half of the test data, and the red curve for when evaluated on all the test data.

For each VOC class a binary classifier is trained to predict whether the image contains that object class or not. We use a support vector machine (SVM) with a linear kernel, trained using LIBSVM (Chang and Lin, 2011). An optimal value for the parameter C is first found by optimising classification performance on a held-out validation set (10% of the training dataset), and then the SVM is trained on all of the training data.

5.1.3 Results

Fig. 16 shows, for each object class, the improvement made by the super-classifier over the best performing method at that class. The super-classifier outperforms any individual method on 17 out of 20 object classes. On classes where it performs better, it boosts performance by 4.40% AP on average. The highest increase in performance is on the 'potted plant' class, where the super-classifier improves performance by 13.00% AP. Where the super-classifier performs worse, the performance drops by 0.78% AP on average. The average performance difference across all classes was found to be +3.62% AP. We compare the super-classifier's precision-recall curves with those of the other submissions in Fig. 17 for the 'person' and 'potted plant' categories.

5.2 How much does each method contribute to the super-classifier?

To investigate the role that the scores predicted by each of the methods plays in training the super-classifier, we examine the *magnitude* of the SVM's weights for each of its M input dimensions. In this case we do not intend

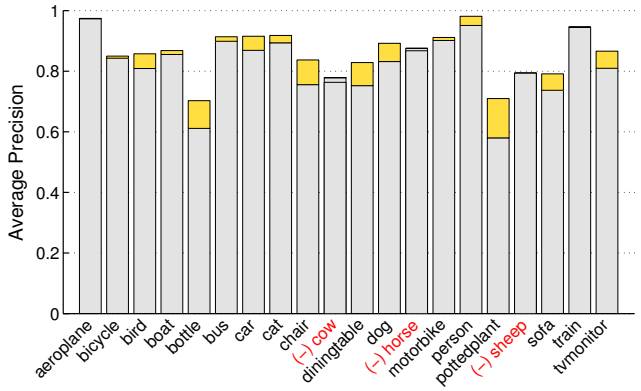
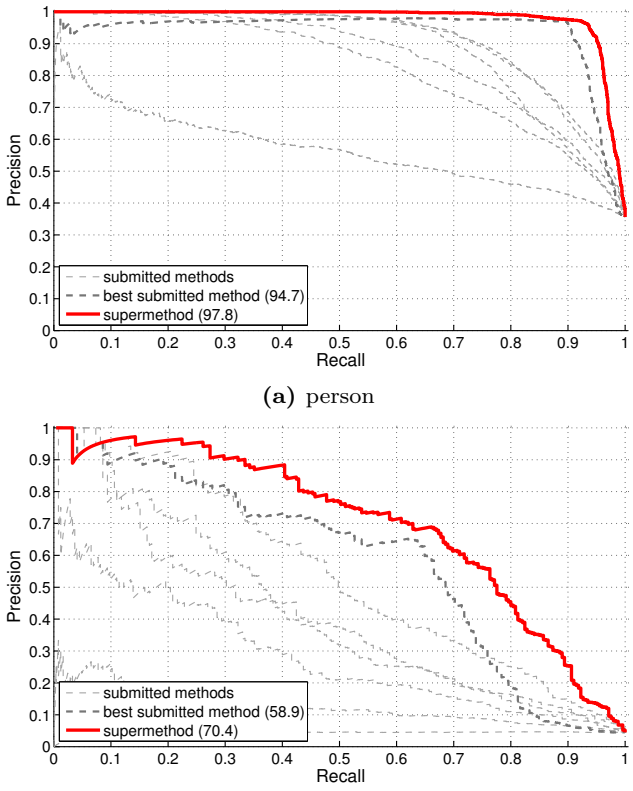


Fig. 16: AP increase of super-classifier. Gold indicates the increase in performance on object classes where the super-classifier beats all other methods, averaged across the two folds. The three classes for which performance drops slightly have been highlighted in red.



(a) person

(b) pottedplant

Fig. 17: PR curves for super-classifier. Precision-recall curves for the submitted methods along with that of the super-classifier for one fold. The best performing submitted method is drawn thicker.

to test the performance of the trained SVM but only to inspect its weight vector, so it is trained on all 10991 feature vectors. As before, an optimal value for the C parameter is found via cross-validation on 10% of the data.

Fig. 18 displays the magnitude of the learnt weights for the scores predicted by each submitted method for the 'aeroplane', 'bicycle', 'person' and 'sofa'classes. Surprisingly, the influence of the different methods varies significantly across object classes. Also note that the method with the best performance is not necessarily the strongest influence on the performance of the super-classifier (as indicated by the PR curves).

5.3 Who should collaborate?

Classifier combination can also be used to identify pairs of methods which make predictions that provide complementary information about each image in question. It might be beneficial for the community if such complementary methods incorporate each other's ideas.

In order to find these so-called 'complementary' pairs, we perform two experiments. In the first, we create super-classifiers as above, but using the predictions of only two submitted methods as input features at a time. For each object class, we select the pair of methods that achieve the highest combined performance.

In the second experiment we proceed as above, except that we report the pair of methods whose combined performance maximises $S^{1+2} = \min(AP^{1+2}/AP^1, AP^{1+2}/AP^2)$, where AP^1 and AP^2 are the recorded APs for each of the two methods, and AP^{1+2} is the performance of the super-classifier resulting from their combination. This measure ensures that the combination boosts the performance of both methods relative to their individual performances.

We report the results of these two experiments in Tables 14 and 15. In Table 14, we note that the performance of the super-classifier trained using only scores from two methods is often significantly higher than the best-performing individual method (e.g. for 'bottle', 'chair' and 'potted plant'), and for some classes higher than that of the super-classifier trained for all methods together (e.g. 'bicycle'). For all classes NUS_SCM is one of the chosen collaborators, which is not entirely surprising given its dominating individual performance (see Table 5).

From Table 15, we note that for several classes a large relative increase in performance can be obtained by combining two moderately performing methods.

5.4 Discussion

We have illustrated the idea of combining the output from submitted methods for the case of the classifica-

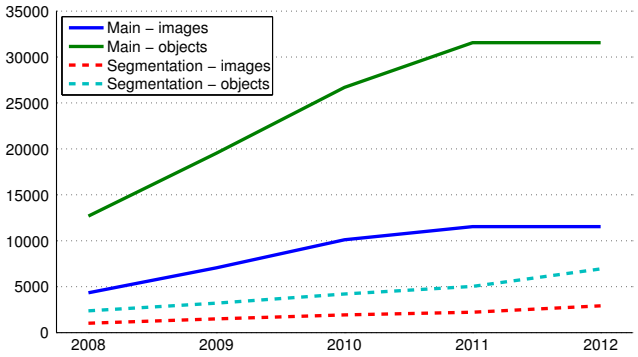


Fig. 19: Summary of the statistics of the main VOC dataset over the years.

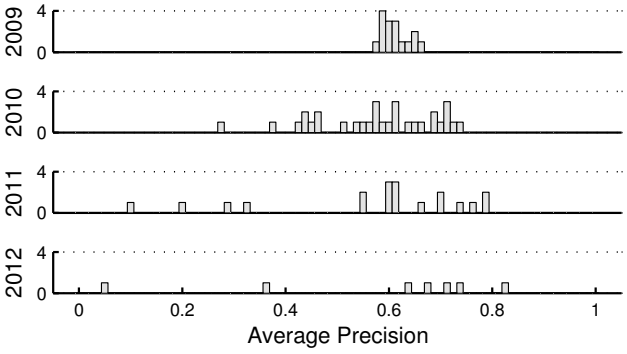


Fig. 20: Overall classification AP over the years. The histograms indicate the number of methods whose mean AP score fell in specific ranges in each year.

tion challenge. A similar approach could be carried out for the other main challenges, detection and segmentation, and we discuss possible solutions here.

For segmentation we require a VOC class (or background) prediction for each pixel (i.e. one of 21 choices) in each test image, and we have available the pixel level class predictions for all the submitted methods. Note, we do not have confidence scores for the submitted methods. There are various super-segmenters that could then be learnt. The simplest would be to take a majority vote for each pixel across all the submitted methods. This does not require the ground truth annotation and essentially involves no learning. An alternative is to learn a multi-way classifier using the class predictions of the submitted methods as the feature vector. In this case, if there are M submitted methods, then the feature vector is a $21M$ dimensional binary vector, where each 21 dimensional block arises from one method and only contains one non-zero entry. There is then a choice of multi-way classifiers; for example a random forest or softmax logistic regression. There is also a choice in how to sample the training data – whether all pixels are used, and whether the classes are balanced.

For detection the combination of predictions is more difficult, as there is a need to first identify which detections from the different methods are referring to the same predicted object (e.g. based on overlaps between bounding boxes), before combining them. This generalises the problem of non-maximum suppression for a single predictor (see e.g. Dalal and Triggs, 2005; Viola and Jones, 2004; Felzenszwalb et al, 2010; Leibe et al, 2004) to the output of multiple predictors.

6 Progress Through Time

In this section we examine trends over the 2008–2012 timespan of the challenges: trends in the datasets themselves, trends in the performance of the algorithms, and trends in the methods employed.

The growth in the amount of data in the main and segmentation datasets is shown graphically in Fig. 19, and in detail in Tables 16 and 17. Between 2008 and 2012 the total number of images (objects) in the main dataset more than doubled from 4332 (12684) to 11540 (31561). Over the same period the number of images (objects) in the segmentation dataset almost trebled from 1023 (2369) to 2913 (6934). For action classification, training and testing cases went up by about 90% from 2011 to 2012. Indeed, by 2012 there were 4588 training and validation images, with about 400 examples of people carrying out each class of action. For the person layout taster, the number of images (objects) similarly increased from 245 (367) in 2008 to 609 (850) in 2012.

To determine the trends in performance, we evaluate all submissions from 2009 to 2012 using the images in the VOC 2009 test dataset. All submissions since 2009 have provided predictions for the VOC 2009 test images, since images were only *added* to the test set over the years; thus this dataset provides a common test set for all submissions. For detection, we also analyse the submissions using the methods of Hoiem et al (2012), and thereby identify the types of errors occurring across time.

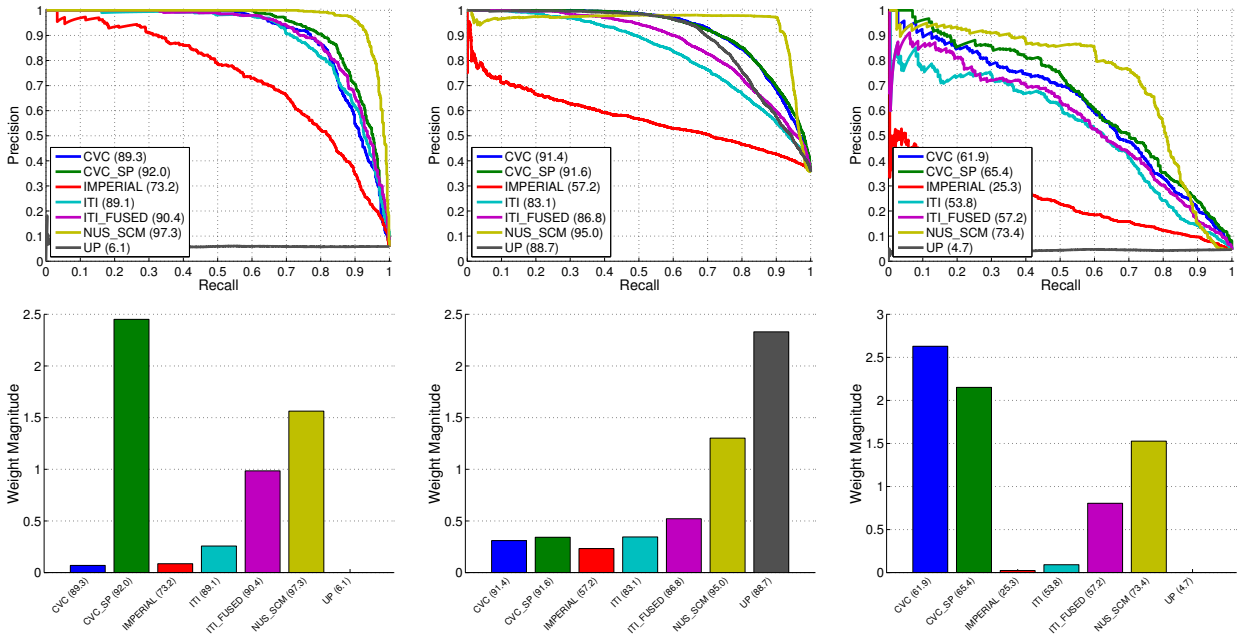
We consider the trends for the classification, detection and segmentation tasks in order. As will be seen, originally these tasks were treated quite independently by participants, but as the challenge progressed there was progressively more cross-fertilisation, with detection and segmentation being used to enhance the classification performance for example.

Class	Top	Method 1		Method 2		Pair combined	All combined
	AP	Name	AP^1	Name	AP^2	AP^{1+2}	AP
aeroplane	97.34	UP	6.00	NUS_SCM	97.34	97.34	**97.41**
bicycle	84.28	ITI_FUSED	65.35	NUS_SCM	84.28	**85.30**	84.98
bird	80.89	CVC	69.78	NUS_SCM	80.89	85.09	**85.73**
boat	85.52	CVC_SP	77.53	NUS_SCM	85.52	86.38	**86.80**
bottle	61.12	CVC_SP	54.37	NUS_SCM	61.12	69.94	**70.27**
bus	89.86	CVC	84.79	NUS_SCM	89.86	91.12	**91.38**
car	86.87	CVC_SP	81.90	NUS_SCM	86.87	**93.07**	91.54
cat	89.37	CVC_SP	76.54	NUS_SCM	89.37	**92.19**	91.81
chair	75.56	CVC_SP	65.21	NUS_SCM	75.56	83.49	**83.70**
cow	77.88	UP	3.93	NUS_SCM	77.88	**77.96**	76.35
diningtable	75.24	CVC_SP	68.59	NUS_SCM	75.24	82.26	**82.86**
dog	83.19	CVC_SP	68.94	NUS_SCM	83.19	**89.25**	89.22
horse	87.53	ITI_FUSED	72.39	NUS_SCM	87.53	**88.84**	86.73
motorbike	90.14	CVC	79.21	NUS_SCM	90.14	90.78	**91.14**
person	95.11	CVC_SP	91.62	NUS_SCM	95.11	97.87	**98.14**
pottedplant	57.99	CVC_SP	56.24	NUS_SCM	57.99	70.11	**70.99**
sheep	79.34	IMPERIAL	23.86	NUS_SCM	79.34	**79.48**	79.34
sofa	73.69	ITI_FUSED	57.42	NUS_SCM	73.69	78.56	**79.12**
train	94.49	UP	5.10	NUS_SCM	94.49	94.49	**94.62**
tvmonitor	80.95	CVC_SP	77.37	NUS_SCM	80.95	85.75	**86.61**

Table 14: Collaboration table from super-classifiers I. Collaborator 1 and 2 (C^1 and C^2) chosen to maximise AP^{1+2}, the performance of super-classifier resulting from the combination of C^1 and C^2. Note that the performance of the super-classifier trained using only scores from two methods as input features (second to last column) is often significantly higher than the best-performing individual method (second column), and for some classes higher than that of the super-classifier trained all methods together (last column). NUS_SCM, the best-performing individual method across all classes, is always chosen to be one of the super-classifier collaborators.

Class	Top	Method 1		Method 2		Pair combined	All combined
	AP	Name	AP^1	Name	AP^2	AP^{1+2}	AP
aeroplane	97.34	ITI	89.09	CVC	89.29	91.03	**97.41**
bicycle	84.28	ITI_FUSED	65.35	NUS_SCM	84.28	**85.30**	84.98
bird	80.89	CVC	69.78	NUS_SCM	80.89	85.09	**85.73**
boat	85.52	ITI_FUSED	72.39	CVC	73.91	74.77	**86.80**
bottle	61.12	CVC_SP	54.37	NUS_SCM	61.12	69.94	**70.27**
bus	89.86	CVC	84.79	NUS_SCM	89.86	91.12	**91.38**
car	86.87	CVC_SP	81.90	NUS_SCM	86.87	**93.07**	91.54
cat	89.37	CVC_SP	76.54	NUS_SCM	89.37	**92.19**	91.81
chair	75.56	CVC_SP	65.21	NUS_SCM	75.56	83.49	**83.70**
cow	**77.88**	ITI_FUSED	55.37	CVC	59.39	60.16	76.35
diningtable	75.24	CVC_SP	68.59	NUS_SCM	75.24	82.26	**82.86**
dog	83.19	CVC_SP	68.94	NUS_SCM	83.19	**89.25**	89.22
horse	87.53	ITI_FUSED	72.39	NUS_SCM	87.53	**88.84**	86.73
motorbike	90.14	ITI_FUSED	77.39	CVC	79.21	80.68	**91.14**
person	95.11	UP	88.74	ITI_FUSED	86.78	91.79	**98.14**
pottedplant	57.99	CVC_SP	56.24	NUS_SCM	57.99	70.11	**70.99**
sheep	**79.34**	ITI_FUSED	61.04	CVC	63.09	63.94	**79.34**
sofa	73.69	ITI_FUSED	57.42	NUS_SCM	73.69	78.56	**79.12**
train	94.49	ITI_FUSED	85.83	CVC	86.77	87.37	**94.62**
tvmonitor	80.95	CVC_SP	77.37	NUS_SCM	80.95	85.75	**86.61**

Table 15: Collaboration table from super-classifiers II. Collaborator 1 and 2 (C^1 and C^2) chosen to maximise S^{1+2}, where $S^{1+2} = \min(AP^{1+2}/AP^1, AP^{1+2}/AP^2)$, and AP^{1+2} is the performance of super-classifier resulting from the combination of C^1 and C^2. For comparison, the best performance of a single method is shown in the Top AP column.



(a) aeroplane **(b)** person **(c)** sofa

Fig. 18: Super-classifier weight vectors and AP. Precision-recall curves and bars are colour-coded to indicate matching methods. The influence of the methods varies significantly for different object classes. Also note that the strongest influence on the performance of the super-classifier is not necessarily the method with the best performance. Method UP is only used in the 'person' class since it only participated in that class.

		aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	total
2008	**Images**	236	192	305	207	243	100	466	328	351	74	105	388	198	204	2002	180	64	134	151	215	4332
	Objects	370	307	549	453	575	150	1022	380	910	197	218	484	306	297	5070	436	234	251	176	299	12684
2009	**Images**	407	348	505	325	420	258	730	543	668	172	271	649	328	338	2779	332	131	308	324	353	7054
	Objects	587	506	833	654	905	386	1499	624	1716	403	414	790	503	494	6717	702	427	447	391	541	19539
2010	**Images**	579	471	666	432	583	353	1030	1005	925	248	415	1199	425	453	3548	450	290	406	453	490	10103
	Objects	814	663	1061	824	1186	539	2060	1142	2339	553	587	1439	653	644	8563	911	847	611	541	714	26691
2011	**Images**	670	552	765	508	706	421	1161	1080	1119	303	538	1286	482	526	4087	527	325	507	544	575	11540
	Objects	954	790	1221	999	1482	637	2364	1227	2906	702	747	1541	750	751	10129	1099	994	786	656	826	31561
2012	**Images**	670	552	765	508	706	421	1161	1080	1119	303	538	1286	482	526	4087	527	325	507	544	575	11540
	Objects	954	790	1221	999	1482	637	2364	1227	2906	702	747	1541	750	751	10129	1099	994	786	656	826	31561

Table 16: Statistics of the main VOC dataset over the years. For the `trainval` dataset, the number of images containing at least one object of the given class, and the corresponding number of object instances are shown. Note that because images may contain objects of several classes, the totals shown in the images rows are not simply the sum of the corresponding row.

		aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonitor	total
2008	**Images**	58	50	59	60	58	52	80	65	89	44	67	65	67	59	343	60	47	60	57	69	1023
	Objects	67	68	77	87	96	71	153	73	188	89	71	82	78	75	619	128	121	72	66	88	2369
2009	**Images**	87	77	107	87	86	77	114	98	124	66	88	101	92	83	417	88	61	97	86	99	1499
	Objects	101	101	138	123	136	107	190	116	260	129	92	123	117	100	720	163	152	117	98	128	3211
2010	**Images**	116	97	136	106	117	99	161	169	170	82	104	156	105	103	526	115	83	117	106	113	1928
	Objects	136	124	176	153	176	147	276	195	348	161	108	194	136	126	925	209	209	138	118	148	4203
2011	**Images**	131	110	166	117	144	113	187	190	210	89	120	185	111	112	632	127	93	144	125	126	2223
	Objects	158	144	214	176	259	166	345	219	431	186	127	224	149	139	1170	229	232	165	138	163	5034
2012	**Images**	178	144	208	150	183	152	255	250	271	135	157	249	147	157	888	167	120	183	167	157	2913
	Objects	218	197	277	232	357	237	457	286	550	284	168	299	204	204	1738	322	308	209	189	198	6934

Table 17: Statistics of the segmentation dataset over the years. For the `trainval` dataset, the number of images containing at least one object of the given class, and the corresponding number of object instances are shown. Note that because images may contain objects of several classes, the totals shown in the images rows are not simply the sum of the corresponding row.

6.1 Classification

In Fig. 20 we plot histograms of the mean AP scores achieved on the classification task by the various methods in the different years. The diversity in performances has gradually increased, as has the highest AP achieved. We also plot, for each class, the AP of the best-performing method in each year on that class in Fig. 21. This shows improved performance between 2009 and 2012 for all classes, although these increases are not always monotonic over the intervening years.

The basic classification method that was dominant in VOC 2007 was the bag-of-visual-words (Csurka et al, 2004). Local features were extracted (e.g. SIFT descriptors, Lowe 2004), vector quantised into a visual vocabulary (e.g. using k-means), and each image represented by histograms of how often the extracted features were assigned to each visual word. A support vector machine (SVM) classifier was then used on top of this histogram representation.

One idea to go beyond the simple bag-of-words (BOW) representation was to use histograms on regions of the image, e.g. spatial pyramid matching (SPM, Lazebnik et al 2006), where representations for a nested pyramid of regions are computed.

An alternative approach is classification-by-detection, making use of the output of a classifier looking at a specific region of the image (e.g. a sliding window detector), and combining the results of these detections.

Over 2008–2012 there were various developments of these methods. For example the winning NEC/UIUC entry in 2009 used local coordinate coding (LCC, Yang et al 2009) to replace the vector quantisation step, so that a feature vector could be represented by more than one template. In 2010 the winning entry from NUS/PSL used multiple kernels to combine BOW-type representations with the outputs of the object detector due to Felzenszwalb et al (2010). The 2011 entry from the University of Amsterdam used classification-by-detection, but instead of using sliding windows they proposed candidate windows based on multiple segmentations (Van de Sande et al, 2011). The 2012 winning entry from NUS/PSL was described in Sec. 3.1.

We note that recent work by Krizhevsky et al (2012) on deep convolutional neural networks (CNNs) has significantly outperformed methods similar to those described above on the large scale ImageNet classification challenge. These networks, trained on ImageNet, have subsequently been applied to VOC classification in two different ways: the first is to 'remove' the final classification layer of the CNN, and use the remaining architecture to compute image level features; the second is to use the ImageNet data as supervised pre-training, and then refine the network by training with VOC data. Both have led to impressive performance improvements on the VOC classification task (Donahue et al, 2013; Oquab et al, 2014; Zeiler and Fergus, 2013). This is another important development in the image level feature learning/encoding story.

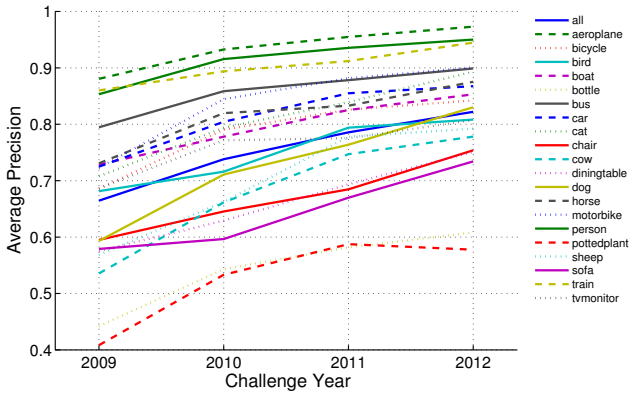


Fig. 21: Classification AP by class over the years. For each year and class we plot the AP obtained by the best-performing method at that class in that year.

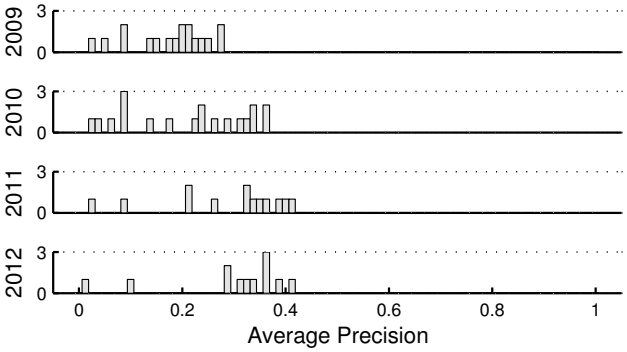


Fig. 22: Overall detection AP over the years. The histograms indicate the number of methods whose mean AP score fell in specific ranges in each year.

6.2 Detection

In Fig. 22 we plot histograms of the mean AP scores achieved on the detection task by the various methods in the different years. Again the diversity in performances has gradually increased, as has the highest AP achieved. We also plot, for each class, the AP of the best-performing method in each year on that class in Fig. 23. For most classes the AP of the best performing method has gradually increased over the years, although this trend is less strong than for the classification task (see Fig. 21).

The reference method for object detection in VOC 2008–2012 was the deformable part model (DPM; Felzenszwalb et al, 2010). The method is based on a coarse scale 'root' filter using a histogram of oriented gradients representation (HOG; Dalal and Triggs, 2005), plus finer-scale HOG part templates that can move relative to the root. This model is applied everywhere in the image using efficient sliding windows. The outputs are post-processed involving regression to predict the bounding box from the root and part locations, greedy non-maximum suppression, and rescoring of each bounding box with an SVM using contextual information about the maximal strengths of detections for all classes. Teams led by Felzenszwalb were joint winners of the challenge in 2008 and 2009, and the released code meant that this method was widely used/developed by others.

The team from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences which won in 2010 enhanced the DPM method by incorporating local binary pattern (LBP) features in addition to HOG, and including spatial, global and inter-class context into the post-processing SVM.

Also in 2010, the MIT/UCLA team (Zhu et al, 2010) extended the DPM model by allowing a 3-layer architecture (corresponding to a root, parts and sub-parts), and by incorporating data terms involving a histogram of 'visual words' for region appearance as well as HOG features for edges.

One issue with the DPM is that it performs search densely in the image, and uses relatively simple matches between the HOG input representation and filters. In VOC 2009, Vedaldi et al (2009) used a cascade of methods of increasing complexity using linear, quasi-linear and non-linear kernels, with the idea that the more expensive non-linear methods need only be run on a subset of locations determined by the cheaper methods. In the VOC 2011 and 2012 contests this group instead re-scored the top 100 candidates output by the DPM, using a SVM with inputs including additional features, geometry, and context (described in more detail in Section 3.2).

Another way of avoiding sliding windows search is to hypothesise bounding boxes bottom up, e.g. based on multiple segmentations (Van de Sande et al, 2011). This is the route, termed 'selective search', followed by the University of Amsterdam entry which won in 2012, and allowed them to use more computationally expensive features and classifiers.

In more recent work (Girshick et al, 2014), the CNN features described in Section 6.1 have been used to represent the selective search bounding boxes, and have achieved a substantial improvement in detection performance.

6.2.1 Patterns of errors through time

Hoiem et al (2012) produced a very interesting analysis of patterns of false-positive and false-negative errors made by two different detection algorithms on the Pas-

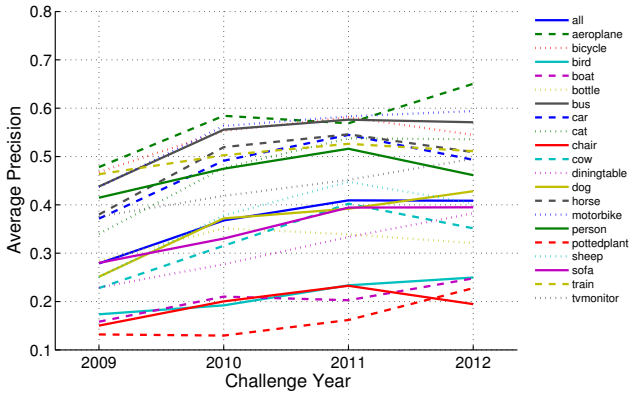


Fig. 23: Detection AP by class over the years. For each year and class we plot the AP obtained by the best-performing method at that class in that year.

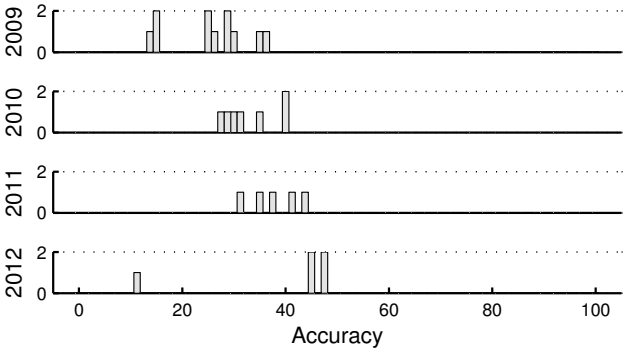


Fig. 26: Overall segmentation accuracy over the years. The histograms indicate the number of methods whose mean segmentation accuracy score fell in specific ranges in each year.

cal VOC 2007 dataset (for which ground truth is also publicly available for the test set). We have carried out a similar analysis on the top three performing methods from both 2009 and 2012.

For false positives (FPs), Hoiem et al (2012, Sec. 2) looked at four kinds of errors. These are: *localisation*, where the target category is detected with a misaligned bounding box (with overlap between 0.1 and 0.5, while the threshold for a correct detection is set at 0.5); *confusion with similar objects*, where the groups of similar objects are taken to be {all vehicles}, {all animals including person}, {chair, diningtable, sofa}, {aeroplane, bird}; *confusion with other VOC objects*, describes remaining false positives which have at least 0.1 overlap with an object with a non-similar label; all other FPs are classified as *confusion with background*. Let the number of true positives of class j in the test set be N_j. Following Hoiem et al (2012), we consider the top ranked N_j detections for class j, and compute the percentage that are correct, and the four kinds of errors (localisation, similar objects, dissimilar objects, background). These results are plotted as pie charts by Hoiem et al (2012, Fig. 2).

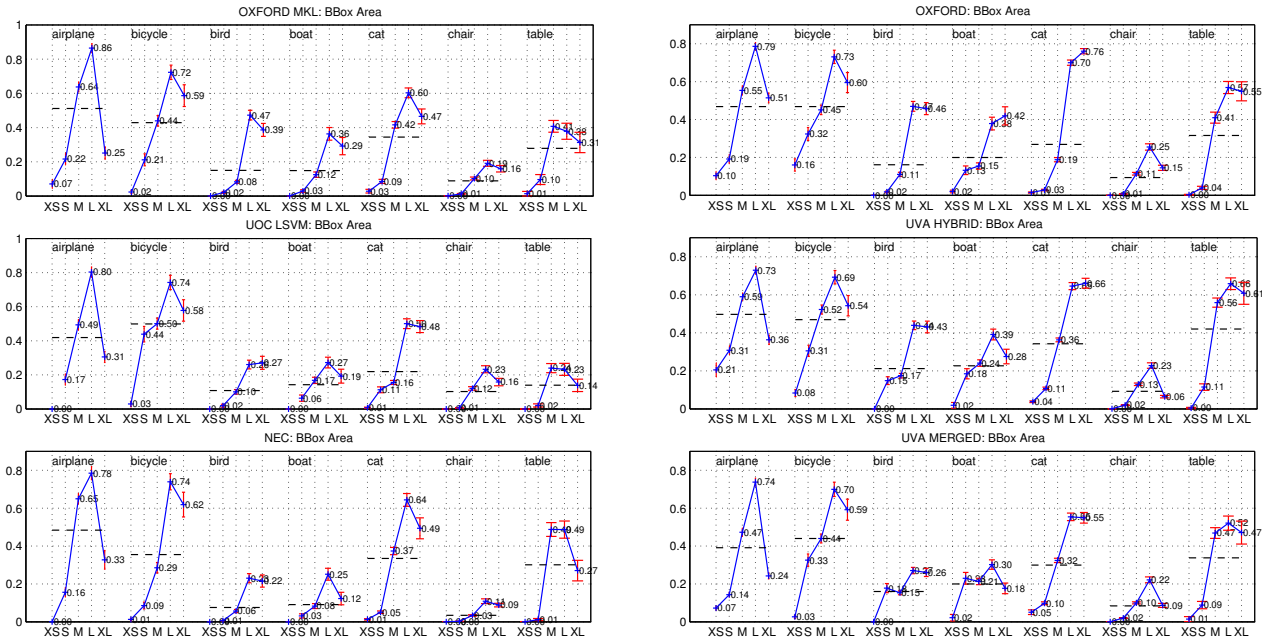
Figure 25 plots these results for three groups (animals, vehicles, furniture) and four individual classes from both 2009 and 2012. For 2012, the three top performing methods were OXFORD, UVA_HYBRID, and UVA_MERGED. There is a marked trend that the percentage of background errors has increased between 2009 and 2012, while in general the confusions with similar and dissimilar objects have decreased.

Following Hoiem et al (2012) we also examined the impact of object characteristics (object size, aspect ratio and truncation) on false negatives. Object size was measured by the bounding box area. Objects in each class were partitioned into five size categories, depending on the object's percentile size within its category: extra-small (XS: bottom 10%), small (S: next 20%), medium (M: next 40%); large (L: next 20%) and extra-large (XL: top 10%). For aspect ratio (defined as bounding box width divided by height), objects were categorised as extra-tall, tall, medium, wide and extra-wide, using the same percentiles. For truncation, the Pascal VOC annotation of truncated/not-truncated was used. For object size the results are plotted in Fig. 24. For all classes we observe trends similar to those noted by Hoiem et al (2012) in that the normalised precision (see Eq. 6) increases as function of size, except that there is sometimes a drop for extra-large objects, which are often highly truncated. In general (data not shown) performance with respect to aspect ratio is better for less-extreme aspect ratios, and it is better for non-truncated objects than truncated ones (except that the top three methods in 2009 and 2012 all prefer truncated over non-truncated cats).

6.3 Segmentation

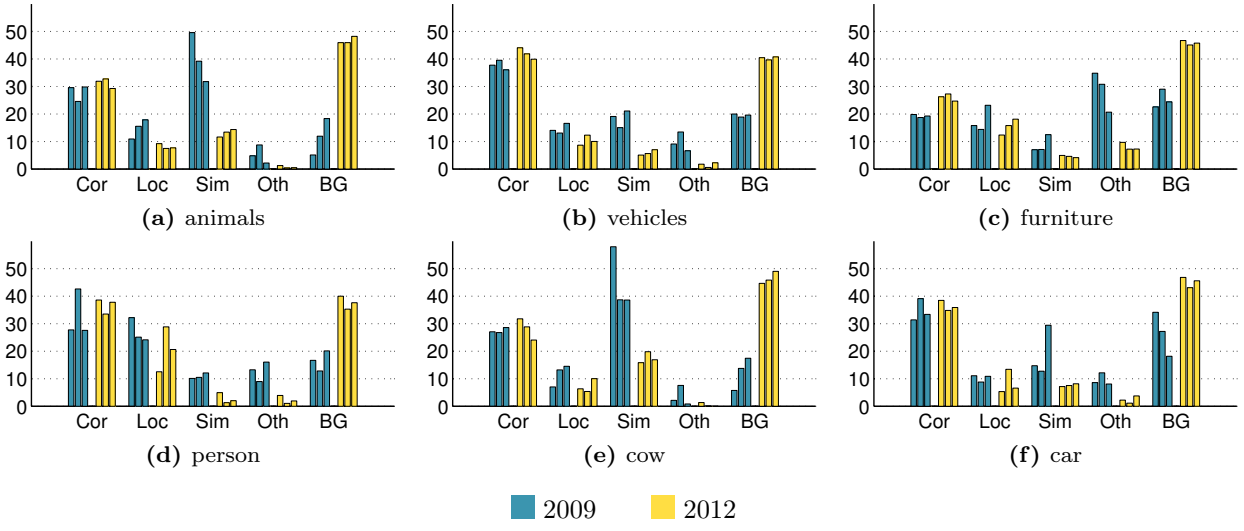
Fig. 26 shows histograms of the mean segmentation accuracy scores achieved on the segmentation task by the methods in the different years. Notice that the highest accuracy achieved has increased at a steady and significant pace from each year to the next.

Fig. 27 shows, for each class, the accuracy of the best-performing method in each year on that class. For most classes the accuracy of the best performing method has increased between 2009 and 2012, although often not monotonically with the year. In particular, in the final years from 2011 to 2012 there was substantial improvement in all classes except bus. This gives hope that there is scope for further improvement in years to come, although note from Table 17 that there was an



(a) 2009 **(b)** 2012

Fig. 24: Per-Category Analysis of Characteristics: Object size. norm. prec. det. ('+') with standard error bars (red). Black dashed lines indicate the overall value of this measure. Key: XS=extra-small; S=small; M=medium; L=large; XL =extra-large. Following Hoiem et al (2012) standard error is used for the average precision statistic as a measure of significance, rather than confidence bounds.



(a) animals **(b)** vehicles **(c)** furniture

(d) person **(e)** cow **(f)** car

Fig. 25: Patterns of errors through time. Percentage of top-ranked detections that are correct (Cor), or are false positives due to poor localisation (Loc), confusion with similar objects (Sim), confusion with other VOC objects (Oth), or confusion with background or unlabelled objects (BG). For each label there are six bars: the left three (blue) are 2009 results (OXFORD_MKL, UOC_LSVM, NEC) while the right three (yellow) are 2012 results (OXFORD, UVA_HYBRID, UVA_MERGED).

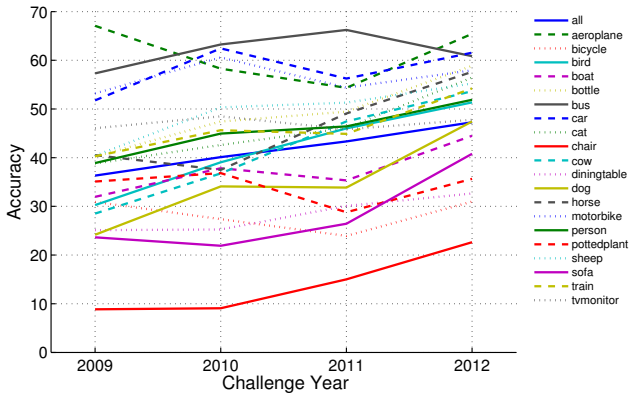


Fig. 27: Segmentation accuracy by class over the years. For each year and class we plot the accuracy obtained by the best-performing method on that class in that year.

increase of over 30% in the amount of data between these years.

We will now discuss the evolution of the methods used in the segmentation challenge since its inception in 2007. In the first year of the segmentation challenge we expected few submissions due to the newness of the challenge and so each participant in the detection challenge was entered automatically in the segmentation challenge to provide a baseline set of methods. This was achieved by filling in the predicted bounding boxes to produce a segmentation. The 2008 segmentation competition had six participants and was the first year where the segmentation methods beat the automatic entries from the detection challenge. The methods used included:

- use of bottom-up segmentations (superpixels),
- MRFs, including high-order potentials,
- refinement of bounding boxes from a detector,
- use of image-level classifiers from the classification challenge.

The 2009 competition had probably the broadest set of entries, twelve in all, whose methods built on and extended those used in the previous year. Key extensions were:

- use of *multiple* bottom-up segmentations to avoid making early incorrect boundary decisions,
- Hierarchical MRFs e.g. modelling object co-occurrence,
- use of parts-based instance models to refine detections,
- deeper integration of segmentation model with detection/classification models,
- use of 3D information.

From 2010 until the final competition in 2012, participants used combinations and refinements of these techniques. The two dominant methods were: hierarchical random fields with a range of potentials, and the use of multiple bottom-up segmentations, combined with a classifier to predict the degree of overlap of a segment with an object. In 2012 the number of participating organisations was down to three, suggesting that the methods used were performing so well it was difficult for a new participant to enter the competition with an original method and get competitive results. That said, the above results through time suggest that segmentation methods are still improving rapidly and we should expect further improvements on these methods in the immediate future.

7 Discussion and the Future

This section appraises certain choices we made in running the challenge and gives suggestions for future challenge organisers.

7.1 Design choices: what we think we got right

7.1.1 Standard method of assessment

Having software that provides a standard method of assessment is essential. For the challenge we provided the following: (i) The train, validation and test splits; (ii) A standard evaluation protocol; and (iii) Evaluation software that computes the evaluation measures. This includes a baseline classifier, detector and segmenter, and the software to run everything 'out of the box', from training through to generating a PR curve and AP on the validation (or test) data.

So, for example, in the classification challenge it is only necessary to replace the call to the baseline classifier we supplied with a call to the new classifier to be trained and evaluated, and then the PR curve and AP measure are generated automatically on the validation data. Providing the splits and evaluation code has meant that results on VOC can be consistently compared in publications.

The lesson here is that *all* challenges should provide code for their standard method of assessment. In cases where this is not provided we still see many examples in papers of authors evaluating the results in different ways – for example some using micro-averaging (giving equal weight to each image) and others using macro-averaging (giving equal weight to each class) – so that results are not comparable.

It is also a good idea to provide written guidelines for annotation, see Pascal VOC annotation guidelines (2012), so that there is little ambiguity and instead

greater consistency in the labelling; and 'best practice' guidelines, see PASCAL VOC best practice guidelines (2012), so that participants know how the organisers intend the data to be used for training, validation and testing.

7.1.2 Measuring performance on the test data

There are (at least) three possibilities on how to obtain the performance on the test data from participants: (i) Release test data and annotation and participants can assess performance themselves; (ii) Release test data, but test annotation is withheld – participants submit results and organisers assess performance (e.g. using an evaluation server); (iii) No release of test data – participants have to submit software and organisers run this and assess performance.

We now discuss these options and our reasons for choosing (ii). The first option (release test data annotations) is the most liberal, but is open to abuse – for example, since participants have the test annotations then the test data could be used for choosing parameters, and only the best result reported (optimising on the test data). There was some evidence that this was happening with the VOC 2007 release, and this was one of the reasons that we adopted option (ii) from VOC 2008 onwards.

In the second option (withhold test annotation) participants simply run their method over the test data and return the results (e.g. a confidence score for each image for each class for the classification challenge) in a standard form. The performance can then be assessed automatically by an evaluation server. Since the data consists of images, there is still some danger that performance of different parameter choices will be assessed by eye on the test data. It is also theoretically possible for participants to hand-annotate the test data, though we rely on participants' honesty and to some extent also the limited time available between the release of the test data and submission of the results to prevent this. Note, this danger may not apply to other challenges where the test data is more abstract than images (e.g. feature vectors).

The third option (participants submit code) is the most safe as there is no release of test data at all. However, there are potentially huge computational and software costs both in being able to run the supplied software on other machines (e.g. the software may use a mix of Matlab/C/GPUs) and in providing the computational infrastructure (clusters, sufficient memory) to run the code over the test dataset in a reasonable time.

Given the potential costs of the third option, we adopted the second. Other than some participants making multiple similar submissions (which contravenes the best practice of choosing the method submitted using the validation set), there were no problems that we were aware of. Here it should be added that in case of multiple submissions, participants were forced to select a single submission without knowing the respective test scores.

7.1.3 Augmentation of dataset each year

Each year new images were added to the dataset and data splits (into training, validation and test) of earlier years maintained. The statistics on how the data was augmented are given in Tables 16 and 17. Both the number of images and number of objects more than doubled between 2008 and 2012.

This had a number of useful consequences: first, it has mitigated overfitting to the data (which might have happened if the data had not changed from year to year); and second, since the earlier year's data is available as subsets, progress can be measured from 2008 to 2012 using the *same* test set. For example, the 2009 dataset is a subset of 2010, 2011 and 2012, and as the assignments to training, validation and test are maintained, performance of all methods can be measured each year using the same 2009 test data (albeit with different training and validation data each year).

Note, the alternative of releasing an entirely new dataset each year would also prevent overfitting, but the disadvantages are that performance would then be measured on a different test set each year. Also there would be the additional cost of preparing entirely new releases of similar sizes, compared to more gradually increasing the sizes.

7.2 Room for improvement

The biggest risk when running a competition like VOC is that it reduces the diversity of methods within the community. A good strategy for a participant is to make an incremental improvement on the previous year's winning method. New methods that have the potential to give substantial improvements may be discarded before they have a chance to mature, because they do not yet beat existing mature methods. Our attempts to solve this problem were:

- to add new competitions (such as the segmentation and action competitions) with different objectives that required new methods to be developed. However, the existing challenges were kept largely fixed so that we could track progress over time, as we saw in Sec. 6.

– to encourage novelty explicitly through invited talks and novelty prizes at the annual workshop. However, it was difficult to assess novelty in an objective fashion and it was easier to favour methods which combined an element of novelty with close to state-of-the-art performance.

However, we feel that there is still room for improvement in maintaining diversity. To this end, we will now suggest some other strategies that may help increase diversity in future competitions.

7.2.1 Multiple evaluation metrics

The design of the evaluation metric for a competition is often controversial since small changes in the metric can give substantial advantages to one method or another. Indeed the metrics used in each VOC competition have typically been changed or refined at least once during the lifetime of the competition. These changes may have reduced but not eliminated another type of reduction in diversity: that methods may end up overfitting to the particular evaluation metrics selected for each competition.

One possible suggestion that may mitigate this is to use and publish several different metrics when evaluating participants in each competition. Each metric should pick up on a different aspect of the success of a method. Not only would this reduce overfitting but it would provide more detail into the strengths and weaknesses of each method, help inform collaboration between teams with different strengths, and encourage the discussion and development of informative metrics. However, it would still be necessary to combine the metrics somehow to determine the competition winner.

Another suggestion would be to report a diversity metric. This could be a metric across all participants, which would allow the diversity to be tracked over time making any reduction in diversity more visible and encouraging more discussion and improvements relating to diversity. Alternatively (or in addition), there could be an evaluation metric that indicates how different each participant is from the average participant – effectively giving additional weight to participants that do well on images where other participants do badly.

7.2.2 Community boosting

This idea can be taken further by formalising it into a community-level boosting algorithm. The idea would be to attach weights to each test image and upweight images that were poorly handled by the community of methods in the previous year's challenge. These weights would then be used when calculating evaluation metrics to favour methods that differ from the previous year, and thus encourage greater diversity.

One way this could be achieved in practice would be to combine together the previous year's methods, for example, by using a super-classifier like the one used in Sec. 5. This super-classifier would then be used as the 'weak learner' in a standard boosting algorithm to compute the image weights for the current year. The result would be a boosting algorithm running at one iteration per year. This would have the interesting side benefit of producing a classifier that combined all methods of all years into a single 'strong' classifier.

Boosting is known to suffer from overfitting and this could be a problem here, although the small number of iterations should limit the scope for overfitting. Another issue is this could lead the community to focus on specialised solutions to niche problems. Nonetheless, we believe this approach would be worth considering for future challenges.

7.2.3 Community analysis of results

Our current analysis has been centred around rather global evaluation metrics, i.e. a set of numbers that cannot summarise all aspects. However, as the work by Hoiem et al (2012) has shown, there are many interesting and relevant aspects that could well stay under the radar but would warrant further attention. This is especially the case for failure cases that may be rare – and would therefore hardly impact global performance measures – but nonetheless need solving if object recognition is to come of age. It would therefore be interesting to make all submissions public, e.g. by extending the evaluation server so that everyone can query them. However, this is not our current intention, as it would require additional resources, and would also create complications with the policy of withholding the testing annotations from future participants.

7.3 Conclusions and impact

The Pascal VOC challenges and workshops have certainly contributed to the surge in interest in category recognition in the computer vision community over the last decade, and have been mentioned in thousands of papers. They have been used for research areas that we did not have in mind at all when they were created, such as studying dataset bias (Torralba and Efros, 2011) – where algorithm performance is compared across different training and testing datasets (e.g. VOC vs. ImageNet), or automatically inferring attributes for each object (e.g. has wheel, has head) with additional annotation provided to support this (Farhadi et al, 2009).

Our current intention is to keep the VOC 2008–2012 test data annotations secret (so as to minimise the effects of overfitting), and to keep the evaluation server available (Pascal VOC evaluation server, 2012). We have added a leaderboard so that future submissions may be easily recorded and compared. We intend for this leaderboard to include the bootstrapping technique described in Sec. 3.5, so that the significance of the difference between entries on the leaderboard can be assessed.

We note here some of the particular successes of the period. First, has been the development of a new class of object category detector, the DPM of Felzenszwalb et al (2010), with open source code provided to the community. Second, has been the steady increase in performance over all the three main challenges. Third, has been the development of cross fertilisations between the challenges – detectors used as part of the segmentation and classification methods, and (unsupervised) segmentation used to propose windows in detection methods – where originally methods were myopic in only applying to one of the three challenges. Fourth, VOC has contributed to establishing the importance of benchmarks, and in turn has led to efforts to refine and analyse the results in more detail, e.g Hoiem et al (2012). Finally, VOC has led to a new generation of challenges with far greater number of classes (e.g. ImageNet), with explorations of more structure (e.g. parts and attributes), and with finer grain visual categorisations (e.g. distinguishing between sub-ordinate classes of flowers, birds, dogs).

Winston Churchill famously said *"democracy is the worst form of government except all those other forms that have been tried from time to time"*. It could equally be said that organising a challenge is the worst way to track and encourage progress in a research community, except for all the other ways. Certainly a widely adopted challenge can be a curse as well as a blessing. In running the Pascal VOC challenge, we have tried to steer a course that maximises the community benefits and minimises the costs, and believe that we have had some reasonable success in doing so. As we move into a new generation of machine vision problems, challenges will continue to play a critical role in assessing, recognising and communicating progress. We wish the organisers of these challenges the very best in steering their own paths through the uncharted territory ahead.

Acknowledgements First, we thank all the groups that participated in the challenge – without these VOC would just have been a dataset.

Second, we would like to thank those who have been 'friends of the challenge' – making helpful suggestions and criticisms throughout: Alyosha Efros, David Forsyth, Derek Hoiem, Ivan Laptev, Jitendra Malik and Bill Triggs.

Third, we thank those who have given additional assistance in developing and maintaining the PASCAL challenge: Marcin Eichner, Sam Johnson, Lubor Ladicky, Marcin Marszalek, Arpit Mittal and Andrea Vedaldi. In particular, we thank Alexander Sorokin for the first version of the evaluation server, and Yusuf Aytar for subsequent versions.

Fourth, we gratefully acknowledge the annotators from VOC2008 onwards: Yusuf Aytar, Lucia Ballerini, Jan Hendrik Becker, Hakan Bilen, Patrick Buehler, Kian Ming Adam Chai, Ken Chatfield, Mircea Cimpoi, Miha Drenik, Chris Engels, Basura Fernando, Adrien Gaidon, Christoph Godau, Bertan Gunyel, Hedi Harzallah, Nicolas Heess, Phoenix/Xuan Huang, Sam Johnson, Zdenek Kalal, Jyri Kivinen, Lubor Ladicky, Marcin Marszalek, Markus Mathias, Alastair Moore, Maria-Elena Nilsback, Patrick Ott, Kristof Overdulve, Konstantinos Rematas, Florian Schroff, Gilad Sharir, Glenn Sheasby, Alexander Sorokin, Paul Sturgess, David Tingdahl, Diana Turcsany, Hirofumi Uemura, Jan Van Gemert, Johan Van Rompay, Mathias Vercruysse, Vibhav Vineet, Martin Vogt, Josiah Wang, Ziming Zhang, Shuai Kyle Zheng.

Fifth, we are grateful to the IST Programme of the EC under the PASCAL2 Network of Excellence, IST-2007-216886 who provided the funding for running the VOC challenge, and Michele Sebag and John-Shawe Taylor who coordinated the challenge programme and PASCAL2 respectively.

Finally, we would like to thank the anonymous reviewers for their encouragement and feedback – their suggestions led to significant improvements to the paper.

References

Alexe B, Deselaers T, Ferrari V (2010) What is an object? In: Proceedings of Conference on Computer Vision and Pattern Recognition, pp 73–80

Alexiou I, Bharath A (2012) Efficient Kernels Couple Visual Words Through Categorical Opponency. In: Proceedings of British Machine Vision Conference

Bertail P, Clémençon SJ, Vayatis N (2009) On Bootstrapping the ROC Curve. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) Advances in Neural Information Processing Systems 21, pp 137–144

Carreira J, Caseiro R, Batista J, Sminchisescu C (2012) Semantic segmentation with second-order pooling. In: Proceedings of European Conference on Computer Vision

Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. Transactions on Intelligent Systems and Technology 2:27:1–27:27, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

Chen Q, Song Z, Hua Y, Huang Z, Yan S (2012) Generalized Hierarchical Matching for Image Classification. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Csurka G, Dance C, Fan L, Williamowski J, Bray C (2004) Visual categorization with bags of keypoints.

In: Proceedings of ECCV2004 Workshop on Statistical Learning in Computer Vision, pp 59–74

Dalal N, Triggs B (2005) Histograms of Oriented Gradients for Human Detection. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2013) Decaf: A deep convolutional activation feature for generic visual recognition. CoRR abs/1310.1531

Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2010) The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision 88:303–338

Farhadi A, Endres I, Hoiem D, Forsyth D (2009) Describing objects by their attributes. In: Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, pp 1778–1785

Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object Detection with Discriminatively Trained Part Based Models. Transactions on Pattern Analysis and Machine Intelligence 32(9):1627–1645

Flickr website (2013) http://www.flickr.com/

Girshick RB, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Hall P, Hyndman R, Fan Y (2004) Nonparametric confidence intervals for receiver operating characteristic curves. Biometrika 91:743–50

Hoai M, Ladicky L, Zisserman A (2012) Action Recognition from Still Images by Aligning Body Parts. URL http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/workshop/segmentation_action_layout.pdf, slides contained in the presentation by Luc van Gool on Overview and results of the segmentation challenge and action taster

Hoiem D, Chodpathumwan Y, Dai Q (2012) Diagnosing error in object detectors. In: Proceedings of European Conference on Computer Vision

Ion A, Carreira J, Sminchisescu C (2011a) Image segmentation by figure-ground composition into maximal cliques. In: Proceedings of International Conference on Computer Vision

Ion A, Carreira J, Sminchisescu C (2011b) Probabilistic Joint Image Segmentation and Labeling. In: Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K (eds) Advances in Neural Information Processing Systems 24, pp 1827–1835

Karaoglu S, Van Gemert J, Gevers T (2012) Object Reading: Text Recognition for Object Recognition. In: Proceedings of ECCV 2012 workshops and demonstrations

Khan F, Anwer R, Van de Weijer J, Bagdanov A, Vanrell M, Lopez AM (2012a) Color Attributes for Object Detection. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Khan F, Van de Weijer J, Vanrell M (2012b) Modulating Shape Features by Color Attention for Object Recognition. International Journal of Computer Vision 98(1):49–64

Khosla A, Yao B, Fei-Fei L (2011) Combining Randomization and Discrimination for Fine-Grained Image Categorization. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: Bartlett P, Pereira F, Burges C, Bottou L, Weinberger K (eds) Advances in Neural Information Processing Systems 25, pp 1106–1114

Lazebnik S, Schmid C, Ponce J (2006) Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: Proceedings of Conference on Computer Vision and Pattern Recognition, pp 2169–2178

Leibe B, Leonardis A, Schiele B (2004) Combined Object Categorization and Segmentation With An Implicit Shape Model. In: Proceedings of ECCV Workshop on Statistical Learning in Computer Vision

Lempitsky V, Zisserman A (2010) Learning to count objects in images. In: Advances in Neural Information Processing Systems 23

Li F, Carreira J, Lebanon G, Sminchisescu C (2013) Composite Statistical Inference for Semantic Segmentation. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Lowe DG (2004) Distinctive Image Features from Scale-invariant Keypoints. International Journal of Computer Vision 60(2):91–110

Nanni L, Lumini A (2013) Heterogeneous bag-of-features for object/scene recognition. Applied Soft Computing 13(4):2171–2178

Van Gemert J (2011) Exploiting photographic style for category-level image classification by generalizing the spatial pyramid. In: Proceedings of International Conference on Multimedia Retrieval

Van de Sande K, Uijlings J, Gevers T, Smeulders A (2011) Segmentation As Selective Search for Object Recognition. In: Proceedings of International Conference on Computer Vision

O'Connor B (2010) A Response to "Comparing Precision-Recall Curves the Bayesian Way?". A comment on the blog post by Bob Carpenter on *Comparing Precision-Recall Curves the Bayesian Way?* at http://lingpipe-blog.com/2010/01/29/comparing-precision-recall-curves-bayesian-way/

Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Russakovsky O, Lin Y, Yu K, Fei-Fei L (2012) Object-centric spatial pooling for image classification. In: Proceedings of European Conference on Computer Vision

Russell B, Torralba A, Murphy K, Freeman WT (2008) LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision 77(1–3):157–173, http://labelme.csail.mit.edu/

Salton G, Mcgill MJ (1986) Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA

Sener F, Bas C, Ikizler-Cinbis N (2012) On Recognizing Actions in Still Images via Multiple Features. In: Proceedings of ECCV Workshop on Action Recognition and Pose Estimation in Still Images

Song Z, Chen Q, Huang Z, Hua Y, Yan S (2011) Contextualizing Object Detection and Classification. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Pascal VOC 2012 challenge results (2012) http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/results/index.html

Pascal VOC annotation guidelines (2012) http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/guidelines.html

Pascal VOC best practice guidelines (2012) http://pascallin.ecs.soton.ac.uk/challenges/VOC/#bestpractice

Pascal VOC evaluation server (2012) http://host.robots.ox.ac.uk:8080/

Torralba A, Efros AA (2011) Unbiased look at dataset bias. In: Proceedings of Conference on Computer Vision and Pattern Recognition, IEEE, pp 1521–1528

Uijlings J, Van de Sande K, Gevers T, Smeulders A (2013) Selective Search for Object Recognition. International Journal of Computer Vision 104(2):154–171

Vedaldi A, Gulshan V, Varma M, Zisserman A (2009) Multiple Kernels for Object Detection. In: International Conference on Computer Vision

Viola P, Jones M (2004) Robust real-time object detection. International Journal of Computer Vision 57(2):137–154

Wang X, Lin L, Huang L, Yan S (2013) Incorporating Structural Alternatives and Sharing into Hierarchy for Multiclass Object Recognition and Detection. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Wasserman L (2004) All of Statistics. Springer

Xia W, Song Z, Feng J, Cheong LF, Yan S (2012) Segmentation over Detection by Coupled Global and Local Sparse Representations. In: Proceedings of European Conference on Computer Vision

Yang J, Yu K, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Zeiler MD, Fergus R (2013) Visualizing and understanding convolutional networks. CoRR abs/1311.2901

Zhu L, Chen Y, Yuille A, Freeman W (2010) Latent Hierarchical Structural Learning for Object Detection. In: Proceedings of Conference on Computer Vision and Pattern Recognition

Zisserman A, Winn J, Fitzgibbon A, Van Gool L, Sivic J, Williams C, Hogg D (2012) In Memoriam: Mark Everingham. Transactions on Pattern Analysis and Machine Intelligence 34(11):2081–2082