An Exemplar Model for Learning Object Classes

Ondřej Chum Andrew Zisserman
Visual Geometry Group, Department of Engineering Science, University of Oxford
{ondra,az}@robots.ox.ac.uk

Abstract

We introduce an exemplar model that can learn and generate a region of interest around class instances in a training set, given only a set of images containing the visual class. The model is scale and translation invariant.

In the training phase, image regions that optimize an objective function are automatically located in the training images, without requiring any user annotation such as bounding boxes. The objective function measures visual similarity between training image pairs, using the spatial distribution of both appearance patches and edges. The optimization is initialized using discriminative features.

The model enables the detection (localization) of multiple instances of the object class in test images, and can be used as a precursor to training other visual models that require bounding box annotation.

The detection performance of the model is assessed on the PASCAL Visual Object Classes Challenge 2006 test set. For a number of object classes the performance far exceeds the current state of the art of fully supervised methods.

1. Introduction

The objective of this work is object class detection, *i.e.* identifying class instances and their spatial extent. Since 2003 there has been a tremendous improvement in object *classification* performance, *i.e.* classifying an image as positive if it contains one or more instances of the object class, as is demonstrated by the striking increases in scores for the Caltech 101 test set (*e.g.* see [24]) and the high performance on the PASCAL Visual Object Classes (VOC) Challenge classification task [7]. However, *detection* has not reached such levels of performance – a consequence of the greater difficulty of the task – though there have been notable improvements for several classes (e.g. pedestrians [5, 19], bicycles, cars, motorbikes, *etc*. [17, 18, 20]).

We have two goals here: the first is to learn a region of interest (ROI) for class instances in weakly supervised training data, *i.e.* given only a set images known to contain instances of an object class, determine the scale and position of the instance in each image. The second goal is to learn a class model from these ROIs that can be used to detect instances of the object class in (unseen) test images.

The first problem requires a method of measuring visual similarity across the set of training images in order to "tease out" the class instance in each image. Several previous methods have cast this as an optimization problem, fitting a generative model – for example LOCUS [23] and the Constellation model [8] optimize model likelihood. We also formulate the problem as one of optimization, but with a more general model than those of [8, 23]. LOCUS is limited by its use of the EM algorithm, since this depends on a good initialization. It can succeed provided the class instance is sufficiently large compared to image clutter, does not vary significantly in scale over the image set (since only a limited range of scales are tried), and is unoccluded. We represent the class by a set of exemplars, with each exemplar recording the spatial layout of appearance patches and edges. The visual similarity between images is measured using the hierarchical spatial histogram method of [10, 13], but here generalized to apply to a ROI, rather than an entire image, and also to include edges as well as appearance patches. We demonstrate much larger scale variability than is achieved in LOCUS, using discriminative visual feature [3, 6] to initialize the optimization, and also greater robustness to missing instances in the training data than exhibited by previous methods [1, 2].

The second problem, that of learning a detector given the ROI, is explored in two ways. First, we use the exemplar model as a detector. Previously exemplar models have been used for classification [24]. Second, we demonstrate that the ROIs determined in the training set may be used to train other models, and illustrate this with an SVM based region classifier. Though any previous model that requires manual bounding box annotation, such as [14, 18, 20], could now be trained automatically in this manner.

The exemplar model is described in section 2, and the learning algorithm in section 3. We then describe the method of detection on new images in section 4. The detection models are tested in section 5 on both the standard 'Caltech 4' datasets used by [8], and also on the far more challenging PASCAL VOC 2006 set. In section 6, we discuss

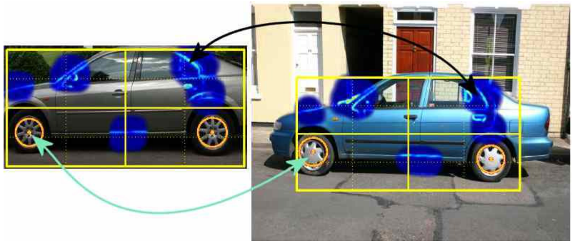

Figure 1. An exemplar image and a corresponding class instance in a car side training set. The hierarchical representation and cost function measure the spatial correspondence between sparse visual words and dense edge distributions. Some corresponding visual words and edges are highlighted.

straightforward extension of our model, and show some preliminary results.

2. The Exemplar Model

The model for each class consists of a set of exemplars obtained from ROIs around the object class instances in the training images. Each exemplar represents the spatial layout of visual words and edge directions in the region using a hierarchical spatial histogram. The spatial correspondence between an exemplar and a target image region can then be assessed by a level-weighted distance [13] between the histograms representing the exemplar and target. Figure 1 illustrates this correspondence. Implementation details are given in section 2.2, but first we describe how the model is learnt.

2.1. Learning the exemplar model

Suppose we know the model, and wish to detect a class instance in a target image. This can be done by a search for a ROI in the target image that matches well with one of the exemplars, *i.e*. as a minimization of the distance between the exemplars and target region as the target region is varied. We define the following cost function to measure this similarity:

$$C_D = \sum_X (d(X^w, Y^w)) + \alpha\,(d(X^e, Y^e)) + \beta \frac{(A-\mu)^2}{\sigma^2} \quad (1)$$

where X^w and X^e are the hierarchical spatial histograms of visual words and edge directions, respectively, in the exemplars, and Y is similarly defined for the target image ROI. The sum is over the set of exemplars X of the model. A is the aspect ratio of the target region, μ and σ are the aspect ratio average and variance, respectively, of the exemplar ROIs. The cost C_D is a weighted sum of three terms: the pair wise distance between the visual words of the target region and exemplar, the pair wise distance between the edge directions of the target region and exemplar, and a cost for the aspect ratio of the target region deviating from the average aspect ratio. The distance function used is defined below.

The detection problem involves finding the target region that minimizes C_D. We now turn to learning the exemplar set model from training images. Suppose we are given a set $\mathcal{T}$ of N training images, we wish to find the region in each training image which best matches with regions in the other training images. These regions will define the exemplar set. This is equivalent to the detection problem above, where now we must learn the regions in all images simultaneously. The cost function is then a sum of distances between all pairs of training examples

$$C_L = \sum_{X\in\mathcal{T}} \sum_{Y\in\mathcal{T}} (d(X^w, Y^w)) + \alpha\,(d(X^e, Y^e)) + \beta \frac{(A-\mu)^2}{\sigma^2} \quad (2)$$

and we wish to find the region in each training image such that C_L is minimized.

Thus, learning the model involves: (i) automatical location of the exemplar regions from the training set; and (ii) selecting the value of the parameters α, β and learning the parameters μ and σ. Examples of learnt models are given in figure 2. The learning algorithm is given in section 3.

Distance functions. It is well known that distances may be strongly corrupted by the presence of an outlier, *i.e*. in this case an example image not containing an instance of a category object, or a missed detection. Instead of histogram intersection we use a (squared) χ^2 distance since then a single training image has a limited influence on the model. This follows from the fact that the cost function (2) is additive and the contribution of each exemplar is bounded by a constant. Note, that other costs with this *robustness* property could be used, for example the Jensen-Shannon divergence. So,

$$d(x,y) = (\chi^2(x,y))^2$$

where $\chi^2(x,y) = \sum \frac{(x_i-y_i)^2}{x_i+y_i}$. In our experiments, the sum of squared χ^2 distances outperformed the sum of χ^2 distances as well as the Jensen-Shannon divergence.

Discussion. The model records the feature's position relative to the ROI. This may be compared to recording the relative position of "object parts" with respect to the model centroid and scale in the manner of the Implicit Shape Model [14] or star model [9]. These latter models represent the variation in feature position over different class instances. In the case of the constellation model [8], especially, this has a high cost in learning. In our approach, learning the feature position variation is avoided by the use of multiple exemplars, and correspondence is handled by the pyramidal spatial histograms. We are bringing together three previous ideas: (i) the generalized Hough Implicit Shape model of Leibe *et al.* [14] using sparse appearance patches, (ii) the edge/boundary representation of [17, 18, 20], and (iii) correspondence matching using a hierarchical spatial histogram [10, 13]. Our representation of edge information is dense, and use of the spatial pyramid means that we are able to capture a bag of orientations at the lowest level [22], with stronger spatial correspondences represented by the higher levels. Modelling the aspect ratio is not essential for the method, but improves the precision of the object's bounding box for the PASCAL VOC evaluation.

2.2. Implementation details

Appearance patches. The image features are detected using the Hessian-Laplace [17] operator, and described by a rotation variant SIFT descriptor [15]. The SIFT descriptors are then vector-quantized using k-means into visual words [21]. This procedure is performed over all training images, which includes images containing class instances as well as a database of (mostly) non-class images.

Edge directions. To construct a histogram of edge directions we use the Canny edge detector to compute the edges. Eight different directions are extracted, edges with the same direction and opposite gradient are unified (*i.e.* contrast sign is ignored). For an efficient implementation, an integral image of edge density is computed for each direction. The integral images allow quick computation of edge direction histograms over rectangular regions.

Hierarchical spatial pyramid histogram. The edge distribution histogram uses the representation of [13] as a pyramid with three levels, uniformly weighted. The dimensionality of the edge histogram is $(1 + 4 + 16) \times 8 = 168$.

The (spatial) histogram of visual words also uses the representation of [13] but extended to also include the scale of the Harris-Laplace feature. For every spatial bin at each level there are two scale bins. So, for example, if the pyramid has two spatial levels ($L = 0, 1$) then there are 5 spatial bins in the original model and 10 scale–spatial bins in our model. Visual words are assigned to the scale bins using soft assignment, and are weighted by their discriminability D for the given class. Here discriminability, *i.e.* how much a particular visual word w is relevant to the class, is computed by a likelihood ratio discriminability function [6],

$$D(w) \sim \frac{\#\text{class labelled images containing } w}{\#\text{images in database containing } w}. \quad (3)$$

where the database will mostly contain non-class images. We will give examples of these datasets in the experiments of section 5. Only the top 256 most discriminative visual words are used. The appearance patches are represented by a pyramid histogram X^w with four levels. The top level $L = 0$ representing a bag of words is ignored (weighted 0) since the visual words used were preselected. The other three levels are weighted 1, 1, and 2 respectively. Note that inclusion of the feature scale in the representation means that a correspondence requires similar spatial position *and* similar spatial scale.

	Initialization
1	Calculate the discriminability of visual words using all features in the training images using (3).
2	Initialize the ROI in each training image by a bounding box of the 64 most discriminative features.
	Iterative minimization
3	Find the image and a displacement of the ROI in that image so that the cost function (2), with $\beta = 0$, is lowered the most.
4	Reinitialization by detection. Construct a model from training images where the cost function converges, and search for the class instance in the remaining training images.
	Refinement
5	Enlarge the ROI in the training images by 10%
6	Calculate the discriminability of visual words using only the features inside the ROI by (3).
7	Execute iterative minimization using new set of discriminative visual words.

Table 1. Overview of the learning algorithm.

Cost function parameters. The parameter α is manually chosen so that the χ^2 distance of sparse feature histograms and dense edge histograms are of approximately the same magnitude. We choose $\beta = 0.1$. Both parameters α and β are fixed across all categories. Note, their values can be learnt by cross-validation if more extensive annotation data is available.

3. Learning algorithm

Here we describe how the model representation is learnt automatically from a set of training images. We will use the object classes cars and bicycles (with training images from the PASCAL VOC 2006 set) as our running example.

We are given a set of training images, and no other information (*i.e.* no positional or segmentation for the class instance). Learning proceeds in a number of stages, as outlined in table 1. The first stage provides an initial estimate of a class instance region in each training image. To achieve this, *discriminative* visual words are learnt for the object class (from the training and a negative set), and their distribution in each image determines the initial region estimate. In the second stage, a cost function using the model representation to measure visual similarity between the regions is optimized over the region's position (aspect ratio is not considered in this stage, *i.e.* $\beta = 0$). In a final refinement stage, discriminative words are re-learnt, based on words within the current region estimates, as well as the aspect ratio parameters μ and σ. The cost function incorporates these words and parameters for the final optimization.

Initialization. The ROIs are initialized as a bounding box of the 64 most discriminative features [3]. The number is not crucial, we have observed the same performance using

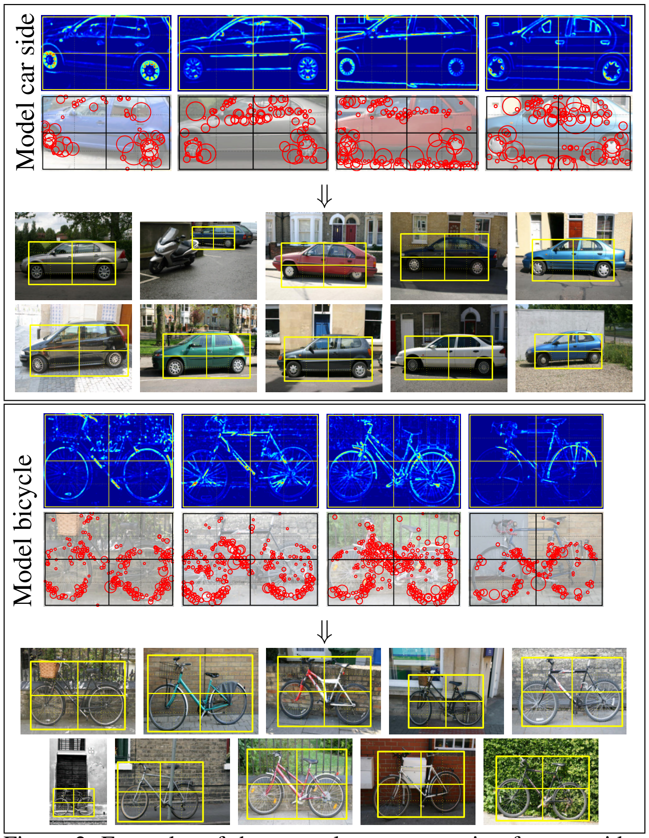

Figure 2. Examples of the exemplar representation for cars side (top) and bicycles (bottom). Models show the spatial distribution of edges and appearance patches. In each case the images below the model show samples from the training images with the automatically learnt ROIs overlaid.

32 – 128 of the most discriminative features. Discriminability is measured using the likelihood ratio discriminability function D of (3), and provides a ranking of the visual words. The top ten most discriminative visual words for various classes are shown in figure 3.

Optimization. In each image, a number of new positions for the ROI are hypothesized. The hypotheses are generated from the current ROI position by translation, and isotropic and anisotropic scaling. At each iteration one image is selected so that the new position of the ROI minimizes the cost function. The ROI in this image is then updated to the new position. Note, that the cost function can be calculated efficiently (by a sliding window update using the integral images), since only one image is updated at a time. When the cost function is trapped in a local minima, a ROI that increases the cost function the least is taken. In such a case, an image containing that region cannot be updated for another $N/2$ steps (where N is a number of training images) to avoid returning to the same local minima. The progress of the cost error against the number of iterations is plotted in figure 4.

Sometimes, the optimization procedure does not converge for individual images. Such a situation typically occurs when the initialization is bad (*e.g*. due to the presence of multiple instances of the object in the image, or simply by an unlucky co-occurrence of features not relevant to the object – see figure 5). However, those images can easily be identified as their distance to other images is significantly larger than the distances between most image pairs. In these cases the ROI is reinitialized by detection in that image: a new ROI is sought for in the image, not only in vicinity of the current ROI, but over the whole image. This search is done efficiently by detection of the model defined by ROIs in the other images (*i.e*. minimizing the cost function C_D of (2)). If this search fails, then the image is rejected as an outlier from the training set. The regions remaining at the end of the algorithm (*i.e*. those not rejected) are the exemplar model learnt for this class.

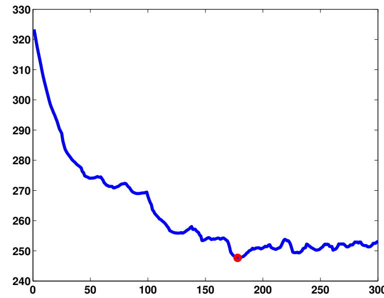

Figure 4. The cost function against the number of iterations while learning the car side category. The circular marker denotes the minimum.

Refinement. At the start of the optimization the discriminability of visual words is estimated from whole training images, since there is no information about the location of the objects within the images at that stage. For this reason, some background features not directly related to the object are included as well. In the refinement stage, the discriminability of visual words using (3) is re-estimated only from those words within the ROIs (enlarged by 10%) found in the previous step. Another optimization step is executed with the new values of D, and, consequently, a potentially different set of discriminative words. Aspect ratio μ and σ learnt in the previous step are also used.

Computational cost. One iteration of the minimization process involves computing pyramid histograms for the newly proposed locations of the ROI, and computing distances of those new descriptors to the other training images used in optimization. This has complexity $\mathcal{O}(N^2)$, where N is the number of training images. On a 2GHz machine, our MATLAB implementation of the learning process takes 3-7 minutes to complete for 15-20 training images (less than 0.5 sec per iteration).

Discussion. The underlying assumption of the learning method is that the object class whose model we are trying to learn has similar appearance (visual and spatial) in

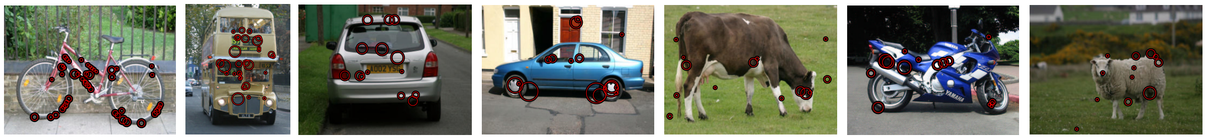

Figure 3. The top 10 most discriminative visual words for various classes of the PASCAL VOC 2006 image set.

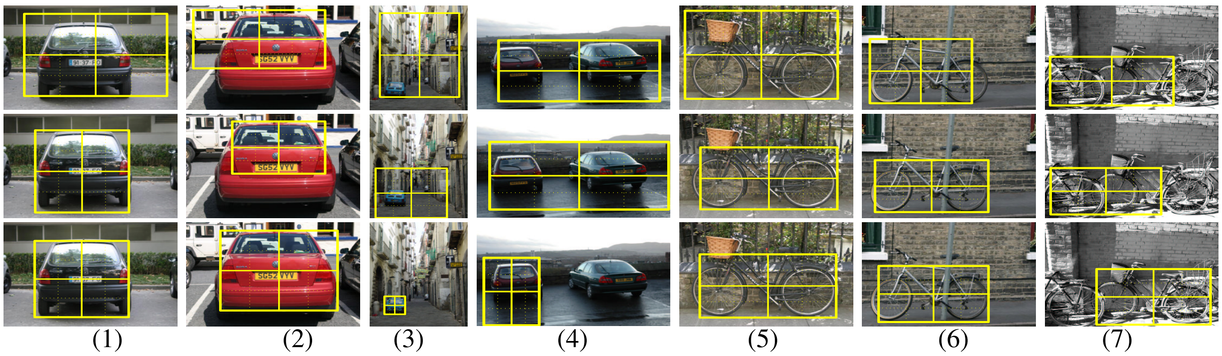

Figure 5. Three stages of the optimization process from top to bottom: initialization, after 100 iterations, after reinitialization by detection. The first four columns show 4 example training images (from 17 used) of car rear, the last three columns show 3 example training images (from 19 used) of the bike left category. The first column shows an example, where the ROI quickly converges to the class instance, whereas in columns 3–4 reinitialization by detection is needed for the convergence. The discriminative features provide very good initialization for the bicycle class, unless there are multiple instances of the class in the image.

all images. The algorithm finds similar regions in the set of training images and their description is then used as a class model. As usual in such learning algorithms, if there is insufficient variability in the background of the positive training images, then it can be incorporated as part of the class model. A common example is shadows under cars, where part of the road is included in the car side model. We find that the edge features are quite helpful in identifying background dissimilarity and limiting the growth of the class instance regions.

4. Detection

Having learnt the exemplar model we now describe how it may be used to detect a class instance in a new (test) image. We consider two cases. In the first the exemplar model is used both to determine the ROI of class instances and to make a decision on whether there is an instance there or not. In the second case the ROIs generated by the exemplar model are used to train a different recognition method – in the example here a SVM.

4.1. Using the exemplar model

The detection is formulated as a cost function minimization, essentially identical to the function minimized in the learning phase. To efficiently find all local minima of the cost function, *i.e*. possible locations of (multiple) instances of the object, a hypothesize and locally optimize approach is adopted. Individual visual words (features) are used to generate a hypotheses for the class instance location. Then the location is refined by minimizing the cost function (1) over a ROI search, initialized from the hypothesis.

In detail, a hypothesis is a pair (w, R) of visual word w and a rectangle R. The rectangle represents the ROI with fixed relative position and scale with respect to the position and scale of the visual word w. The pairs (w, R) are learnt from the ROI of the exemplar images during the training stage. Consider a particular visual word w. In the training images there will be a number of rectangles R_i associated with w – in a similar manner to a number of centroids being associated with a part in the Implicit Shape Model of [14]. Rather than learning a distribution over R_i, we aggregate these into a single rectangle using mean-shift clustering. This idea is similar to that of [16], and is illustrated in figure 6. The uncertainty of the object location is then handled by the iterative cost function minimization. This approach exploits the rough localization provided by the sparse appearance patches as well as the dense information provided by the edge orientation histograms, which cannot be directly used in the generalized Hough transform. The hypothesis can be seen as a rough localization of an "average" class instance given an object part (a visual feature). The local optimization can be seen as adapting the location given the intra-class variation of the specific instance.

It is clear that not all hypotheses are created equally. For example, hypotheses originating from visual words that are either common in non-class images, or often appear in class images but at different locations, *etc*., are unlikely to provide a good estimate of object location. We measure the quality of a hypothesis (w, R) by a score proportional to the likelihood ratio D given in (3), and the number $n_{(w,R)}$

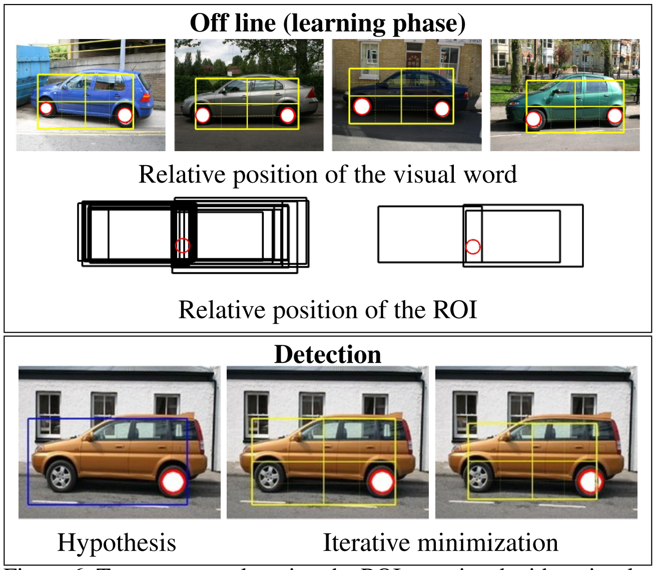

Figure 6. Top two rows: learning the ROI associated with a visual word related to a visual word representing a wheel of the car – in this case the relative positions of the ROI (car) with respect to the visual word (wheel) gathered over the exemplar images (left) and quantized (right). Bottom row: in the detection phase, a detected wheel gives a rough hypothesis of car location (left) which is iteratively refined (right).

of exemplar images consistent with the hypothesis, and inversely proportional to the number $\#w$ of appearances of the visual word w in the exemplar images. This defines the strength S of a hypothesis as

$$S(w, R) = D(w)\frac{n_{(w,R)}}{\#w}. \tag{4}$$

The 20 strongest hypotheses are tested on each image during the detection. The cost function of hypothesized detections is thresholded and non-maxima suppression is applied before the hypothesis is accepted.

On a typical 2GHz machine the detection (MATLAB implementation) takes about 25 seconds per test image for a 20-exemplar model.

Performance results for detection using this model are given in section 5. We next illustrate the fact that the exemplar model learnt as in section 3 can be used to train a different type of model. We consider the problem of class confusion and learn a model targetted at this.

4.2. Using other models

We use the exemplar model to provide ROIs, which then can be used for training any model. Detections in images labeled as class positive provide positive examples, detections in images labeled as class negative provide negative examples.

To illustrate the idea, here we train an SVM (using SVM light [11]). The features used for the SVM are spatial histograms of visual words, similar to those used in the detection model – the difference being that no preprocessing regarding disciminativity of the visual words is done – all visual words are used, all have equal weight. In the testing phase, all detections are re-ranked by the SVM score.

We show in section 5 that this model reduces the class confusion that occurs when two different classes share appearance patches as well as their spatial distribution, such as for bicycles and motorbikes.

Note the SVM model could not be used for detection directly (*i.e.* without the exemplar model first providing ROIs), though it could be used to classify the images.

5. Experiments

In this section, we assess the performance of the model on standard datasets: the PASCAL VOC 2006 detection challenge, and the Caltech 4.

5.1. PASCAL VOC 2006

The PASCAL VOC 2006 Detection Challenge [7] requires participants to predict the bounding boxes and confidence values for any instances of 10 possible classes detected in a test image. The classes and number of training and test images are listed in [7]. The confidence values are used to generate a precision/recall curve, and success is measured by the average precision of this curve.

For our experiments, we use six categories, and for these the following aspects are learnt: car (left, right, rear, frontal), bicycle (left, right), bus (left, right, frontal), motorbike (left, right), cow (left, right), and sheep (side). We randomly select 15-20 training images per aspect (where possible) to train the model. The test results combine the detections from all aspects of each category. We use a vocabulary of 3000 visual words clustered over training images of the PASCAL VOC 2006 [7] set.

For the PASCAL evaluation it is necessary to provide a bounding box for each detection. During training we also learn the relation between our final ROI and the provided bounding box in the training data. Note, the bounding box information is *not* used to train the exemplar model at any stage.

Exemplar model results. The results are presented in form of the precision recall plots. Plots in figure 7 and the left column of figure 8 show the detection performance of the exemplar model alone. For some categories (car, motorbike, cow, sheep), the result of our method are comparable to others, for other categories (bus, bicycle), our method significantly outperforms the current state of the art. The method fails to learn a model for the 'person' category, probably due to the articulations and significant class variability in the PASCAL training set.

The method does not use the ground truth bounding box in the training step and only a small number of training images are needed. We next investigate both of these.

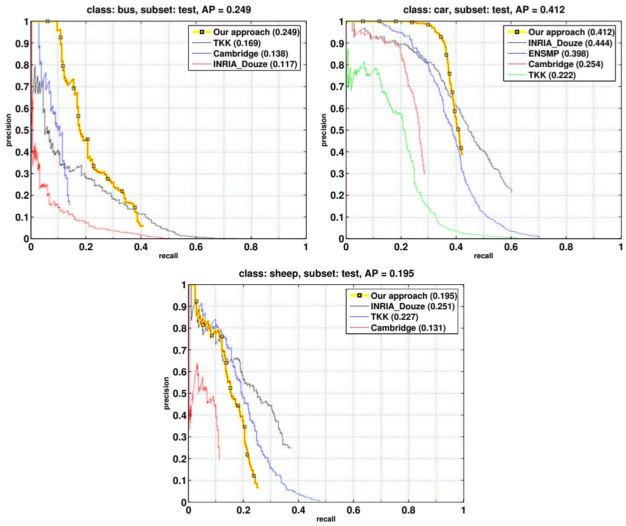

Figure 7. Precision–recall curves for classes from the PASCAL VOC test set – bus, car, and sheep using the exemplar model. The plots display results of PASCAL VOC 2006 participants for comparison.

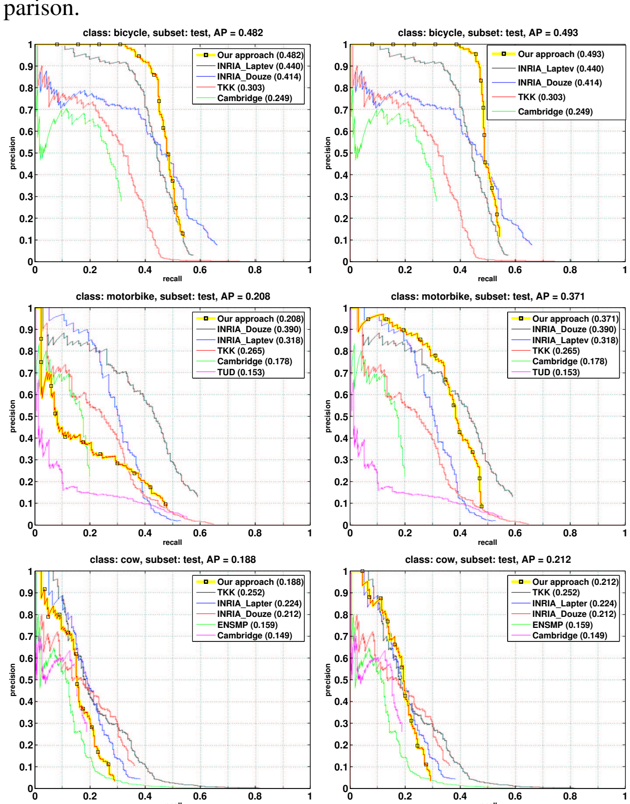

Figure 8. Precision recall curves for bicycles, motorbikes, and cows. Left: results using the exemplar model alone; right: results using an SVM trained on the regions generated by the exemplar model.

Number of training images. Depending on the category, the performance starts to decrease when the number of training images falls below 15 images. The precision recall curves still show high precision, but the recall decreases and the average precision drops by 1-3% per training image removed.

	Airplane	Car	Face	Motorbike
AUC	0.995	0.993	1.000	0.999
EER	0.032	0.019	0.007	0.017

Table 2. **Classification results for CALTECH 4.** The same test sets as [8] is used (but is not trained on Caltech images).

Using ground truth localization. An interesting question is how much we lose by not using the ground truth bounding box provided by the PASCAL VOC annotations. Two experiments were conducted: first, the iterative learning process is initialized by the bounding box of the object; second, the bounding box of the object is taken as a ROI in our model. We have compared the two cases on the car rear category. In the first case, an identical model to our proposed method was learnt. In the second case, the performance decreased by 5.1%. Our explanation for the drop in performance is that the bounding box does not align with the informative features properly, and this negatively affects both the hypothesis generation and optimization steps in the detection phase.

SVM model results. Plots in the right column of figure 8 show the performance achieved training SVMs on top of the exemplar model. For some categories, the results are unchanged by the SVM re-ranking, as there was sufficient discriminative information already present in the exemplar model. However, the results for bicycles and motorbikes, which are classes that are confused using the exemplar model alone, are improved significantly. The differene between the left and right columns of figure 8 demonstrate where the SVM models confers improvements.

5.2. Caltech

In these experiments, we use the same vocabulary of 3000 visual words *trained on the PASCAL VOC 2006 training set*. This makes the task more difficult, compared to [8], as neither airplane nor face images are present in the visual vocabulary construction. Such a situation is closer to a real application, where it may not be possible to rebuild the visual vocabulary with every new class.

For the car and motorbike categories, models trained on PASCAL 2006 are used. For airplane and face categories, we have randomly selected exemplars (15 images each) from the training set and learnt the model. On the Caltech 4 dataset, a classification experiment is carried out. The performance is tested on the same test datasets as [8]. The classification is performed on a test set of the desired class plus background images (there is a different set of background images for the car set). The classification results are summarized in table 2. The performance is comparable to the state of the art. Note, that we are carrying out a more difficult task of classification by detection.

Figure 9. Examples of automotically generated segmentations for the 'solid' classes using a graph cuts algorithm.

6. Extension - Automatic Segmentation

The learned localization of the object is sufficient to compute image color statistics and apply a graph-cut algorithm [4] in order to automatically segment the object out. Examples of such segmentations are given in figure 9. Since the model provides not only a localization, but also a weak mapping between the target and exemplar images, the segmentation can in principle be improved by using information from the exemplars. For example, weak edges between the object and a background of similar colour in the target image can be strengthened using the learnt edge distribution, in the manner of ObjCut [12].

7. Conclusions

We have developed an algorithm for automatically learning regions of a set of training images that correspond to instances of a common object class. Here, we have used these regions, together with the representation used for learning, to form an SVM object class detector. However, the learning algorithm could equally well be used as the starting point for learning a different detection model, e.g. one of the several models that currently requires a bounding box to be provided in the training images [14, 18, 20]. It could also be used to jump start algorithms which are currently very expensive when they explore the entire image during learning, e.g. the constellation model of Fergus *et al.* [8].

Acknowledgements. We are grateful for financial support from the Royal Academy of Engineering and the EU Visiontrain Marie-Curie network, and for discussions with John Winn.

References

[1] A. Bar Hillel, T. Hertz, and D. Weinshall. Efficient learning of relational object class models. In *Proc. ICCV*, 2005.

[2] G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *Proc. CVPR*, 2005.

[3] C. Bouveyron, J. Kannala, C. Schmid, and S. Girard. Object localization by subspace clustering of local descriptors. In *ICVGIP*, 2006.

[4] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, 2001.

[5] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[6] G. Dorkó and C. Schmid. Object class recognition using discriminative local features. *IEEE PAMI*, (Submitted), 2004.

[7] M. Everingham, A. Zisserman, C. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.

[9] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *Proc. CVPR*, 2005.

[10] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proc. ICCV*, 2005.

[11] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods-Support Vector Learning*, 1999.

[12] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *Proc. CVPR*, 2005.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[14] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in CV, ECCV*, 2004.

[15] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999.

[16] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. In *Proc. CVPR*, 2006.

[17] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *Proc. CVPR*, 2006.

[18] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual alphabet. In *Proc. CVPR*, 2006.

[19] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In *Proc. BMVC.*, 2005.

[20] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *Proc. ICCV*, 2005.

[21] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

[22] J. Thureson and S. Carlsson. Appearance based qualitative image description for object class recognition. In *Proc. ECCV*, 2004.

[23] J. Winn and N. Joijic. Locus: Learning object classes with unsupervised segmentation. In *Proc. ICCV*, 2005.

[24] H. Zhang, A. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *Proc. CVPR*, 2006.