# Learning to Localize Objects with Structured Output Regression

Matthew B. Blaschko and Christoph H. Lampert

Max Planck Institute for Biological Cybernetics
72076 Tübingen, Germany
{blaschko,chl}@tuebingen.mpg.de

**Abstract.** Sliding window classifiers are among the most successful and widely applied techniques for object localization. However, training is typically done in a way that is not specific to the localization task. First a binary classifier is trained using a sample of positive and negative examples, and this classifier is subsequently applied to multiple regions within test images. We propose instead to treat object localization in a principled way by posing it as a problem of *predicting structured data*: we model the problem not as binary classification, but as the prediction of the bounding box of objects located in images. The use of a *joint-kernel* framework allows us to formulate the training procedure as a generalization of an SVM, which can be solved efficiently. We further improve computational efficiency by using a branch-and-bound strategy for localization during both training and testing. Experimental evaluation on the PASCAL VOC and TU Darmstadt datasets show that the structured training procedure improves performance over binary training as well as the best previously published scores.

## 1 Introduction

*Object localization*, also called *object detection*, is an important task for image understanding, *e.g.* in order to separate an object from the background, or to analyze the spatial relations of different objects. Object localization is commonly performed using sliding window classifiers [1,2,3,4,5,6]. Sliding window classifiers train a discriminant function and then scan over locations in the image, often at multiple scales, and predict that the object is present in subwindows with high score. This approach has been shown to be very effective in many situations, but suffers from two main disadvantages: (i) it is computationally inefficient to scan over the entire image and test every possible object location, and (ii) it is not clear how to optimally train a discriminant function for localization. The first issue has been recently addressed in [7] by using a branch-and-bound optimization strategy to efficiently determine the bounding box with the maximum score of the discriminant function. We address the second issue in this work by proposing a training strategy that specifically optimizes localization accuracy, resulting in much higher performance than systems that are trained, *e.g.*, using a support vector machine.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 2–15, 2008.

In particular, we utilize a machine learning approach called *structured learning*. Structured learning is the problem of learning to predict outputs that are not simple binary labels, but instead have a more complex structure. By appropriately modeling the relationships between the different outputs within the output space, we can learn a classifier that efficiently makes better use of the available training data. In the context of object localization, the output space is the space of possible bounding boxes, which can be parameterized, *e.g.*, by four numbers indicating the top, left, right, and bottom coordinates of the region. The coordinates can take values anywhere between 0 and the image size, thus making the setup a problem of *structured regression* rather than classification. Furthermore, the outputs are not independent of each other; the right and bottom coordinates have to be larger than the top and bottom coordinates, and predicting the top of the box independently of the left of the box will almost certainly give worse results than predicting them together. Additionally, the score of one possible bounding box is related to the scores of other bounding boxes; two highly overlapping bounding boxes will have similar objectives. By modeling the problem appropriately, we can use these dependencies to improve performance and efficiency of both the training and testing procedures.

The rest of the paper is organized as follows. In Section 2 we discuss previous work in object localization and structured learning and its relation to the proposed method. In Section 3 we introduce the optimization used to train our structured prediction model. The loss function is presented in Section 3.1, while a joint kernel map for object localization is presented in Section 3.2. We discuss a key component of the optimization in Section 4. Experimental results are presented in Section 5 and discussed in Section 6. Finally, we conclude in Section 7.

## 2 Related Work

Localization of arbitrary object classes has been approached in many ways in the literature. Constellation models detect object parts and the relationship between them. They have been trained with varying levels of supervision and with both generative and discriminative approaches [8,9,10]. A related approach has been to use object parts to vote for the object center and then search for maxima in the voting process using a generalized Hough transform [11]. This approach has also been combined with a discriminatively trained classifier to improve performance [12]. Alternatively, [13] have taken the approach of computing image segments in an unsupervised fashion and cast the localization problem as determining whether each of the segments is an instance of the object. Sliding window classifiers are among the most popular approaches to object localization [1,2,3,4,5,6,7], and the work presented in this paper can broadly be seen to fall into this category. The sliding window approach consists of training a classifier, *e.g.* using neural networks [5], boosted cascades of features [6], exemplar models [2,7], or support vector machines [1,3,4,7], and then evaluating the trained classifier at various locations in the image. Each of these techniques rely

on finding modes of the classifier function in the image, and then generally use a non-maximal suppression step to avoid multiple detections of the same object. This of course requires on a classifier function that has modes at the location of objects and not elsewhere. However, while discriminatively trained classifiers generally have high objectives at the object location, they are not specifically trained for this property and the modes may not be well localized. One approach to address this problem is to train a classifier iteratively in a boosting fashion: after each step, localization mistakes are identified and added to the training data for the next iteration, *e.g.* [3,5]. These techniques, however, cannot handle the case when earlier iterations partially overlap with the true object because incorporating these locations would require either an overlap threshold or fractional labels. In contrast, we propose an approach that uses *all* bounding boxes as training examples and that handles partial detections by appropriately scaling the classifier loss. As we show in subsequent sections, we can efficiently take advantage of the structure of the problem to significantly improve results by using this localization specific training.

## 3 Object Localization as Structured Learning

Given a set of input images $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ and their associated annotations $\{y_1, \ldots, y_n\} \subset \mathcal{Y}$, we wish to learn a mapping $g : \mathcal{X} \mapsto \mathcal{Y}$ with which we can automatically annotate unseen images. We consider the case where the output space consists of a label indicating whether an object is present, and a vector indicating the top, left, bottom, and right of the bounding box within the image: $\mathcal{Y} \equiv \{(\omega, t, l, b, r) \mid \omega \in \{+1, -1\},\ (t, l, b, r) \in \mathbb{R}^4\}$. For $\omega = -1$ the coordinate vector $(t, l, b, r)$ is ignored. We learn this mapping in the structured learning framework [14,15] as

$$g(x) = \operatorname{argmax}_y f(x, y) \tag{1}$$

where $f(x, y)$ is a discriminant function that should give a large value to pairs $(x, y)$ that are well matched. The task is therefore to learn the function $f$, given that it is in a form that the maximization in Equation (1) can be done feasibly. We address the issue of maximization in Section 4.

To train the discriminant function, $f$, we use the following generalization of the support vector machine [14]

$$\min_{w,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{2}$$

$$\text{s.t. } \xi_i \geq 0, \quad \forall i \tag{3}$$

$$\langle w, \phi(x_i, y_i)\rangle - \langle w, \phi(x_i, y)\rangle \geq \Delta(y_i, y) - \xi_i, \quad \forall i, \forall y \in \mathcal{Y} \setminus y_i \tag{4}$$

where $f(x_i, y) = \langle w, \phi(x_i, y)\rangle$, $\phi(x_i, y)$ is a joint kernel map implicitly defined by the kernel identity $k\left((x, y), (x', y')\right) = \langle \phi(x, y), \phi(x', y')\rangle$,

$$w = \sum_{i=1}^{n} \sum_{y \in \mathcal{Y} \setminus y_i} \alpha_{iy} \left(\phi(x_i, y_i) - \phi(x_i, y)\right), \tag{5}$$

and $\Delta(y_i, y)$ is a loss function that decreases as a possible output, $y$, approaches the true output, $y_i$. This optimization is convex and, given appropriate definitions of $\phi(x_i, y)$ and $\Delta(y_i, y)$, does not significantly differ from the usual SVM primal formulation except that there are an infeasibly large number of constraints in Equation (4) (the number of training samples times the size of the output space, which can even become infinite, *e.g.* in the case of continuous outputs). We note, however, that not all constraints will be active at any time, which can be seen by the equivalence between Equation (4) and

$$\xi_i \geq \max_{y \in \mathcal{Y} \setminus y_i} \Delta(y_i, y) - \left(\langle w, \phi(x_i, y_i)\rangle - \langle w, \phi(x_i, y)\rangle\right), \quad \forall i \tag{6}$$

which indicates that the $\alpha_{iy}$ in Equation (5) will be sparse. At training time, we can use *constraint generation* to solve the optimization in Equations (2)–(4). Estimates of $w$ are trained using fixed subsets of constraints, and new constraints are added by finding the $y$ that maximize the right hand side of Equation (6). This alternation is repeated until convergence, generally with a small set of constraints compared to the size of $\mathcal{Y}$. We therefore can efficiently optimize the discriminant function, $f$, given a choice of the loss $\Delta(y_i, y)$ and the kernel $k\left((x, y), (x', y')\right)$, as well as a method of performing the maximization in Equation (6). We discuss the loss function in Section 3.1, while we discuss the joint kernel in Section 3.2. A branch-and-bound procedure for the maximization step is explained in Section 4.

### 3.1 Choice of Loss Function

The choice of loss function $\Delta(y_i, y)$ should reflect the quantity that measures how well the system performs. We have chosen the following loss, which is constructed from the measure of *area overlap* used in the VOC challenges [16,17,18]

$$\Delta(y_i, y) = \begin{cases} 1 - \frac{\text{Area}(y_i \bigcap y)}{\text{Area}(y_i \bigcup y)} & \text{if } y_{i\omega} = y_\omega = 1 \\ 1 - \left(\frac{1}{2}(y_{i\omega} y_\omega + 1)\right) & \text{otherwise} \end{cases} \tag{7}$$

where $y_{i\omega} \in \{-1, +1\}$ indicates whether the object is present or absent in the image. $\Delta(y_i, y)$ has the desirable property that it is equal to zero in the case that the bounding boxes given by $y_i$ and $y$ are identical, and is 1 if they are disjoint. It also has several favorable properties compared to other possible object localization metrics [19], *e.g.* invariance to scale and translation. The formulation (7) is attractive in that it scales smoothly with the degree of overlap between the solutions, which is important to allow the learning process to utilize partial detections for training. In the case that $y_i$ or $y$ indicate that the object is not present in the image, we have a loss of 0 if the labels agree, and 1 if they disagree, which yields the usual notion of margin for an SVM. This setup automatically enforces by a maximum margin approach two conditions that are important for localization. First, in images that contain the object to be detected, the localized region should have the highest score of all possible boxes. Second, in images that do not contain the objects, no box should get a high score.

### 3.2 A Joint Kernel Map for Localization

To define the joint kernel map, $\phi(x_i, y)$, we note that kernels between images generally are capable of comparing images of differing size [1,4,20,21]. Cropping a region of an image and then applying an image kernel is a simple and elegant approach to comparing image regions. We use the notation $x|_y$ to denote the region of the image contained within the bounding box defined by $y$, and $\phi_x(x|_y)$ to denote the representation of $x|_y$ in the Hilbert space implied by a kernel over images, $k_x(\cdot,\cdot)$. If $y$ indicates that the object is not present in the image, we consider $\phi_x(x|_y)$ to be equal to the $\mathbf{0}$ vector in the Hilbert space, *i.e.* for all $x'$, $k_x(x|_y, x') = 0$. The resulting joint kernel map for object localization is therefore

$$k((x,y),(x',y')) = k_x(x|_y, x'|_{y'}). \tag{8}$$

Image kernels generally compute statistics or features of the two images and then compare them. This includes for example, bag of visual words methods [22], groups of contours [4], spatial pyramids [1,21], and histograms of oriented gradients [3]. An important property of the joint kernel defined in Equation (8) is that overlapping image regions will have common features and related statistics. This relationship can be exploited for computational efficiency, as we outline in the subsequent section.

## 4 Maximization Step

Since the maximization in Equation (6) has to be repeated many times during training, as well as a similar maximization at test time (Equation (1)), it is important that we can compute this efficiently. Specifically, at training time we need to compute

$$\begin{aligned}&\max_{y\in\mathcal{Y}\setminus y_i} \Delta(y_i, y) + \langle w, \phi(x_i, y)\rangle \\ &= \max_{y\in\mathcal{Y}\setminus y_i} \Delta(y_i, y) + \sum_{j=1}^{n}\sum_{\tilde{y}\in\mathcal{Y}} \alpha_{j\tilde{y}} \left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y)\right)\end{aligned} \tag{9}$$

We therefore need an algorithm that efficiently maximizes

$$\max_{\substack{y\in\mathcal{Y}\setminus y_i \\ y_\omega = y_{i\omega} = 1}} -\frac{\text{Area}(y_i \bigcap y)}{\text{Area}(y_i \bigcup y)} + \sum_{j=1}^{n}\sum_{\tilde{y}\in\mathcal{Y}} \alpha_{j\tilde{y}} \left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y)\right) \tag{10}$$

and for testing, we need to maximize the reduced problem

$$\max_{\substack{y\in\mathcal{Y} \\ y_\omega = 1}} \sum_{j=1}^{n}\sum_{\tilde{y}\in\mathcal{Y}} \alpha_{j\tilde{y}} \left(k_x(x_j|_{y_j}, x_i|_y) - k_x(x_j|_{\tilde{y}}, x_i|_y)\right) \tag{11}$$

The maximizations in Equations (10) and (11) can both be solved using a sliding window approach. In Equation (10), the maximization finds the location in the image that has simultaneously a high score for the given estimate of $w$ and a

high loss (*i.e.* low overlap with ground truth). This is a likely candidate for a misdetection, and the system therefore considers it as a training constraint with the margin scaled to indicate how far the estimate is from ground truth. Because of the infeasible computational costs involved in an exhaustive search, sliding window approaches only evaluate the objective over a subset of possible bounding boxes and therefore give only an approximate solution to Equation (9). This can be viewed as searching for solutions in a strongly reduced set $\hat{\mathcal{Y}} \subset \mathcal{Y}$, where $\hat{\mathcal{Y}}$ includes only the bounding boxes that are evaluated in the sliding window search. However, we can it is more efficient to use a branch-and-bound optimization strategy as in [7], which gives the maximum over the entire set, $\mathcal{Y}$. We adapt this approach here to the optimization problems in Equations (10) and (11).

The branch and bound strategy consists of keeping a priority queue of sets of bounding boxes, which is ordered by an upper bound on the objective function. The algorithm is guaranteed to converge to the globally optimal solution provided the upper bound is equal to the true value of the quantity to be optimized when the cardinality of the set of bounding boxes is equal to one. The sets of bounding boxes, $\tilde{\mathcal{Y}}$, are represented compactly by minimum and maximum values of the top, left, bottom, and right coordinates of a bounding box. This procedure is fully specified given bounding functions, $\hat{h}$, for the objectives in Equations (10) and (11) (Algorithm 1).

**Algorithm 1.** Branch-and-Bound Optimization Procedure

**Require:** image $I \in \mathbb{R}^{n \times m}$
**Require:** quality bounding function $\hat{h}$
**Ensure:** $y = \text{argmax}_{R \subset I} f(R)$
  initialize $P$ as empty priority queue
  initialize $\tilde{\mathcal{Y}} = [0, n] \times [0, m] \times [0, n] \times [0, m]$ indicating the top, left, bottom, and right of the box could fall anywhere in $I$
  **repeat**
    split $\tilde{\mathcal{Y}} \rightarrow \tilde{\mathcal{Y}}_1 \dot{\cup} \tilde{\mathcal{Y}}_2$ by splitting the range of one of the sides into two
    push ( $\hat{h}(\tilde{\mathcal{Y}}_1)$, $\tilde{\mathcal{Y}}_1$ ) and ( $\hat{h}(\tilde{\mathcal{Y}}_2)$, $\tilde{\mathcal{Y}}_2$ ) into $P$
    retrieve top state, $\tilde{\mathcal{Y}}$, from $P$
  **until** $\tilde{\mathcal{Y}}$ consists of only one rectangle, $y$

We note that Equation (11) is simply a linear combination of kernel evaluations between $x_i|_y$ and the support vectors, and therefore is in exactly the form that was solved for in [7]. Bounds were given for a variety of kernels commonly used in the literature for image classification, while bounds for arbitrary kernels can be constructed using interval arithmetic [7]. Similarly, Equation (10) can be bounded by the sum of the bound for Equation (11) and a bound for the overlap term

$$\forall \tilde{y} \in \tilde{\mathcal{Y}}, -\frac{\text{Area}(y_i \bigcap \tilde{y})}{\text{Area}(y_i \bigcup \tilde{y})} \leq -\frac{\min_{y \in \tilde{\mathcal{Y}}} \text{Area}(y_i \bigcap y)}{\max_{y \in \tilde{\mathcal{Y}}} \text{Area}(y_i \bigcup y)}. \tag{12}$$

## 5 Evaluation

For evaluation we performed experiments on two publicly available computer vision datasets for object localization: TU Darmstadt `cows` and PASCAL VOC 2006 (Figures 1 and 2).

**Fig. 1.** Example images from the TU Darmstadt `cow` dataset. There is always exactly one cow in every image, but backgrounds vary.

**Fig. 2.** Example images from the PASCAL VOC 2006 dataset. Images can contain multiple object classes and multiple instances per class.

### 5.1 Experimental Setup

For both datasets we represent images by sets of local SURF descriptors [23] that are extracted from feature point locations on a regular grid, on salient points and on randomly chosen locations. We sample 100,000 descriptors from training images and cluster them using $K$-means into a 3,000-entry visual codebook. Subsequently, all feature points in train and test images are represented by their coordinates in the image and the ID of the corresponding codebook entry. Similar representations have been used successfully in many scenarios for object and scene classification [1,2,7,21,22].

To show the performance of the proposed *structured training* procedure, we benchmark it against *binary training*, which is a commonly used method to obtain a quality function for sliding window object localization [1,2,3,4,5,6,7]. It relies on first training a binary classifier and then using the resulting real-valued classifier function as quality function. As positive training data, one uses the ground truth object boxes. Since localization datasets usually do not contain boxes with explicitly negative class label, one samples boxes from background regions to use as the negative training set. In our setup, we implement this sampling in a way that ensures that negative boxes do not overlap with ground truth boxes or each other by more than 20%. The *binary training* consists of

training an SVM classifier with a kernel that is the linear scalar product of the *bag-of-visual-words* histograms. The SVM's regularization parameter $C$ and number of negative boxes to sample per image are free parameters.

Our implementation of the proposed *structured training* makes use of the `SVMstruct` [14] package. It uses a *constraint generation* technique as explained in Section 3 to solve the optimization problem (2). This requires iterative identification of example-label pairs that most violate the constraints (6). We solve this by adapting the public implementation of the branch-and-bound optimization `ESS` [7] to include the loss term $\Delta$.[1] As in the case of binary training, we use a linear image kernel (8) over the space of bag-of-visual-word histograms. The $C$ parameter in Equation (2) is the only free parameter of the resulting training procedure.

### 5.2 Results: TU Darmstadt Cows

The TU Darmstadt `cow` dataset consists of 111 training and 557 test images of side views of cows in front of different backgrounds, see Figure 1 [24]. The dataset is useful to measure pure localization performance, because each training and test image contains exactly one cow. For other datasets, performance is often influenced by the decision whether an object is present at all or not, which is the problem of classification, not of localization. We train the binary and the structured learning procedure as described in the previous section. First we perform 5-fold cross validation on the training set, obtaining the SVM's regularization parameter $C$ between $10^{-4}$ and $10^{4}$ for both training procedures, and the number of negative boxes to sampled between 1 and 10 for the binary training.

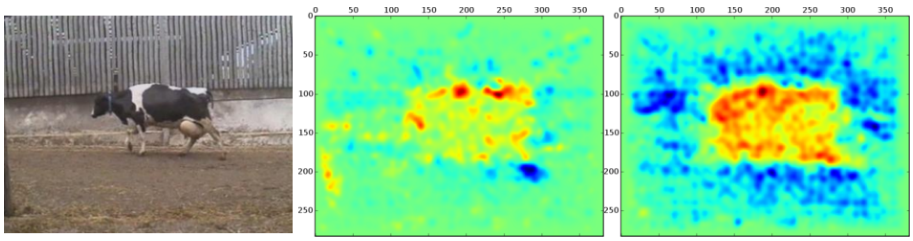


**Fig. 3.** Weight distribution for a TU Darmstadt `cow` test image (best viewed in color). Red indicates positive weights, blue indicates negative weights. Both methods assign positive weights to the cow area, but the structured learning better distributes them across the spatial extent. Additionally, structured learning better learns to give negative weight to image features that lie outside the object.

Afterwards, the systems are retrained on all images in the training set. The resulting systems are applied to the test set, which had not been used in any of the previous steps. We predict three possible object locations per image and

[1] The source code is available at the authors' homepages.

rank them by their detection score (Equation (1)). Figure 3 shows the resulting distribution of weights for an example image in the test set.

The object localization step detect in each image the rectangular region that maximizes the sum of scores, which is a 4-dimensional search space. We visualize the quality function with contour lines of different two-dimensional intersections through the parameter space (Figure 4). The left block of plots shows the quality function for the upper left corner when we fix the lower right corner of the detection box to the ground truth location. The right block shows the quality for the box center when fixing the box dimensions to their ground truth size. Structured training achieves tighter contours, indicating a stronger maximum of the quality function at the correct location.

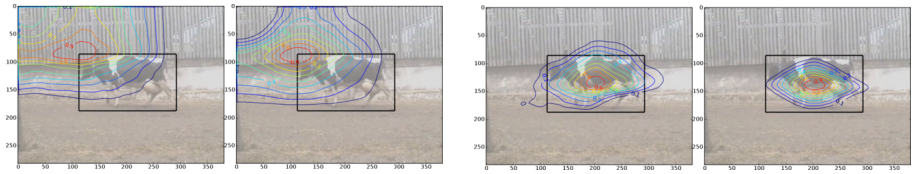

**Fig. 4.** Contour plots of the learned quality function for a TU Darmstadt `cow` test image (best viewed in color). The first and third image corresponds to the quality function learned by binary training, the second and fourth image shows structured training. In left block shows the quality of the upper left corner when fixing the bottom right corner at its ground truth coordinates. The right block shows the quality of the center point when keeping the box dimensions fixed at their ground truth values. Structured learning achieves tighter contours, indicating less uncertainty in localization.

This effect is also shown numerically: we calculate precision–recall curves using the overlap between detected boxes and ground truth as the criterion for correct detections (for details see [12]). Table 1 contains the performance at the point of equal-error rate. The structured detection framework achieves performance superior to binary training and to the previously published methods.

**Table 1.** Performance on TU Darmstadt `cows` dataset at equal error rate. *Binary training* achieves result on par with the best previously reported *implicit shape model (ISM)*, *local kernels (LK)* and their combination *(LK+ISM)* [12]. *Structured training* improves over the previous methods.

| | ISM | LK | LK+ISM | binary training | structured training |
|---|---|---|---|---|---|
| performance at EER | 96.1% | 95.3% | 97.1% | 97.3% | **98.2%** |

### 5.3 Results: PASCAL VOC 2006

The PASCAL VOC 2006 dataset [17] contains 5,304 images of 10 object classes, evenly split into a *train/validation* and a *test* part. The images were mostly downloaded from the internet and then used for the PASCAL challenge on Visual

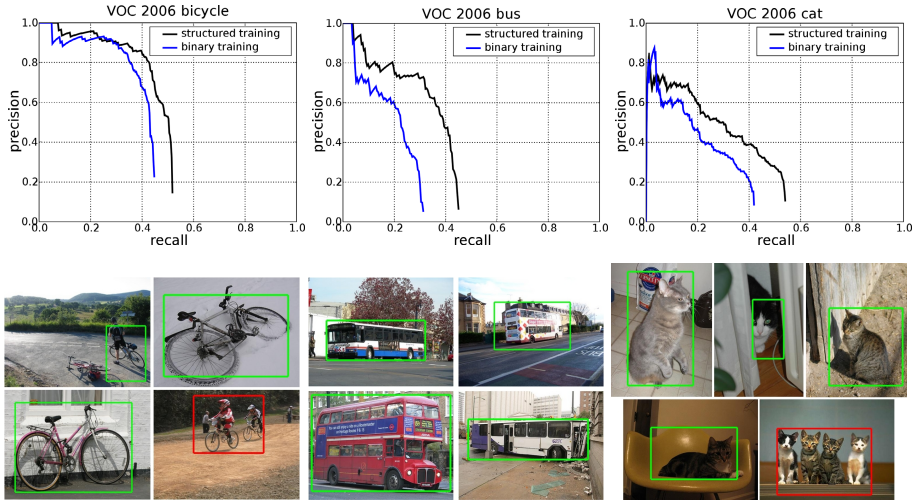


**Fig. 5.** Precision–recall curves and example detections for the PASCAL VOC `bicycle`, `bus` and `cat` category (from left to right). Structured training improves both, precision and recall. Red boxes are counted as mistakes by the VOC evaluation routine, because they are too large or contain more than one object.

Object Categorization in 2006. The dataset contains ground truth in the form of bounding boxes that were generated manually. Since the images contain natural scenes, many contain more than one object class or several instances of the same class. Evaluation is performed based on precision-recall curves for which the system returns a set of candidate boxes and confidence scores for every object category. Detected boxes are counted as correct if their area overlap with a ground truth box exceeds 50% [17].

We use the binary and the structured procedures to train localization systems for all 10 categories. Parameter selection is done separately for each class, choosing the parameter $C$ and number of boxes to sampled based on the performance when trained on the *train* and evaluated on the *val* part of the data. The range of parameters is identical to the TU Darmstadt `cow` dataset. The resulting system is then retrained on the whole *train/val* portion, excluding those which are marked as *difficult* in the ground truth annotation. For the *structured training*, we only train on the training images that contained the object to be detected, while for the *binary training* negative image regions were sampled from images with and without the object present.

The VOC dataset is strongly unbalanced, and in per-class object detection, most test images do not contain the objects to be detected at all. This causes the sliding window detection scores to become an unreliable measure for ranking. Instead, we calculate confidence scores for each detection from the output of a separate SVM with $\chi^2$-kernel, based on the image and box cluster histograms. The relative weight between box and image kernel is determined by cross-validation. The same resulting classifier is used to rank the detection outputs of both training methods.

**Table 2.** Average Precision (AP) scores on the 10 categories of PASCAL VOC 2006. Structured training consistently improves over binary training, achieving 5 new best scores. In one category binary training achieves better results than structured training, but both methods improve the state-of-the-art. Results best in competition were reported in [17]. Results post competition were published after the official competition: $^{\dagger}$[25], $^{\ddagger}$[2], $^{*}$[7], $^{+}$[10].

| | bike | bus | car | cat | cow | dog | horse | m.bike | person | sheep |
|---|---|---|---|---|---|---|---|---|---|---|
| structured training | .472 | **.342** | .336 | **.300** | **.275** | .150 | **.211** | **.397** | .107 | .204 |
| binary training | .403 | .224 | .256 | .228 | .114 | **.173** | .137 | .308 | .104 | .099 |
| best in competition | .440 | .169 | .444 | .160 | .252 | .118 | .140 | .390 | .164 | **.251** |
| post competition | **.498**$^{\dagger}$ | .249$^{\ddagger}$ | **.458**$^{\dagger}$ | .223$^{*}$ | — | .148$^{*}$ | — | — | **.340**$^{+}$ | — |

Figure 5.3 shows the resulting precision–recall curves on the test data for 3 of the categories. For illustration, we also show some example detections of the detection system based on structured learning. From the curves we can see that structured training improves both precision and recall of the detection compared to the binary training. Table 2 summarizes the results in numerical form using the *average precision* (AP) evaluation that was also used in the original VOC challenge. For reference, we also give the results of the best results in the 2006 challenge and the best results from later publications. Object localization with structured training achieves new best scores for 5 of the 10 categories. In all but one category, it achieved better results than the binary training, often by a large margin. In the remaining category, binary training obtains a better score, but in fact both training methods improve over the previous state-of-the-art.

## 6 Discussion

We have seen in the previous sections that the structured training approach can improve the quality of object detection in a sliding window setup. Despite the simple choice of a single feature set and a linear image kernel, we achieve results that often exceed the state-of-the art. In the following we discuss several explanations for its high performance.

First, structured learning can make more efficient use of the possible training data, because it has access to *all* possible boxes in the input images. During the training procedure, it automatically identifies the relevant boxes and incorporates them into the training set, focusing the training on locations where mistakes would otherwise be made. This is in contrast to binary training in which the ground truth object boxes are used as positive examples and negative examples are sampled from background regions. The number of negative boxes is by necessity limited in order balance the training set and avoid degenerate classifiers. However, sampling negative regions prior to training is done "blindly," without knowing if the sampled boxes are at all informative for training.

A second explanation is based on the observation that machine learning techniques work best if the statistical sample distribution is the same during the

training phase as it is during the test phase. For the standard sliding window approach that has been trained as a binary classifier, this is not the case. The training set only contains examples that either completely show the object to be detected, or not at all. At test time, however, many image regions have to be evaluated that contain portions of the object. Since the system was not trained for such samples, one can only hope that the classifier function will not assign any modes to these regions. In contrast, structured training is able to appropriately handle partial detections by scaling the loss flexibly, depending on the degree of overlap to the true solution. Note that a similar effect cannot easily be achieved for a binary iterative procedure: even when iterating over the training set multiple times and identifying wrong detections, only completely false positive detections can be reinserted as negative examples to the training set and made use of in future iterations. Partial detections would require a training label that is neither $+1$ or $-1$, and binary classifiers are not easily adapted to this case.

## 7 Conclusions

We have proposed a new method for object localization in natural images. Our approach relies on a structured-output learning framework that combines the advantages of the well understood sliding window procedure with a novel training step that avoids prediction mistakes by implicitly taking into account all possible object locations in the input image.

The approach gives superior results compared with binary training because it uses a training procedure that specifically optimizes for the task of localization, rather than for classification accuracy on a training set. It achieves this in several ways. First, it is statistically efficient; by implicitly using all possible bounding boxes as training data, we can make better use of the available training images. Second, it appropriately handles partial detections in order to tune the objective function and ensure that the modes correspond exactly to object regions and is not distracted by features that may be discriminative but are not representative for the object as a whole.

The structured training procedure can be solved efficiently by constraint generation, and we further improve the computational efficiency of both training and testing by employing a branch-and-bound strategy to detect regions within the image that maximize the training and testing subproblems. The resulting system achieves excellent performance, as demonstrated by new best results on the TU Darmstadt `cow` and PASCAL VOC 2006 datasets.

In future work, we will adapt our implementation to different image kernels, and explore strategies for speeding up the training procedure. We have only explored a margin rescaling technique for incorporating the variable loss, while a promising alternate formulation would rely on slack rescaling. We plan an empirical evaluation of these alternatives, along with a comparison to related adaptive training techniques, *e.g.* bootstrapping or boosted cascades.

## Acknowledgments

This work was funded in part by the EC project CLASS, IST 027978. The first author is supported by a Marie Curie fellowship under the EC project PerAct, EST 504321. We would like to thank Mario Fritz for making the *TU Darmstadt cow* dataset available to us.

## References

1. Bosch, A., Zisserman, A., Muñoz, X.: Representing Shape with a Spatial Pyramid Kernel. In: CIVR (2007)
2. Chum, O., Zisserman, A.: An Exemplar Model for Learning Object Classes. In: CVPR (2007)
3. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, pp. 886–893 (2005)
4. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of Adjacent Contour Segments for Object Detection. PAMI 30, 36–51 (2008)
5. Rowley, H.A., Baluja, S., Kanade, T.: Human Face Detection in Visual Scenes. In: NIPS, vol. 8, pp. 875–881 (1996)
6. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. CVPR 1, 511 (2001)
7. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In: CVPR (2008)
8. Fergus, R., Zisserman, P.P.A.: Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition. IJCV 71, 273–303 (2007)
9. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: CVPR, Washington, DC, USA, pp. 710–715. IEEE Computer Society Press, Los Alamitos (2005)
10. Felzenszwalb, P., McAllester, D., Ramanan, D.: A Discriminatively Trained, Multiscale, Deformable Part Model. In: CVPR (2008)
11. Leibe, B., Leonardis, A., Schiele, B.: Combined Object Categorization and Segmentation with an Implicit Shape Model. In: Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic (2004)
12. Fritz, M., Leibe, B., Caputo, B., Schiele, B.: Integrating representative and discriminative models for object category detection. In: ICCV, pp. 1363–1370 (2005)
13. Viitaniemi, V., Laaksonen, J.: Techniques for Still Image Scene Classification and Object Detection. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 35–44. Springer, Heidelberg (2006)
14. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML, p. 104 (2004)
15. Bakır, G.H., Hofmann, T., Schölkopf, B., Smola, A.J., Taskar, B., Vishwanathan, S.V.N.: Predicting Structured Data. MIT Press, Cambridge (2007)
16. Everingham, M., et al.: The 2005 PASCAL Visual Object Classes Challenge. In: Selected Proceedings of the First PASCAL Challenges Workshop, pp. 117–176. Springer, Heidelberg (2006)
17. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results (2006), `http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf`

18. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007), http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
19. Hemery, B., Laurent, H., Rosenberger, C.: Comparative study of metrics for evaluation of object localisation by bounding boxes. In: ICIG, pp. 459–464 (2007)
20. Eichhorn, J., Chapelle, O.: Object Categorization with SVM: Kernels for Local Features. Technical Report 137, Max Planck Institute for Biological Cybernetics, Tübingen, Germany (2004)
21. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: CVPR, pp. 2169–2178 (2006)
22. Nowak, E., Jurie, F., Triggs, B.: Sampling strategies for bag-of-features image classification. In: ECCV, pp. 490–503 (2006)
23. Bay, H., Tuytelaars, T., Van Gool, L.J.: SURF: Speeded Up Robust Features. In: ECCV, pp. 404–417 (2006)
24. Magee, D.R., Boyle, R.D.: Detecting Lameness Using 'Re-Sampling Condensation' and 'Multi-Stream Cyclic Hidden Markov Models'. Image and Vision Computing 20, 581–594 (2002)
25. Crandall, D.J., Huttenlocher, D.P.: Composite models of objects and scenes for category recognition. In: CVPR (2007)