Beyond Bounding-Boxes: Learning Object Shape by Model-Driven Grouping

Antonio Monroy and Björn Ommer

Interdisciplinary Center for Scientific Computing, University of Heidelberg, Germany
{antonio.monroy,bommer}@iwr.uni-heidelberg.de

Abstract. Visual recognition requires to learn object models from training data. Commonly, training samples are annotated by marking only the bounding-box of objects, since this appears to be the best tradeoff between labeling information and effectiveness. However, objects are typically not box-shaped. Thus, the usual parametrization of object hypotheses by only their location, scale and aspect ratio seems inappropriate since the box contains a significant amount of background clutter. Most important, however, is that object shape becomes only explicit once objects are segregated from the background. Segmentation is an ill-posed problem and so we propose an approach for learning object models for detection while, simultaneously, learning to segregate objects from clutter and extracting their overall shape. For this purpose, we exclusively use bounding-box annotated training data. The approach groups fragmented object regions using the Multiple Instance Learning (MIL) framework to obtain a meaningful representation of object shape which, at the same time, crops away distracting background clutter to improve the appearance representation.

1 Introduction

Recognizing and localizing all instances of object categories in a novel query image is a core task on the way to automatic scene understanding. Object detection typically proceeds by localizing object bounding-boxes (e.g. [1]), which are parameterized by their location, scale and aspect ratio. A classifier is then evaluated for each detection window, thereby providing hypotheses that are ranked by their score. Such approaches have proven to be very successful in benchmarks, but there are two issues that remain unresolved. First, objects are not box-shaped and so the detection window contains a significant amount of background clutter that tends to deteriorate the whole window's classification result. And indeed, even complex models like [1] are eventually based on a holistic representation of the whole bounding-box, including the clutter. The second problem is that object shape becomes only available for detection once the object has been segregated from the background. To overcome both problems, not only background suppression is required, but also a reasoning about the object shape is essential. Recent work (e.g. [2]) in the field of segmentation has shown that relying only on low-level cues is not enough. Furthermore, it appears reasonable

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part III, LNCS 7574, pp. 582–595, 2012.

to combine class-specific top-down information to achieve better results. The purpose of this paper is to learn object models for detection by explicitly representing object shape and segregating it from the background, however, *without* requiring manual segmentation of the training samples. Therefore, we propose a model-based approach that does not require supervision, but automatically learns object shape and appearance while segregating objects from background. Since we use more than a mere bottom-up segmentation, we are able to capture the overall object shape in a model-driven manner by grouping the corresponding foreground regions.

Detection approaches which use shape information can be classified according to the degree of supervision they require during training. On the one hand, methods like ([3–5]) require ground-truth pixel-wise segmentation masks during training. The main disadvantage is that such information is usually not available for large-scale detection tasks or is tedious to obtain and so we are proposing an automatic MIL learning-based approach to circumvent these shortcomings. On the other hand, we have methods which only require bounding-box information during training [6–12]. These methods differ in the way, how shape information is integrated into the detection task. Pure bottom-up methods [6, 7] are susceptible to segmentation artifacts. While [6] directly classifies bottom-up generated segments using a k-nearest neighbor classifier, [7] computes hierarchical segmentations to find object subtrees similar to those learned during training. [8, 9] can be viewed as top-down approaches. [8] divides the bounding-box into cells and infers an occlusion map by clustering the response scores of a linear SVM on each cell, where occluded regions are defined as the groups with a negative overall response. This approach does not use any shape information to train the linear SVM. Furthermore, negative response scores can also be caused by occlusion or by other factors, such as background or an uncommon shape. Based on the model of [13], [9] attempts to capture the object's shape by means of a fixed number of coarse box-shaped patterns. Finally, methods like [11, 10, 12] attempt to combine bottom-up and top-down cues. Gu et al. [11] proposed a method for detection using regions. Starting with regions as the basic elements, a generalized Hough-like voting strategy for generating hypotheses is used (see [14] for improvements to the idea of voting). The method's drawbacks are twofold. First, it needs a general sliding window classifier for verification, which does not take shape into account. Second, ground-truth pixel-wise segmentation masks for the training data are required. Recently, [10] proposed a method for object detection based on the category-independent figure-ground segmentation masks of [15]. To train with only bounding-box information, the authors assume that the best ranked segment within the bounding-box covers the entire object. This segment is thus used to learn a regression function that predicts the quality of query segments. Consequently, the performance of their detection system is highly susceptible to the fact that the first bottom-up generated segment actually covers the entire object. In datasets like PASCAL VOC we observe that in many images this assumption is too strong. Finally, [12] utilizes multiple over-segmentations to propose class-independent bounding-boxes for classification. However, the

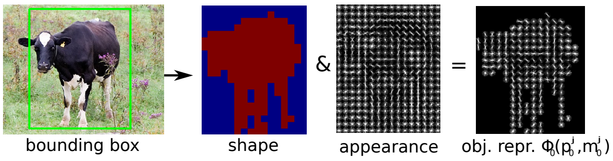

Fig. 1. Object representation (best viewed in color). We divide the bounding-box into cells and calculate features on each cell. Inferring a foreground segmentation cell-mask from unsegmented training data, we suppress the background features by setting the corresponding cells to zero.

authors discard the shape information contained in the super-pixels. They sample at each pixel 5 different color features and utilize them within a standard bag-of-words model to classify the object.

The paper is organized as follows. In Sec. 2 we explain how to suppress the background and represent the shape of an object and in Sec. 3 we then describe how to learn our model (without using pixel-wise segmentation masks in the training stage) and infer the shape of an object using a MIL framework.

2 Model

2.1 Suppressing the Background

Detection window approaches like [16, 1] have demonstrated a good performance in difficult benchmarks. Consequently, such a framework offers us a good basis to implement our idea. The detection window is commonly divided into a grid of cells and we learn object shape to suppress cells in the clutter and concentrate on the actual object. In this section we describe how to model a foreground/background segregation.

Suppose an object $\mathcal{O}^j$ within image I is given and we assume for a moment a pixel-wise foreground object's segmentation is also given. In the next section we will describe how to automatically learn a cell-accurate shape estimation for the object's foreground.

First, we divide the bounding-box j into an array of size $l_0 \times h_0$. For each cell we calculate a $d-$dimensional feature. This $l_0 \times h_0 \times d$ matrix is called $\hat{\phi}_0(p_0^j)$, where $p_0^j = (x, y)$ is the top-left position of the bounding-box in image I. Specifically, in this paper we use histograms of oriented gradients (HoG) as features. These widely used and fast to calculate descriptors capture the edge or gradient structure that is very characteristic of local shape. Additionally, they exhibit invariance to local geometric and photometric transformations ([1, 16]). However, our framework is independent of this specific choice of features. A combination of different descriptors (e.g. like in [12]) can be integrated into our model and should enable further performance improvements.

The foreground of an object is modeled by defining a binary vector $m_0^j \in \mathbb{B}^{1 \times l_0 h_0}$. This vector contains ones if the corresponding cell is covered by the

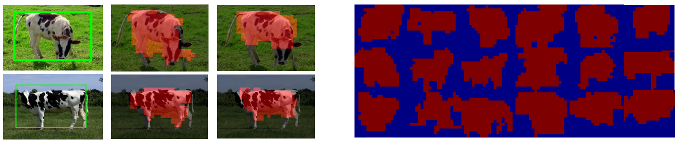

Fig. 2. Left: the first column shows a detection and the last two columns the two most similar prototypical segments. Right: Subset of prototypical segments for the category cow.

object, otherwise it is zero. We call this vector m_0^j the root-cell mask for object $\mathcal{O}^j$ — part-cell masks are introduced in Sec. 4.1. Using m_0^j we set to zero the cells of $\hat{\phi}_0(p_0^j)$ corresponding to the zero entries in m_0^j. Formally, the foreground representation of object $\mathcal{O}^j$ is defined as

$$\phi_0(p_0^j, m_0^j) := (m_0^j \otimes \mathbf{1}_d) \odot \hat{\phi}_0(p_0^j), \tag{1}$$

where $\otimes$ defines the Hadamard-Product and $\odot$ the element-wise multiplication. Fig. 1 shows how to suppress the background of a bounding-box if a root-cell mask for the object's foreground is given.

2.2 Matching Objects

Due to different pose variations, occlusion and clutter, the foreground root-cell masks m_0^j and m_0^u of two objects may differ substantially. Therefore, building an euclidean dot product between the feature representations $\phi_0(p_0^j, m_0^j)$ and $\phi_0(p_0^u, m_0^u)$ as [3] or [1] do, will lead to unstable matching scores. Rather than using a simple dot product, we represent each object with a prototypical set of shape segments $C_0 = \{\bar{m}_0^\iota\}_{\iota=1}^{\nu}$. This set of segments is automatically learnt from unsegmented training data (see section 3.3 for more details). The idea is to reduce the high intra-class shape variability by using a reduced number of typical class-specific views of its shape. We then use a weighted sum to match both representations. Precisely, the matching score is given by

$$\begin{aligned} &d_0(\phi_0(p_0^j, m_0^j),\ \phi_0(p_0^u, m_0^u)) := \\ &\qquad \frac{1}{\nu}\sum_{\iota=1}^{\nu} < a(m_0^j, \bar{m}_0^\iota)\phi_0(p_0^j, \bar{m}_0^\iota), a(m_0^u, \bar{m}_0^\iota)\phi_0(p_0^u, \bar{m}_0^\iota) >, \end{aligned} \tag{2}$$

where

$$a(m_0^j, \bar{m}_0^\iota) := exp\left(-\beta * \frac{\|m_0^j - \bar{m}_0^\iota\|_2}{|\bar{m}_0^\iota|}\right) \tag{3}$$

represents the dissimilarity score between the root-cell mask m_0^j and the prototypical root-cell mask $\bar{m}_0^\iota$. The parameter β is obtained by cross-validation. In our experiments, we obtained an optimal value in the range of 1.1 ± 0.1 for the different object classes. Here $|\bar{m}_0^\iota|$ represents the total number of active cells in the prototypical root-cell mask $\bar{m}_0^\iota$.

Equation (2) induces a Mercer kernel, since the sum of Mercer kernels is a Mercer kernel again. By the "Kernel Trick" we know, that there exists a (possibly unknown) transformation $\varPhi$ into a space in which the kernel (2) is a scalar product. To keep the notation simple we identify $\mathcal{O}^j := \varPhi(\phi_0(p_0^j, m_0^j))$ and refer to this scalar product as

$$< \mathcal{O}^j, \mathcal{O}^u >_{CB} := d_0(\phi_0(p_0^j, m_0^j), \phi_0(p_0^u, m_0^u)). \tag{4}$$

In praxis we do not require to evaluate the function $\varPhi$ to learn our model, but use the kernel values instead. By defining the kernel (2) we have integrated both of our goals into the detection window approach: We suppress the features corresponding to the background and robustly represent the shape of an object through a prototypical set of shapes.

3 Learning

Let's assume for the moment that for all objects $\mathcal{O}^j$ in the training data their root-cell masks m_0^j containing the whole object foreground are given. The training set is denoted by $\{(\mathcal{O}^j, y^j)\}$. Here $y^j \in \{1, -1\}$ denotes the label of object $\mathcal{O}^j = \varPhi(\phi_0(p_0^j, m_0^j))$. In this special case, we could easily learn a discriminative function

$$f(\phi_0(p_0^q, m_0^q)) = \sum_{i \in SV} -y_i \alpha_i d_0(\phi_0(p_0^q, m_0^q), \phi_0(p_0^i, m_0^i)) + b \tag{5}$$

to classify the query object $\phi_0(p_0^q, m_0^q)$ (SV is the set of support vectors). However, in contrast to [3], we are not provided with the foreground root-cell masks m_0^j during training, but rather we automatically learn them from unsegmented training data. This is described in the next section. Similar to [3], [10] assumes that the best-ranked foreground segmentation mask of [15] covers the whole object. In practice this assumption is, however, not valid: The second row of figure 3 shows the best ranked CMPC segments that lie within the object bounding-box. None of them covers exactly the whole object.

3.1 Learning from Unsegmented Training Data

The question is now how to learn the classification function f if the foreground root-cell masks m_0^j are not given during training?

Given a discriminatively trained function f, the problem of inferring the foreground root-cell mask m_0^j for an object $\mathcal{O}^j$ can be formulated as

$$m_0^j = \operatorname*{argmax}_{m_0} f(\phi_0(p_0^j, m_0)) \tag{6}$$

i.e. the inference (6) is tackled by grouping cells in a model-driven, top-down manner so as to maximize the classification score.

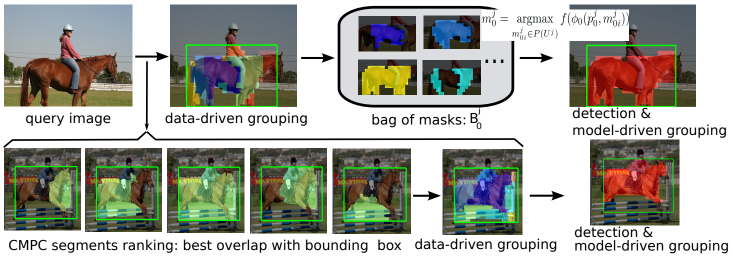

Fig. 3. First row: We simultaneously detect and infer the object foreground. Second row: We show our data-driven grouping from which we infer the foreground of our object. For complex categories we can not assume that the first CMPC segment covers the whole object.

We simultaneously learn the function f and solve the grouping problem by formulating our problem in the Multiple Instance Learning (MIL) framework. Here, a bag contains features corresponding to different root-cell masks. For positive instances, at least one of these features correspond to the foreground of an object. In the ideal case, a bag B_0^j would contain all possible combinations of cells within the bounding-box. Since this is not tractable, we describe in the next section how to create a shortlist of meaningful groups in a bottom-up manner. Suppose we obtain l different groups for a bounding-box j. The i-the group is represented by a root cell mask m_{0i}^j and build the set $U^j := \{m_{0i}^j\}_{i=1}^l$. A bag is then defined as

$$B_0^j := \left\{\phi_0(p_0^j, m_{0i}^j) | m_{0i}^j \in P(U^j)\right\}_{i=1}^{|P(U^j)|}, \tag{7}$$

where $P(U^j)$ is the power set of U^j. If the bounding-box contains an object, the label Y_j of the bag B_0^j is set to 1, otherwise it is -1. Using our kernel (2) the problem of learning the function f transform into:

$$\min_{w_0,b,\xi} \frac{1}{2}\|w_0\| + C\sum_I \xi_I \tag{8}$$

$$s.t. \;\; \forall I : Y_I \max_{i \in I}(< w_0, \mathcal{O}_i^I >_{CB} + b) \geq 1 - \xi_I, \xi_I \geq 0, \tag{9}$$

here $\mathcal{O}_i^I = \Phi(\phi_0(p_0^I, m_{0i}^I))$ are object hypotheses and denote the elements within the bag I. Once the function f is learned, the inference problem (6) for a query image is transformed into

$$m_0^j = \underset{m_{0i}^j \in P(U^j)}{\text{argmax}} \; f(\phi_0(p_0^j, m_{0i}^j)) \tag{10}$$

In other words, in (10) we look for the “most“ positive instance within B_0^j and by doing this, we indirectly infer the corresponding root-cell segmentation mask

m_0^j (s. first row of Fig. 3). In practice the optimization problem (8) is solved by alternating the calculation of the hyperplane w_0 and bias b with the calculation of the margin for the positive bags: $Y_I \max_{i \in I}(< w_0, \mathcal{O}_i^I >_{CB} +b)$ (we used the MIL solver of [17] but other approaches like [18] could also be used). This means that for every positive bag we fix the "most" positive instance and then we use all other instances of the negative bags to learn a SVM using our Mercer kernel (2). In our experiments we used this MIL formulation since it is effective, fast (convergence is reached after a few iterations) and the performance was robust for varying initializations. Specifically, we randomly chose an element for every positive bags to initialize the algorithm.

In the first row of Fig. 3 we visualize the inference of the final foreground root-cell mask m_0^j, given a data-driven grouping of cells for the bounding-box.

3.2 Data-Driven Grouping

In this section we describe how to create a shortlist of candidate groups by means of a data-driven grouping of cells for a given bounding-box. This is necessary to render the inference problem (6) and the creation of bags (7) feasible.

Recently, [15] presented the combinatorial CMPC algorithm for generating a set of binary figure-ground segmentation hypotheses $\{S_t^I\}_{t=1}^{N_s}$ for an image I. In general we can not assume (see second row of Fig. 3) that the best ranked segment covers the whole object (as in [10]). However, the pool of CMPC segments yields a good basis to obtain groups of pixels, which cover only parts of the object. An example of our grouping can be seen in Fig. 3 (second row).

Given a bounding-box BB_j in image I, the idea is to first weight each pixel-wise segment S_t^I generated by [15] with the ratio between the number of pixels p_{kl} belonging to the segment S_t^I which lie outside the bounding-box and the total number of pixels covered by the bounding-box $|BB_j|$ itself:

$$r_t^j := \frac{1}{|BB_j|} \sum_{kl} \mathbb{1}_{[p_{kl} \in S_t^I]} \mathbb{1}_{[p_{kl} \notin BB_j]}. \tag{11}$$

Only segments S_t^I that fully lie within the bounding-box will get high scores, while straddling segments will be penalized. We then take the weighted sum of all segments which intersect the bounding-box and build a density map for this bounding-box

$$\mathcal{H}_{kl}^j := \frac{1}{N_s} \sum_t^{N_s} r_t^j * \mathbb{1}_{[p_{kl} \in S_t^I]} \mathbb{1}_{[p_{kl} \in BB_j]}. \tag{12}$$

The values in this map map indicate which regions within the bounding-box were consistently covered by CMPC segments S_t^I. We then apply a mean-shift clustering algorithm on this 2D density map $\mathcal{H}^j$ and enforce the connectedness of each of the resulting groups. The cells covering each of these groups define the root-cell masks m_{0i}^j used to construct the bags in equation (7).

In practice, for bounding boxes containing an object, we typically obtain between 6 and 8 groups. For boxes in the background, our grouping algorithm typically does not generate any segment, since these regions are not covered by a CMPC segment (the weights in Eq. 11 are zero). This situation renders feasible an exhaustive search using inference (6). We favor mean-shift over other clustering methods because it allows an adaptive bandwidth for different clusters.

3.3 Learning a Prototypical Set of Segments

Our goal is to represent every object through a prototypical set of segments. In section 2.2 we use such a representation to robustly match different object instances. In this section we describe how to learn such a prototypical set.

The idea is to explain the shape complexity of a class through a reduced number of segments that are typical for a certain class. Using the bottom-up grouping described in section 2.2, we obtain for every positive training sample j a bag B_0^j. To find those specific segments that appear frequently within the class, we hierarchically cluster the elements of all positive bags (e.g. using Ward's method). Every group is then represented by its medoid, i.e. the element with the minimal average dissimilarity (using measure (3)) to all the objects in the cluster. The set of all medoids define the prototypical set of segments $C_0 = \{\bar{m}_0^\iota\}_{\iota=1}^{\nu}$ used to train our model. The number of clusters is chosen using cross-validation and ranges between 10 and 40 segments (s. Fig. 2).

4 Results

4.1 Implementation Details

We use a sliding window detection model similar to [1] to implement our idea. The model in [1] describes an object $\mathcal{O}^j$ by means of a bounding-box covering the entire object (root window) as well as eight smaller windows (about half the size) that cover parts of the root window. Every part window i is divided into a grid of cells of size $l_i \times h_i,\ i = 1 \ldots 8$ and a HoG feature is calculated for every cell. During training, weights (used as linear filters) are learnt for the root window and additional 8 linear filters are trained for the parts. In our case, if we ignore the parts for a moment, we first would need to learn the prototypical set of segments using the positive training samples as described in Sec. 3.3 and then learn the classifier (Eq. 5) as described in Sec. 3.1. To include the concept of parts from [1], we will first introduce the notion of a bag for each of the part windows and then extend our matching kernel (2) also for these parts. Thereafter, the corresponding classifier can be trained analogously to [1] and thus we remit to that work for further details.

Modeling Parts: Running the bottom-up grouping described in section 3.3 exclusively on the root window, results in the bags B_0^j for each training sample j (τ being the number of instances in each bag). We then define a bag B_i^j for each of the part-windows as follows:

$$B_i^j := \{\phi_i(p_i^j, m_{ik}^j)\}_{k=1}^{\tau},\ i = 1 \ldots 8. \tag{13}$$

Here p_i^j denotes the position of the i-th part-window for sample j. The binary vector $m_{ik}^j \in \mathbb{B}^{l_i h_i}$ denotes the k-th part-cell segmentation mask of part i. It is obtained by taking the overlap of part i with the root-cell mask $m_{0k}^j \in \mathbb{B}^{l_0 h_0}$. In doing so, we obtain the feature representation $\phi_i(p_i^j, m_{ik}^j)$ for the i-th part (similar to Eq. (1)). Following this notation, the matching score of Eq. 2 between two objects $\mathcal{O}^j, \mathcal{O}^u$ can be extended to include parts,

$$d(\mathcal{O}^j, \mathcal{O}^u) := \sum_{i=0}^{8} d_i(\phi_i(p_i^j, m_i^j), \phi_i(p_i^u, m_i^u)) + < p_i^j - p_0^j, p_i^u - p_0^u > . \qquad (14)$$

Here the last term compares the displacement of the i-th part w.r.t. the object center. $d_i(.,.)$ denotes the matching score for part i defined as in Eq. (2). To obtain $d_i(.,.)$ we also use a set of prototypical segments to represent each one of the parts. This set is obtained in a similar way as for the root window by hierarchically clustering the elements of all positive bags B_i^j. In practice, 7 prototypical segments are used to represent each part. Using the kernel (14) we are then capable of learning a discriminative function along the lines of (5).

4.2 Experimental Results

The purpose of our experiments is to show that if only bounding-box annotated data is available during training, using a top-down generated prototypical representation of the object shape, as well as suppressing the background within a bounding-box, helps to improve pixel-wise object detection.

The methods of [10] and [11] are the most similar to ours and therefore provide us with a baseline for our results. Both methods present pixel-wise detection results exclusively on the ETHZ-Shape dataset ([19]). Specifically, [10] also presents results for the PASCAL segmentation challenge. However, this challenge assesses a simpler problem than that in our paper, since pixel-wise segmentation masks are used for training the model. For purposes of comparison with state-of-the-art [10, 11] we also use the ETHZ-Shape dataset to test our model's performance. Larger and more complex datasets for object detection (e.g. INRIA Horses or PASCAL VOC) are suboptimal to demonstrate this papers' purpose, since there are no pixel-wise masks for the whole test-set and measuring detection performance is only possible up to a bounding-box.

[10] is currently the state-of-the-art for pixel-wise detection on the ETHZ-Shape dataset. This dataset contains 5 object categories and 255 images. We follow the experimental settings in [19]. The image set is evenly split into training and testing sets and performance is averaged over 5 random splits. Following [11] and [10], we report pixel-wise average precision (AP) on each class. The PASCAL criterion is used to decide if a detection is correct. The ground-truth segmentation masks were provided by [11].

Our results are displayed in table 1. Our method outperforms the state-of-the-art approach of [10] by 7% mean AP and our detection rate is comparable with the detection rate at 0.02, 0.3 and 0.4 FPPI in [10] (see table 2).

Table 1. Detection results for the ETHZ-Shape dataset. Performance is measured as *pixel-wise* AP over 5 trials, following [10, 11]. For completeness, we include the performance of [1] measured using a bounding-box parametrization. We improve state-of-the-art by 7% AP.

	Our Method	Carreira etal. [10]	Gu etal. [11]	Felz. etal.[1]
Apples	0.963 ± 0.023	0.890 ± 0.019	0.772 ± 0.112	0.934 ± 0.048
Bottles	0.877 ± 0.011	0.900 ± 0.021	0.906 ± 0.015	0.891 ± 0.028
Giraffes	0.823 ± 0.038	0.754 ± 0.019	0.742 ± 0.025	0.817 ± 0.048
Mugs	0.885 ± 0.037	0.777 ± 0.059	0.760 ± 0.044	0.856 ± 0.073
Swans	0.927 ± 0.023	0.805 ± 0.028	0.606 ± 0.013	0.813 ± 0.125
Mean	**0.896** ± 0.026	0.825 ± 0.012	0.757 ± 0.032	0.862 ± 0.051

For the sake of completeness, we also evaluate our model on the level of bounding-boxes for the detected objects (standard setting). We used the INRIA horses dataset, which contains 340 images. Half of the images contain one ore more horses and the rest are negative images. 50 horse images and 50 negative images are used for training. The remaining 120 horse images plus 120 negative images are used for testing. Results are listed in figure figure 5. Compared to [1], we improve state-of-the-art detection rate at 0.1 fppi by 3.5% achieve a gain of 29% to the recent segmentation-based approach of [20].

Table 2. Detection rate at 0.02, 0.3 and 0.4, fppi on ETHZ-Shape. We reach comparable pixel-wise detection rates to [10].

	Our Method	Carreira etal. [10]	Gu etal. [11]	Felz. etal.[1]
Apples	0.985/0.985/0.985	0.904/0.941/0.941	0.697/0.854/0.916	0.956/0.989/0.989
Bottles	0.860/0.975/0.975	0.891/0.975/0.975	0.745/0.932/0.958	0.835/0.981/0.981
Giraffes	0.830/0.924/0.924	0.920/0.970/0.970	0.543/0.736/0.800	0.675/0.936/0.943
Mugs	0.896/0.956/0.956	0.812/0.925/0.925	0.496/0.816/0.833	0.816/0.932/0.937
Swans	0.934/1/1	0.983/1/1	0.569/0.800/0.800	0.835/0.919/0.919
Mean	0.901/**0.968**/**0.968**	**0.902**/0.963/0.963	0.594/0.829/0.861	0.824/0.951/0.954

Next, we evaluate the impact of our bottom-up grouping (see section 3.2) during training. For this experiment, the union of the first n best-ranked CMPC segmentation masks of [15] lying within the bounding-box was taken to define the bags (13). This setting would be equivalent to [10], which assumes that the best bottom-up generated segment covers the whole object. We varied the number of segments n and measured the detection performance in terms of average precision (AP). The experiment was evaluated on the horse category of PASCAL VOC 2007. The result is plotted in the left side of figure (6). For large n the performance reaches that of [1], since eventually all cells of m_0^j are active. Conversely, performance significantly drops as we approach n=1, which is the

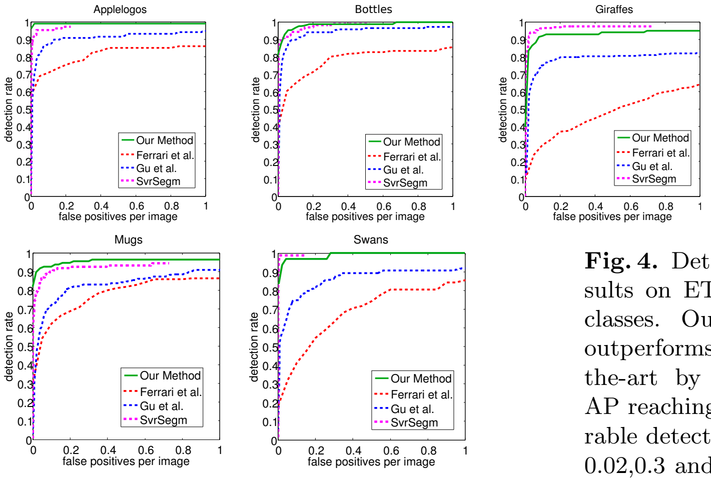

Fig. 4. Detection Results on ETHZ-Shape classes. Our method outperforms state-of-the-art by 7% mean AP reaching a comparable detection rate at 0.02,0.3 and 0.4 FPPI

Method	AP	Det.rate at 0.1 FPPI
Our Method	**0.883**	**0.902**
[21]	-	0.730
BoSS [20]	-	0.630
[22]	-	0.652
[23]	-	0.674
[1]	0.871	0.867

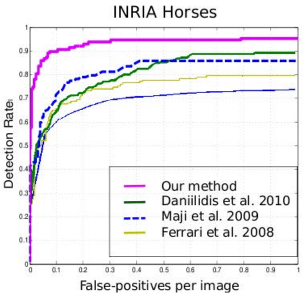

Fig. 5. Detection results for the INRIA horses dataset. We improve [1] by 3.5% and the segmentation-based approach [20] by 29.9% detection rate at 0.1 FPPI.

setting of [10]. Our full model is plotted as a constant line, since it is independent of the number of segments generated by [15].

In a second experiment, we tested the impact of our bottom-up grouping during testing. Instead of obtaining a bottom-up grouping for each sliding window, we tested our model exclusively on all the CMPC segments. We considered the tight bounding-box around each figure-ground segment S_t^I for an image I and used this segment to construct the bags B_i^j (in this case we have as many bags as segments S_t^I, see Eq. (13)). The experiment was carried out using the car category of VOC 2007 (see right plot in figure 6). We observed a 7.3% performance drop in AP. Hence, it is advisable to combine the different segments S_t^I (as we do) to obtain a better detection performance.

We also tested the impact of using a prototypical set of segments (see section 3.3) to represent object shape. Since the matching score (14) use a prototypical set of segments to evaluate each $d_i(.,.)$, we trained in this experiment a linear SVM using the euclidean dot product (instead of using $d_i(.,.)$) between the feature representations $\phi_i(p_i^j, m_i^j)$ for all parts. In this case the matching score (14) is transformed into

$$\hat{d}(\mathcal{O}^j, \mathcal{O}^u) := \sum_{i=0}^{8} < \phi_i(p_i^j, m_i^j), \phi_i(p_i^u, m_i^u) > + < p_i^j - p_0^j, p_i^u - p_0^u > . \quad (15)$$

In doing so, we obtained a very poor performance of 0.45 AP for the horse category compared to the 0.578 AP of our model.

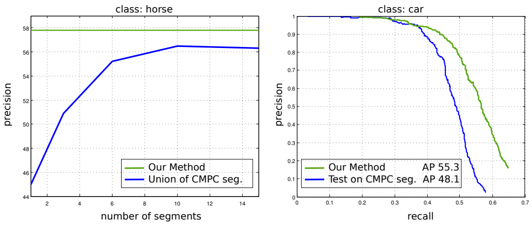

Fig. 6. Impact of bottom-up grouping. Left: We trained our model using the union of the n best-ranked CMPC segments. Right: Test exclusively on CMPC segments.

To the best of our knowledge, there is no approach which explicitly tries to infer the overall object form using a model exclusively learnt from bounding-box annotated training data for any category in the PASCAL dataset. In order to compare our approach with other detection methods we evaluate our model using the standard setting on those PASCAL VOC 2007 categories, where [1] best performs. In table 3 and figure 7, we observe that our model exhibits robust performance (43.68 MAP or Mean Average Precision) under challenging image conditions at the same time that we obtain a richer output than just a bounding-box for detection. While [1] (42.34 MAP) is considered as our baseline model, we also listed comparable state-of-the-art detection methods. Due to the lack of exact precision numbers, the multi-feature approach of [12] is not listed in table 3. However, from the diagram presented in their paper, we read an approximate MAP of 42 for this set of categories. and of 40 if [1] is evaluated exclusively on the proposed windows. Regardless of this, the strength of [12] remains in the usage of 5 different color features to train a Bag-Of-Words model. While we use

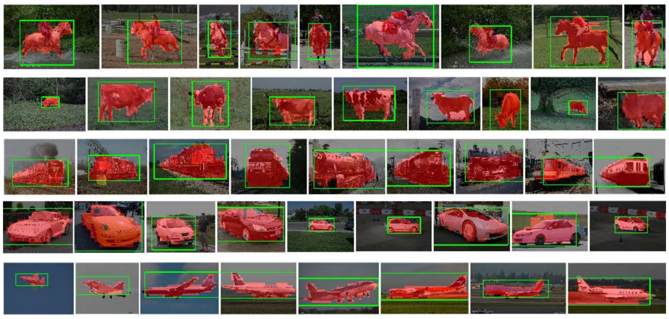

Fig. 7. Detection examples for certain PASCAL VOC 2007 categories. The cells corresponding to the object foreground are grouped and used for detection.

Table 3. AP for best performing categories of [1] in PASCAL VOC 2007

	horse	cow	cat	train	plane	car	mbike	bus	tv	bicycle	sofa	person
Our approach	**57.8**	25.3	23.9	**47.8**	31.9	**59.8**	**49.8**	**51.6**	41.9	**59.8**	**33.7**	**41.9**
Felz. etal. [1]	56.8	25.2	19.3	45.1	28.9	57.9	48.7	49.6	41.6	59.5	33.6	**41.9**
best2007 [26]	37.5	14.0	24.0	33.4	26.2	43.2	37.5	39.3	28.9	40.9	14.7	22.1
UCI [27]	45.0	17.7	12.4	34.2	28.8	48.7	39.4	38.7	35.4	56.2	20.1	35.5
LHS [13]	50.4	19.3	21.3	36.8	29.4	51.3	38.4	44.0	39.3	55.8	25.1	36.6
C2F [28]	52.0	22.0	14.6	35.3	27.7	47.3	42.0	44.2	31.1	54.0	18.8	26.8
SMC [29]	51.0	23.0	16.0	41.0	26.0	50.0	45.0	47.0	38.0	56.0	29.0	37.0
HStruct [30]	48.5	18.3	15.2	34.1	31.7	48.0	38.9	41.3	39.8	56.3	18.8	35.8
LatentCRF [31]	49.1	18.5	14.5	34.3	31.9	49.3	41.9	49.8	41.3	57.0	23.3	35.7
MKL [24]	51.2	**33.0**	**30.0**	45.3	**37.6**	50.6	45.5	50.7	**48.5**	47.8	28.5	23.3

a single, standard feature type, multi-feature approaches (e.g. [12, 24, 25]) are complementary and should enable further performance improvements.

5 Conclusion

We have presented a model that explicitly represents object shape and segregates it from the background, however, *without* requiring segmented training samples. The basis of this method is to capture the overall object form by grouping foreground regions in a model-driven manner and representing it through a class-specific prototypical set of segments automatically learnt from unsegmented training data. By using exlcusively bounding-box annotated training data, our model improves pixel-wise detection results and at the same time it provides a richer object parametrization for detecting object instances.

Acknowledgements. This work was supported by the Excellence Initiative of the German Federal Government, DFG project number ZUK 49/1.

References

1. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI (2010)
2. Levin, A., Weiss, Y.: Learning to combine bottom-up and top-down segmentation. IJCV 81(1), 105–118 (2009)
3. Gao, T., Packer, B., Koller, D.: A segmentation-aware object detection model with occlusion handling. In: CVPR, pp. 1361–1368 (2011)
4. Marszalek, M., Schmidt, C.: Accurate object recognition with shape masks. IJCV (97), 191–209 (2011)
5. Vijayanarasimhan, S., Grauman, K.: Efficient region search for object detection. In: CVPR (2011)
6. Malisiewicz, T., Efros, A.: Improving spacial support for objects via multiple segmentations. In: BMVC (2007)
7. Todorovic, S., Ahuja, N.: Learning subcategory relevances for category recognition. In: CVPR (2008)

8. Wang, X., Han, T., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: ICCV (2009)
9. Chen, Y., Zhu, L(L.), Yuille, A.: Active Mask Hierarchies for Object Detection. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 43–56. Springer, Heidelberg (2010)
10. Carreira, J., Li, F., Sminchisescu, C.: Object Recognition by Sequential Figure-Ground Ranking. IJCV (November 2011)
11. Gu, C., Lim, J., Arbeláez, J., Malik, J.: Recognition using regions. In: ICCV (2009)
12. Van de Sande, K., Uijlings, J., Gevers, T., Smeulders, A.: Segmentation as selective search for object recognition. In: ICCV (2011)
13. Zhu, L., Chen, Y., Yuille, A.L., Freeman, W.: Latent hierarchical structural learning for object detection. In: CVPR, pp. 1062–1069 (2010)
14. Ommer, B., Malik, J.: Multi-scale object detection by clustering lines. In: ICCV (2009)
15. Carreira, J., Scminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. In: CVPR (2010)
16. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
17. Andrews, S., Tscochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: NIPS, vol. 15 (2003)
18. Deselaers, T., Ferrari, V.: A conditional random field for multiple-instance learning. In: ICML (2010)
19. Ferrari, V., Jurie, F., Schmid, C.: Accurate object detection with deformable shape models learnt from images. In: CVPR (2007)
20. Toshev, A., Taskar, B., Daniilidis, K.: Object detection via boundary structure segmentation. In: CVPR (2010)
21. Yarlagadda, P., Monroy, A., Ommer, B.: Voting by Grouping Dependent Parts. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 197–210. Springer, Heidelberg (2010)
22. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR (2009)
23. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. PAMI 30(1), 36–51 (2008)
24. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)
25. Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: ICCV (2009)
26. Mark, E., Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes challenge 2007 (voc 2007). Results (2007)
27. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for mulit-class object layout. In: ICCV, pp. 229–236 (2009)
28. Pedersoli, M., Vedaldi, A., Gonzalez, J.: A coarse-to-fine approach for fast deformable object detection. In: CVPR (2011)
29. Razavi, N., Gall, J., van Gool, L.: Scalable mulit-class object detection. In: CVPR (2011)
30. Schnitzpan, P., Fritz, M., Roth, S., Schiele, B.: Discriminative structure learning of hierarchical representations for object detection. In: CVPR, pp. 2238–2245 (2009)
31. Schnitzspan, P., Roth, S., Schiele, B.: Automatic discovery of meaningful object parts with latent crfs. In: CVPR, pp. 121–128 (2010)