Bottom-up Segmentation for Top-down Detection

Anonymous CVPR submission

Paper ID 1700

Abstract

In this paper we are interested in how semantic segmentation can help object detection. Towards this goal, we propose a novel deformable part-based model which exploits region-based segmentation algorithms that compute candidate object regions by bottom-up clustering follow by ranking of those regions. Our approach allows every detection hypothesis to select a segment (including void), and scores each box in the image using both the traditional HOG filters as well as a set of novel segmentation features. Thus our model "blends" between the detector and the segmentation models. Since our features can be computed very efficiently given the segments, we maintain the same complexity as the original DPM [15]. We demonstrate the effectiveness of our approach in PASCAL VOC 2010, and show that when employing only a root filter our approach outperforms Dalal & Triggs detector [13] on **all** *classes, and achieves 13% higher average AP. When employing the parts, we outperform the original DPM [15] in* 18 *out of* 20 *classes and achieve an improvement of 8% AP.*

1. Introduction

Over the past few years, we have witnessed a push towards holistic approaches that try to solve multiple recognition tasks jointly [28, 8, 20, 18, 30]. This is important as information from multiple sources should facilitate scene understanding as a whole. For example, knowing which objects are present in the scene should simplify segmentation and detection tasks. Similarly, if we can detect where an object is, segmentation should be easier as only figure-ground segmentation is necessary. Existing approaches typically take the output of a detector and refine the regions inside the boxes to produce image segmentations [22, 7, 1, 15]. An alternative approach is to use the candidate detections produced by state-of-the-art detectors as additional features for segmentation. This simple approach has proven very successful [8, 19] in standard benchmarks.

In contrast, in this paper we are interested in exploiting semantic segmentation in order to improve object detection. While bottom-up segmentation has been often believed to be inferior to top-down object detectors due to its frequent over- and under- segmentation, recent approaches [10, 1] have shown impressive results in difficult datasets such as PASCAL VOC challenge. Here, we take advantage of region-based segmentation approaches [9], which compute a set of candidate object regions by bottom-up clustering, and produce a segmentation by ranking those regions using class specific classifiers. Our goal is to make use of these candidate object segments to bias sliding window object detectors to agree with these regions. Unlike the aforementioned holistic approaches, we reason about all possible object bounding boxes (not just candidates) to not limit the expressiveness of our model.

Deformable part-based models (DPM) [15] and its variants [3, 31, 11], are arguably the leading technique to object detection [1]. However, so far, there has not been many attempts to incorporate segmentation into DPMs. In this paper we propose a novel deformable part-based model, which exploits region-based segmentation by allowing every detection hypothesis to select a segment (including void) from a small pool of segment candidates (6-7 segments on average per image). Towards this goal, we derive simple features, which can capture the essential information encoded in the segments. Our detector scores each box in the image using both the traditional HOG filters as well as the set of novel segmentation features. Thus our model "blends" between the detector and the segmentation models, by boosting object hypotheses on the segments, but recovering from making mistakes by exploiting a powerful appearance model. Importantly, as given the segments we can compute our features very efficiently, our approach has the same computational complexity as the original DPM [15].

We demonstrate the effectiveness of our approach in PASCAL VOC 2010, and show that when employing only a root filter our approach outperforms Dalal & Triggs detector [13] by 13% AP, and when employing parts, we outperform the original DPM [15] by 8%. We believe that these results will encourage new research on bottom-up segmentation as

[1] Poselets [6] can be shown to be very similar in spirit to DPMs

well as hybrid segmentation-detection approaches, as our paper clearly demonstrates the importance of segmentation for object detection.

In the remainder of the paper, we first review related work and then introduce our novel deformable part-based model, which we call **segDPM**. We then show our experimental evaluation and conclude with future work.

2. Related Work

Deformable part-based model [15] and its variants have been proven to be very successful in difficult object detection benchmarks such as PASCAL VOC challenge. Several approaches have tried to augment the level of supervision in these models. Azizpour et al. [3] use part annotations to help clustering different poses as well as to model occlusions. Hierarchical versions of these models have also been proposed [31], where each part is composed of a set of sub-parts. The relative rigidity of DPMs has been alleviated in [11] by leveraging a dictionary of shape masks. This allows a better treatment of variable object shape. Desai et al. [14] proposed a structure prediction approach to perform non-maxima suppression in DPMs which exploits pairwise relationships between multi-class candidates. The tree structure of DPMs allows for fast inference but can suffer from problems such as double counting observations. To mitigate this, [26] consider lateral connections between high resolution parts.

In the past few years, a wide variety of segmentation algorithms that employ object detectors as top-down cues have been proposed. This is typically done in the form of unary features for an MRF [19], or as candidate bounding boxes for holistic MRFs [30, 21]. Complex features based on shape masks were exploited in [30] to parse the scene holistically in terms of the objects present in the scene, their spatial location as well as semantic segmentation. In [25], heads of cats and dogs are detected with a DPM, and segmentation is performed using a GrabCut-type method. By combining top-down shape information from DPM parts and bottom-up color and boundary cues, [29] tackle segmentation and detection task simultaneously and provide shape and depth ordering for the detected objects. Dai et al. [12] exploit a DPM to find a rough location for the object of interest and refine the detected bounding box according to occlusion boundaries and color information.

DPMs provide object-specific cues, which can be exploited to learn object segmentations [4]. In [24], masks for detected objects are found by employing a group of segments corresponding to the foreground region. Other object detectors have been used in the literature to help segmenting object regions. For instance, while [5] finds segmentations for people by aligning the masks obtained for each Poselet [6], [23] integrates low level segmentation cues with Poselets in a soft manner.

There are a few attempts to use segments/regions to improve object detection. Gu et al. [17] apply hough transform for a set of regions to cast votes on the location of the object. More recently, [27] learn object shape model from a set of contours and use the learned model of contours for detection. In contrast, in this paper we proposed a novel deformable-part based model, which allows each detection hypothesis to select candidate segments. Simple features express the fact that the detections should agree with the segments. Importantly, these features can be computed very efficiently, and thus our approach has the same computational complexity as DPM [15].

3. Semantic Segmentation for Object Detection

In this paper we are interested in utilizing semantic segmentation to help object detection. In particular, we take advantage of region-based segmentation approaches, which compute candidate object regions by bottom-up clustering and rank those regions to estimate a score for each class. Towards this goal we frame detection as an inference problem, where each detection hypothesis can select a segment from a pool of candidates (those returned from the segmentation for that class as well as void). We derive simple features, which while being very efficient to compute can capture most information encoded in the segments. In the remainder of the section, we first discuss our novel DPM formulation, which we call segDPM. We then define our new segment-based features and discuss learning and inference in our model.

3.1. A Segmentation-Aware DPM

Following [15], let p_0 be a random variable encoding the location and scale of a bounding box in an image pyramid as well as the mixture component id. As shown in [15] a mixture model is necessary in order to cope with variability in appearance as well as the different aspect ratios of the training examples. Let $\{p_i\}_{i=1,\cdots,P}$ be a set of parts which encode bounding boxes at double the resolution of the root. Denote with h the index over the set of candidate segments returned by the segmentation algorithm. We frame the detection problem as inference in a Markov Random Field (MRF), where each root filter hypothesis can select a segment from a pool of candidates. We thus write the score of a configuration as

$$E(\mathbf{p},h) = \sum_{i=0}^{N} w_i^T \cdot \phi(x,p_i) + \sum_{i=1}^{N} w_{i,def}^T \cdot \phi(x,p_0,p_i) + \\ + w_{seg}^T \phi(x,h,p_0) \quad (1)$$

where $h \in \{0,1,\cdots,H(x)\}$, with $H(x)$ the total number of segments for this class in image x. Note that $h = 0$ implies that no segment is selected. We will use $S(h)$ to

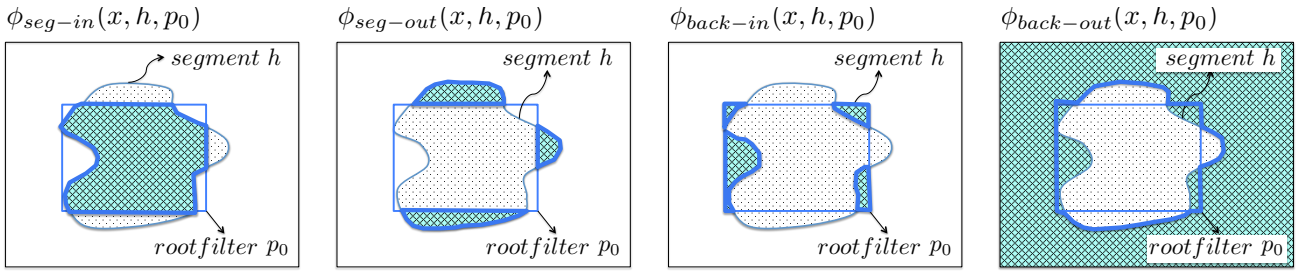


Figure 1. The box-segment features: ϕ_{seg-in} and $\phi_{seg-out}$, encourage the box to contain as many segment pixels as possible. This pair of features alone could result in box hypotheses that "overshoot" the segment. The purpose of the second pair of features, $\phi_{back-in}$ and $\phi_{back-out}$, is the opposite: it tries to minimize the number of background pixels inside the box and maximize its number outside. In synchrony these features would try to tightly place a box around the segment.

denote the segment that h indexes. As in [15], we employ a HOG pyramid to compute $\phi(x, p_0)$, and employ double the resolution to compute the part features $\phi(x, p_i)$. The features $\phi(x, h, p_0)$ link segmentation and detection. In this paper, we define features at the level of the root, but our formulation can be easily extended to include features at the part level.

3.2. Segmentation Features

Given a set of candidate segments, we would like to encode features linking segmentation and detection while remaining computationally efficient. We would also like to be robust to the fact that the candidate segments can possibly leak over the true segmentation or under-segment the image. Moreover, they could also be false positives. Towards this goal, we derive simple features which encourage that a selected segment agrees with the object detection hypothesis. Most of our features employ integral images which makes them particularly efficient, as this computation can be done in constant time. We now describe the features we used in more details.

Segment-In: Given a segment $S(h)$, our first feature counts the percentage of pixels in $S(h)$ that fall inside the bounding box defined by p_0. Thus

$$\phi_{seg-in}(x, h, p_0) = \frac{1}{|S(h)|} \sum_{p \in B(p_0)} \mathbb{1}\{p \in S(h)\}$$

where $|S(h)|$ is the size of the segment indexed by h, and $B(p_0)$ is the set of pixels contained in the bounding box defined by p_0. This feature encourages the bounding box to be positioned inside the segment.

Segment-Out: Our second feature counts the percentage of segment pixels that are outside the bounding box,

$$\phi_{seg-out}(x, h, p_0) = \frac{1}{|S(h)|} \sum_{p \notin B(p_0)} \mathbb{1}\{p \in S(h)\}$$

This feature will try to discourage boxes that do not contain all segment pixels.

Background-In: We additionally compute a feature counting the amount of background inside the bounding box as follows

$$\phi_{back-in}(x, h, p_0) = \frac{1}{N - |S(h)|} \sum_{p \in B(p_0)} \mathbb{1}\{p \notin S(h)\}$$

with N the size of the image. This feature captures the statistics of how often the segments leak outside the true bounding box vs how often they are too small.

Background-Out: This feature counts the amount of background outside the bounding box

$$\phi_{back-in}(x, h, p_0) = \frac{1}{N - |S(h)|} \sum_{p \notin B(p_0)} \mathbb{1}\{p \notin S(h)\}$$

It tries to discourage bounding boxes that are too big and do not tightly fit the segments.

Overlap: This feature penalizes bounding boxes which do not overlap well with the segment. In particular, it computes the intersection over union between the candidate bounding box defined by p_0 and the tighter bounding box around the segment $S(h)$. It is defined as follows

$$\phi_{overlap}(x, h, p_0) = \frac{p_0 \cap B(S(h))}{p_0 \cup B(S(h))} - \lambda$$

with $B(S(h))$ the tighter bounding box around $S(h)$ and λ a constant, which is the intersection over union level that defines a true positive. We employ in practice $\lambda = 0.7$.

Background bias: The value of all of the above features is 0 when $h = 0$. We incorporate an additional feature to learn the bias for the background segment ($h = 0$). This

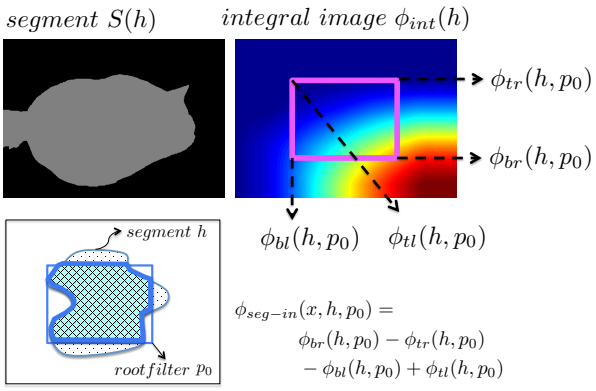


Figure 2. Integral geometry computation

puts the scores of the HOG filters and the segmentation potentials into a common referential. We thus simply define

$$\phi_{bias}(x, h, p_0) = \begin{cases} 1 & \text{if } h = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 1 depicts our features computed for a specific bounding box p_0 and segment $S(h)$. Note that the first two features, ϕ_{seg-in} and $\phi_{seg-out}$, encourage the box to contain as many segment pixels as possible. This pair of features alone could result in box hypotheses that "overshoot" the segment. The purpose of the second pair of features, $\phi_{back-in}$ and $\phi_{back-out}$, is the opposite: it tries to minimize the number of background pixels inside the box and maximize its number outside. In synchrony these features would try to tightly place a box around the segment. The overlap feature has a similar purpose, but helps us better tune the model to the VOC IOU evaluation setting.

3.3. Efficient Computation

Given the segments, all of our proposed features can be computed very efficiently. Note that the features have to be computed for each segment h, but this is not a problem as there are typically only a few segments per image ($6-7$ on average after clustering highly overlapping segments [10]).

We start our discussion with the first four features, which can be computed in constant time using a single integral image per segment. This is not only computationally efficient, but also memory efficient. Let $\phi_{int}(h)$ be the integral image for segment h, which counts at each point (u, v) the % of pixels that belong to this segment which are on the subimage defined by the domain $[0, u] \times [0, v]$. This is illustrated in Fig. 2. Given the integral image $\phi_{int}(h)$ for the h-segment, we compute the features as follows

$$\begin{aligned}
\phi_{seg-in}(x, h, p_0) &= \phi_{br}(h, p_0) - \phi_{tr}(h, p_0) \\
&\quad -\phi_{bl}(h, p_0) + \phi_{tl}(h, p_0) \\
\phi_{seg-out}(x, h, p_0) &= |S(h)| - \phi_{seg-in}(x, h, p_0) \\
\phi_{back-in}(x, h, p_0) &= |B(p_0)| - \phi_{seg-in}(x, h, p_0) \\
\phi_{back-out}(x, h, p_0) &= (N - |S(h)|) - \phi_{back-in}(x, h, p_0)
\end{aligned}$$

where as shown in Fig. 2, $(\phi_{tl}, \phi_{tr}, \phi_{bl}, \phi_{br})$ indexes the integral image of segment $S(h)$ at the four corners, i.e., top-left, top-right, bottom-left, bottom-right, of the bounding box defined by p_0.

The overlap feature between a hypothesis p_0 and a segment $S(h)$ can also be computed very efficiently. First, we compute the intersection as:

$$\begin{aligned}
&p_0 \cap B(S(h)) = \\
&\max\left[0, (\min(x_{0,right}, Bx_{right}) - \max(x_{0,left}, Bx_{left})\right] \cdot \\
&\max\left[0, (\min(y_{0,bottom}, By_{bottom}) - \max(y_{0,top}, By_{top})\right]
\end{aligned}$$

Note that the overlap will be non-zero only when each of the terms is larger than 0. Given that the segment bounding box $B(h)$ is fixed and the width and height of p_0 at a particular level of the pyramid are fixed as well, we can quickly compute the bounds of where in the image the feature needs to be computed (i.e., when the feature is different than 0). The denominator, $p_0 \cup B(S(h))$, can then be simply computed as the sum of the box areas minus the overlap.

3.4. Inference

Inference in our model can be done by solving the following optimization problem

$$\max_{p_0}\Big(\sum_{i=0}^{N} w_i^T \cdot \phi(x, p_i) + \sum_{i=1}^{N} \max_{p_i}(w_{i,def}^T \cdot \phi(x, p_0, p_i)) + \\ + \max_{h}(w_{seg}^T \cdot \phi(x, h, p_0))\Big)$$

Note that this can be done efficiently using dynamic programming as the structure of the graphical model forms a tree. The algorithm works as follows: First, we compute $\max_h w_{seg}^T \phi(x, h, p_0)$ as well as $\max_{p_i}(w_{i,def}^T \cdot \phi(x, p_0, p_i))$ for each root filter hypothesis p_0. We then compute the score as the sum of the HOG and segment score for each mixture component at the root level. Finally, we compute the maximum over the mixture components to get the score of an object hypothesis.

3.5. Learning

We learn a different weight for each feature using a latent structured-SVM [16]. Allowing different weights for the different segmentation features is important in order to

learn how likely is for each class to have segments that undershoot or overshoot the detection bounding box. We employ as loss the intersection over the union of the root filters.

As in DPM [15], we initialize the model by first training only the root filters, followed by training a root mixture model. Finally we add the parts and perform several additional iterations of stochastic gradient descent [15].

Note that we expect the weights of $\phi_{seg-in}(x, h, p_0)$, $\phi_{back-in}(x, h, p_0)$ and $\phi_{overlap}(x, h, p_0)$ to be positive, as we would like to maximize the overlap, the amount of foreground inside the bounding box and background outside the bounding box. Similarly, the weights of $\phi_{seg-out}(x, h, p_0)$ and $\phi_{back-in}(x, h, p_0)$ are expected to be negative as we would like to minimize the amount of background inside the hypothesis bounding box as well as the amount of foreground segment outside. In practice, as the object's shape can be far from rectangular, and the segments are noise, the sign of the weights can vary to best capture the statistics of the data.

4. Experimental Evaluation

We evaluate our detection performance on `val` subset of PASCAL VOC 2010 detection dataset. We select this scenario due to the availability of the CPMC segments along with their features on this split [9]. We train all methods, including the baselines on the `train` subset. We use the standard PASCAL criterion for detection (50% intersection over union overlap) and report average precision (AP) as the measure of evaluation.

We use as baselines the Dalal&Triggs detector [13] (which for fairness we compare to our detector when only using the root filter), the DPM [15], as well as CPMC [9] when used as a detector. To compute the latter, we find all the connected components in the final segmentation output of CPMC [9], and place the tightest bounding box around each component. To compute the score of the box we utilize the CPMC ranking scores for the segments, and utilize the strategy which has been shown to produce the best AP results, which consists on first partitioning the image into a set of super-pixels (we use UCMs [2]), and, for each super-pixel, we compute its score as the weighted average of the score of the segments that the super-pixel overlaps with. The score for the bounding box is then defined as the maximum value inside the box.

For many classes, CPMC works fairly well as a detector, as one can see from the performance in Table 1. One of the main failure cases are the images that contain multiple objects of the same class that are close to each other. The segmentation produces in these cases typically a single segment that spans several instances. This counts as a false positive under the strict VOC IOU measure. Our model has the advantage to deal with such cases as it scores an appearance model as well as the segment features, encoding the slack with respect to the segments.

We evaluate our model in two scenarios. In the first case, we do not consider the deformable parts (equivalent to the Dalal&Triggs [13] model) and learn only a fixed template for the root model. Towards this goal, we employ as features HOG as well as our novel segmentation features, defined in Section 3.2. The HOG cues are reliable for some categories such as bicycle or aeroplane, but less reliable for highly deformable categories such as cat. The learning method determines which features are more suitable for a particular class and find a reasonable trade-off between HOG, which mainly captures the edges of the object, and the segmentation features, which capture the object shapes. As shown, in Table 1, we significantly outperform the Dalal & Triggs detector (by 12.7%) as well as the CPMC baseline (by 10%). In particular, this is consistently the case, as we outperform the detector of Dalal & Triggs in **all** 20 classes and CPMC in 19 classes. With parts, see Table 2, we outperform both in **all** the classes. The precision-recall curves comparing our detector to the baselines are shown in Fig. 3.

Because of high variations in viewpoint and object poses, having a fixed template might hurt the performance. This is particularly the case of articulated objects. In the second scenario, similar to DPM, we add latent parts to the model to better capture class variability. We compare our approach to the DPM model in . One can see that our model achieves a significant boost of 7.8% AP over the DPM, which is a well established and difficult baseline to beat. Importantly, we outperform DPM in 18 out of 20 classes. The only class that is noticeably lower than DPM is the person class. However, the gap in performance is likely due to the fact that we trained the person model for our approach on only half of the positive images, while the DPM was trained on the full 7000 images. Note that our results well demonstrate the effectiveness of using blended detection and segmentation models for object detection.

Finally, Fig. 4 depicts examples illustrating the performance of our approach. Note that our approach is able to both retrieve detections where there is no segment as well as position the bounding box correctly where there is segment evidence. The last example represent a failure mode, where dog gets confuse with cat. This is not surprising as they are both articulated and have similar shape and appearance.

5. Conclusion

In this paper, we have proposed a novel deformable part-based model, which exploits region-based segmentation by allowing every detection hypothesis to select a segment (including void) from a pool of segment candidates. We derive simple yet very efficient features, which can capture the essential information encoded in the segments. Our detector scores each box in the image using both the HOG filters as

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	Avg.
Dalal [13]	29.1	36.9	2.9	3.4	15.6	47.1	27.1	11.4	9.8	5.8	6.0	5.0	24.8	28.4	27.5	2.2	18.4	9.2	27.4	23.2	18.1
CPMC (wo small) [9]	49.9	15.5	18.5	**14.7**	7.4	35.0	19.9	41.4	3.9	16.2	8.5	24.4	26.0	32.1	18.9	5.7	15.3	14.1	29.8	18.7	20.8
segDPM (wo parts)	**53.6**	**41.9**	**24.6**	12.2	**18.7**	**54.6**	**33.0**	**46.9**	**11.0**	**23.2**	**15.6**	**28.9**	**41.4**	**43.4**	**28.4**	**9.5**	**32.4**	**21.5**	**42.6**	**32.9**	**30.8**

Table 1. Average precision (AP) performance (in %) for our detector without parts, the Dalal & Triggs detector [13] and the CPMC-based detector [9].

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	Avg.
DPM [15]	46.3	**49.5**	4.8	6.4	22.6	53.5	38.7	24.8	14.2	10.5	10.9	12.9	36.4	38.7	**42.6**	3.6	26.9	22.7	34.2	31.2	26.6
segDPM (w parts)	**54.6**	49.4	**24.7**	**17.8**	**30.6**	**55.1**	**42.0**	**47.6**	**14.0**	**25.3**	**18.1**	**30.2**	**41.8**	**47.0**	35.7	**11.6**	**35.0**	**23.0**	**45.4**	**40.3**	**34.4**

Table 2. Average precision (AP) performance (in %) for our detector with parts and the DPM [15].

in original DPM, as well as a set of novel segmentation features. This way, our model "blends" between the detector and the segmentation model, by boosting object hypotheses on the segments, while recovering from making mistakes by exploiting a powerful appearance model. We demonstrated the effectiveness of our approach in PASCAL VOC 2010, and show that when employing only a root filter our approach outperforms Dalal & Triggs detector [13] by 13% AP and when employing parts, we outperform the original DPM [15] by 8%. We believe that this is just the beginning of a new and exciting direction. We expect a new generation of object detectors which are able to exploit top-down and semantic segmentation yet to come.

References

[1] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, and L. B. andJitendra Malik. Finding animals: Semantic segmentation using regions and parts. In *CVPR*, 2012. 1

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. In *PAMI*, 2011. 5

[3] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012. 1, 2

[4] L. Bertelli, T. Yu, D. Vu, and B. Gokturk. Kernelized structural svm learning for supervised object segmentation. In *CVPR*, 2011. 2

[5] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. 2

[6] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 1, 2

[7] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *CVPR*, 2011. 1

[8] G. Cardinal, X. Boix, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. In *CVPR*, 2010. 1

[9] J. Carreira, R. Caseiroa, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 1, 5, 6

[10] J. Carreira, F. Li, and C. Sminchisescu. Object Recognition by Sequential Figure-Ground Ranking. *International Journal of Computer Vision*, November 2011. 1, 4

[11] Y. Chen, L. Zhu, and A. Yuille. Active mask hierarchies for object detection. In *ECCV*, 2010. 1, 2

[12] Q. Dai and D. Hoiem. Learning to localize detected objects. In *CVPR*, 2012. 2

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005. 1, 5, 6

[14] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009. 2

[15] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010. 1, 2, 3, 5, 6

[16] R. Girshick, P. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NIPS*, 2009. 4

[17] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 2

[18] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *NIPS*, 2008. 1

[19] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 1, 2

[20] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010. 1

[21] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*, 2010. 2

[22] V. Lempitsky, P. Kohli, C. Rother, and B. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 1

[23] M. Maire, S. X. Yu, and P. Perona. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011. 2

[24] A. Monroy and B. Ommer. Beyond bounding-boxes: Learning object shape by model-driven grouping. In *ECCV*, 2012. 2

[25] O. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. The truth about cats and dogs. In *ICCV*, 2011. 2

[26] M. Pedersoli, A. Vedaldi, and J. Gonzlez. A coarse-to-fine approach for fast deformable object detection. In *CVPR*, 2011. 2

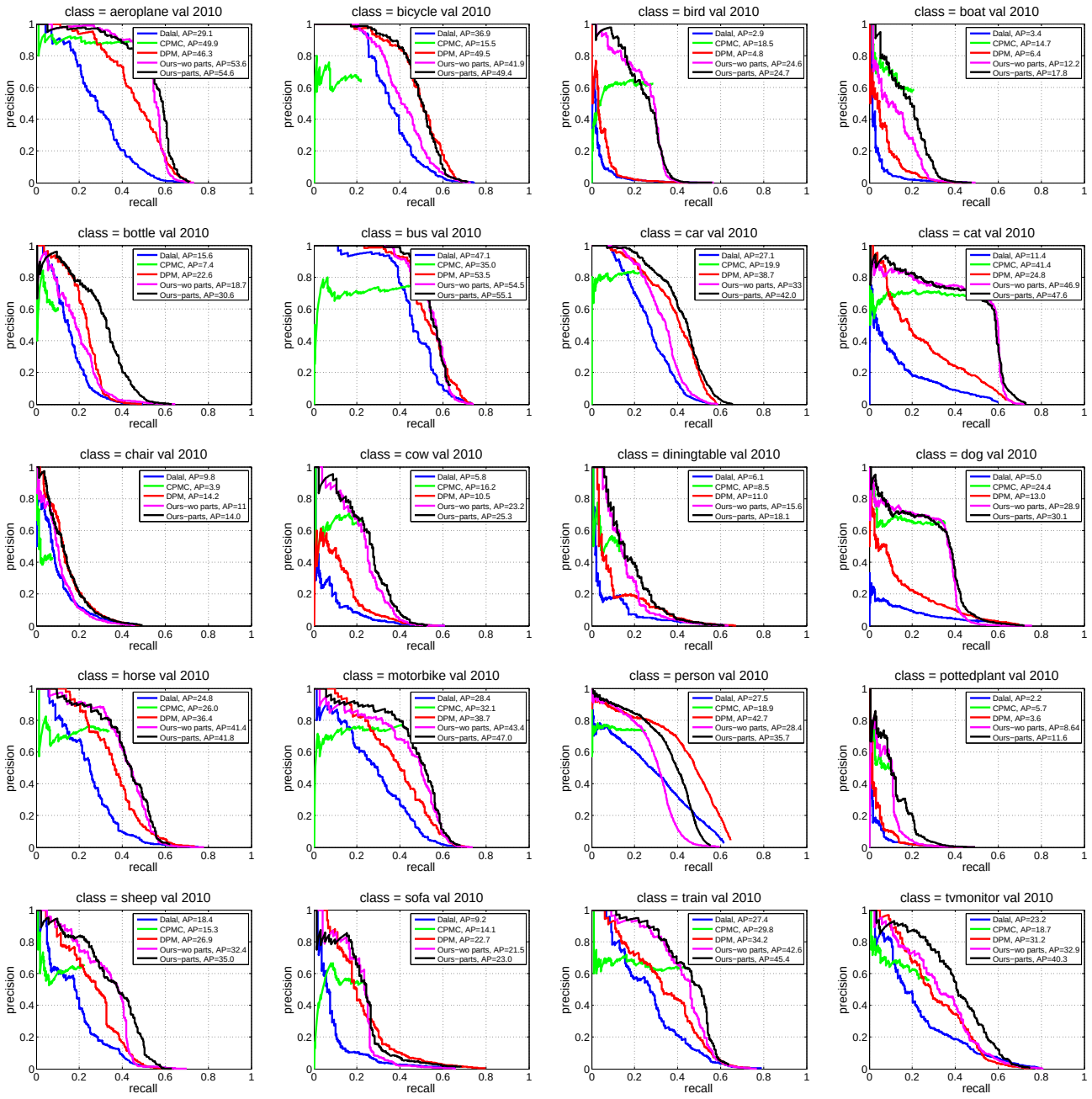


Figure 3. Precision-recall curves for all methods on the validation set of PASCAL VOC 2010. Note that our approach significantly outperforms all baselines.

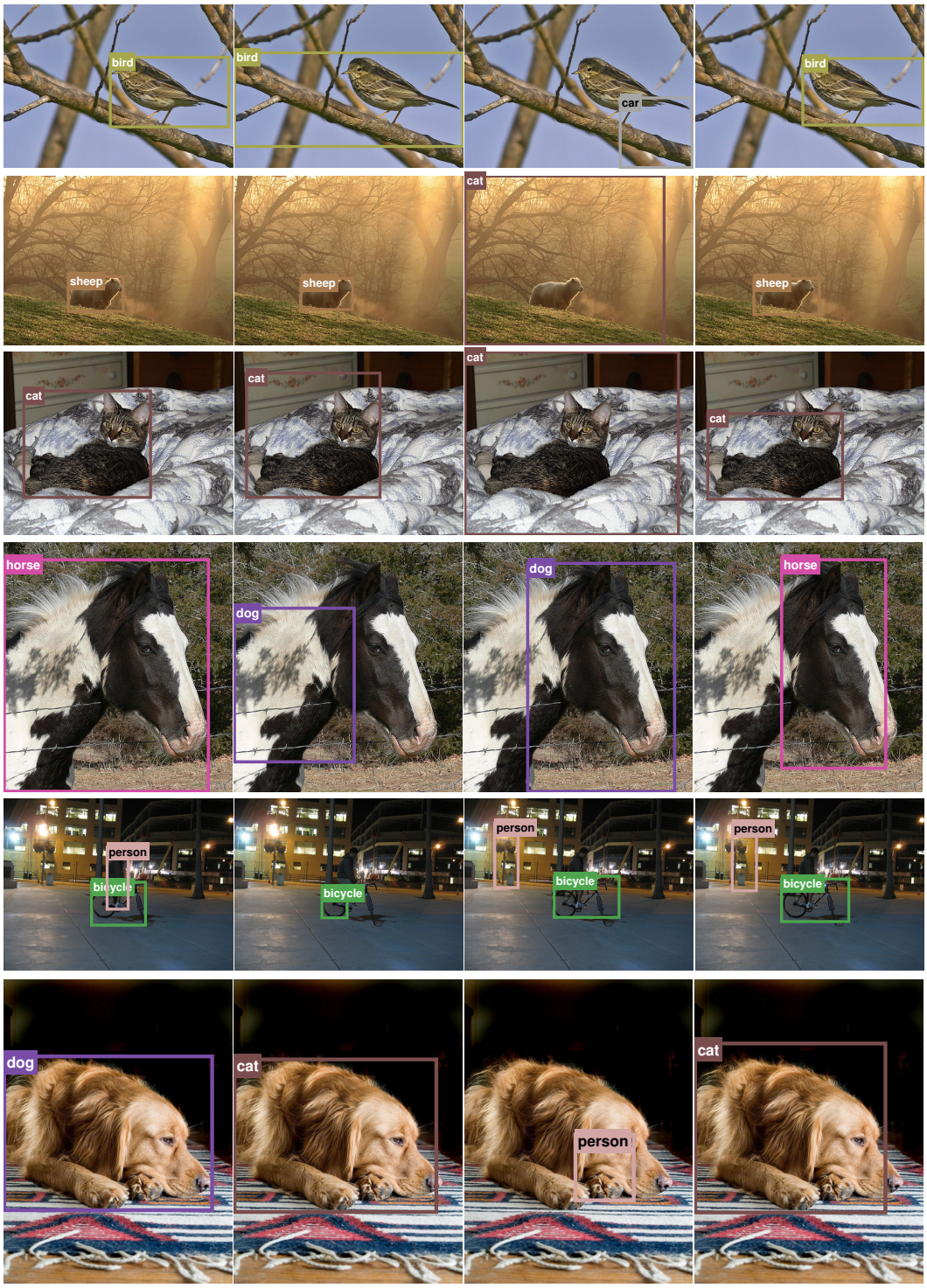
[27] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. In *CVPR*, 2010. 2

[28] E. Sudderth, A. Torralba, W. T. Freeman, and A. Wilsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, 2005. 1

[29] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object models for image segmentation. *PAMI*, 2011. 2

[30] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 1, 2

[31] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 1, 2



(a) GT (b) CPMC (c) DPM (d) segDPM

Figure 4. Example detections on PASCAL VOC 2010 dataset. For each image we show top K detections, where K is the number of GT objects in the image. The last example represent a failure mode, where dog gets confuse with cat.