Detecting Objects using Unsupervised Parts-based Attributes*

Santosh K. Divvala[1], Larry Zitnick[2], Ashish Kapoor[2] , Simon Baker[2]

[1]Carnegie Mellon University.
santosh@cs.cmu.edu

[2]Microsoft Research.
{larryz, ashishk, simonb}@microsoft.com

Abstract

This paper presents a new approach to parts-based object detection. Objects are described using a spatial model based on its constituent parts. Unlike most existing methods, parts are discovered in an unsupervised manner from training images with only object bounding boxes provided. The association between parts is modeled using boosted decision trees that allows arbitrary object-part configurations to be maintained. Experimental results on the challenging VOC 2007 dataset validate our approach.

1. Introduction

In this paper, the problem of detecting and localizing objects of a generic category, such as person, in Flickr-like images is considered. Such images have objects that are unaligned in arbitrary poses with severe occlusion and varying size. To deal with such challenges, recent research has focused on devising parts-based [6] and attributed-based [5, 9] methods. In [6], a deformable parts-based HOG model is developed that uses a coarse root model and multiple part models. The parts are learned in an unsupervised manner by modeling them as latent variables. A star-shaped constellation model is used to link parts together and reason about objects. In [5, 9], an attribute-based classifier is developed that describes objects in terms of basic semantic attributes. The attributes in their framework are gathered from human annotations. A tree-based hierarchical model is used where the first layer learns associations between low-level features (color, texture etc) and the semantic attributes, while the second layer associates the objects to the attributes. While the approach of [6] is attractive in its usage of unsupervised parts, it is limited due to its use of a fixed constellation model, which does not allow arbitrary object configurations to be learned. It is due to this reason, [6] explicitly maintains multiple component models each corresponding to different object poses (canonical pedestrian, sitting human etc). On the contrary, the approach of [5, 9] uses tree-based models that allow arbitrary object configurations to be modeled, but then relies on manually-gathered attributes that limits their approach to a small number of attributes.

In this paper, we propose a new parts-based object detection method that builds upon the success of the above methods while overcoming their limitations. We present an approach that uses part-based models where the parts are discovered using an unsupervised method and the associations between parts are learned using decision trees. While the latter allows great flexibility in the models learned, the former allows us to consider many attributes to be picked in a data-driven manner requiring no human annotations. We emphasize that, unlike the work of [1], we do not use any supervision in gathering the parts. As remarked in [5] "Observations of biological systems suggest that good representations can be learned automatically, leading to much research in unsupervised discovery of latent structure in images or objects. However, for a passive machine that cannot explore or manipulate objects, it is not known whether such structure can be discovered from images without supervision." We attempt to empirically analyze this aspect in this work.

Overview Similar to previous research, our approach uses part-based spatial models to describe object windows that allows to deal with arbitrary pose variations and alignment problems (See figure. 1). Given a set of training images with object bounding boxes, we describe a sampling based technique to discover object parts. The parts correspond to multiple levels of

*work done at Microsoft Research

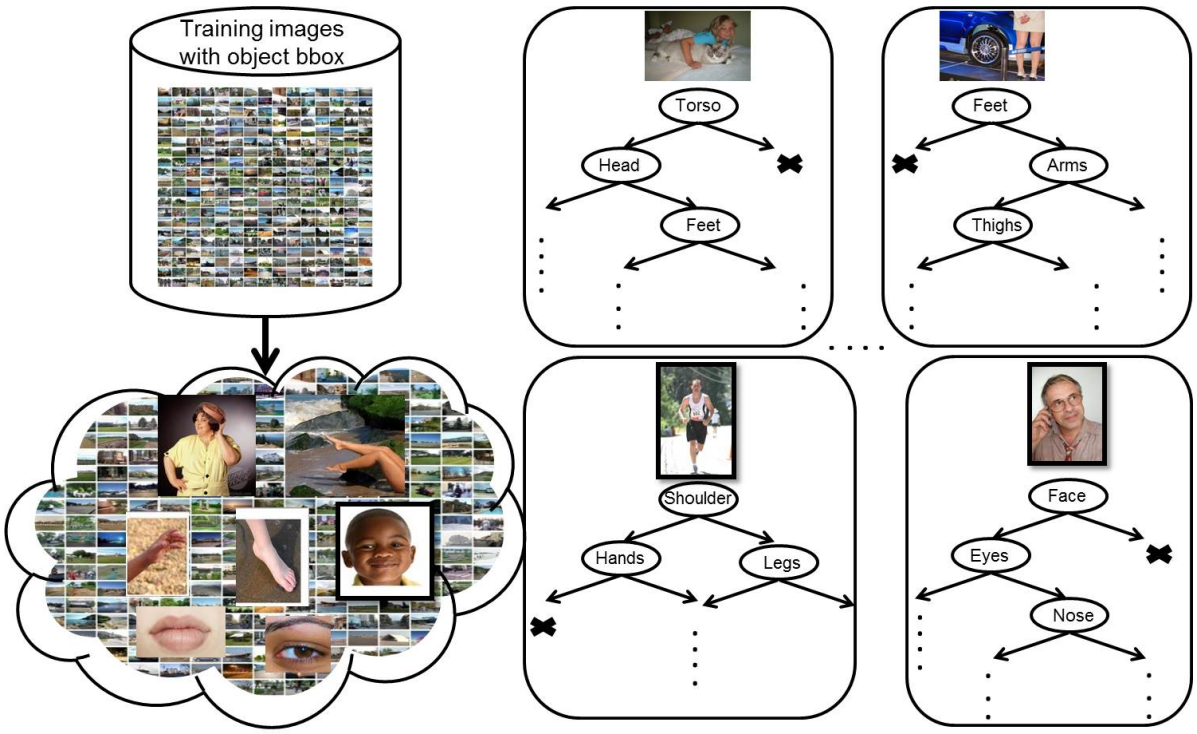

Figure 1. Our approach discovers parts in an unsupervised manner and uses decision-tree based models to model arbitrary object configurations.

object part hierarchy. At the highest level, there are high-resolution object parts such as human torso or lower body, and at the next level there exist mid-resolution parts such as human hand, foot or face and finally at the lowest level, the parts could correspond to human eyes, lips etc (section 2). Given these parts, we learn the spatial model for each that characterizes its location in a canonical space (section 3). This model is used to represent object windows within an input image. Using this representation, we train separate boosted decision tree classifiers to learn the presence or absence of an object at each spatial resolution (section 4). Absolute spatial models often could be noisy leading to poor discriminatory features. In order to improve the discriminability of the features, we explore the use of relative part attributes in section 5. Finally in order to prune redundant parts and improve the part templates over time, we present a feedback strategy in section 6. Our analysis is performed on the the challenging Pascal VOC dataset (section 7).

2. Discovering Parts

The first step of our approach is to obtain representative object parts in an unsupervised manner given images with only object bounding box annotations. This is done by first generating windows, in a sliding-window manner, within each object bounding box at multiple scales . These windows are represented using histogram-of-gradient (HOG) features. Finally representative windows are sampled from each scale to obtain the final part templates.

HOG feature representation The underlying building blocks for our models are the Histogram of Oriented Gradient (HOG) features. We follow the construction in [3] with updates suggested in [6] to define a dense representation of an image at a particular resolution. The image is first divided into $k \times k$ (k=8) non-overlapping pixel regions, or cells. For each cell we accumulate a 1D histogram of gradient orientations over pixels in that cell. These histograms capture local shape properties but are also somewhat invariant to small deformations. The gradient at each pixel is discretized into one of eighteen orientation bins, and each pixel "votes" for the orientation of its gradient, with a strength that depends on the gradient magnitude. For color images, we compute the gradient of each color channel and pick the channel with highest gradient magnitude at each pixel. Finally, the histogram of each cell is normalized with respect to the gradient energy in a neighborhood around it. We look at the four 2×2 blocks of cells that contain a particular cell and normalize the histogram of the given cell with respect to the total energy in each of these blocks. This leads to a vector of length 9×4 representing

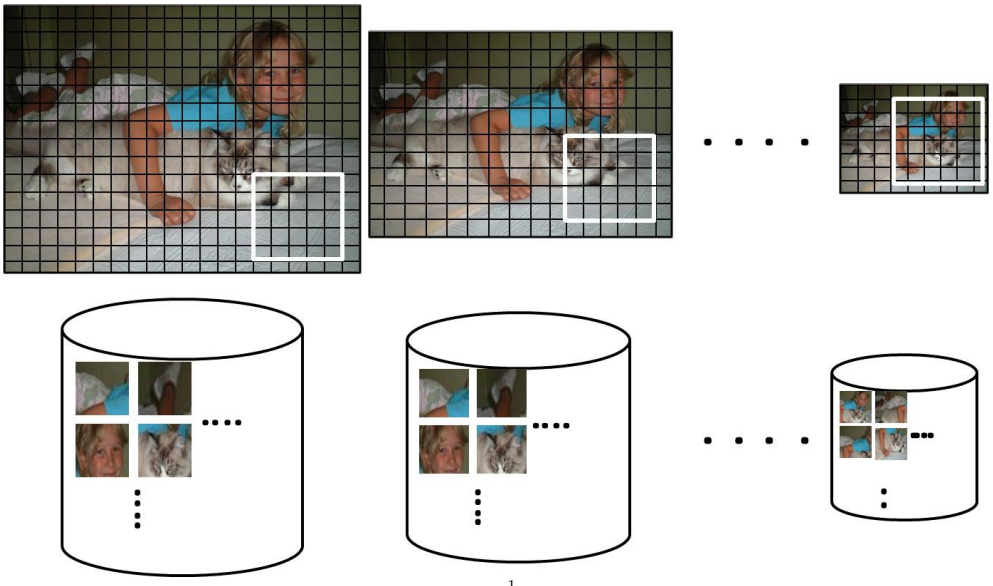

Figure 2. Image pyramid is built by scaling image using a factor of $2^{\frac{1}{6}}$. Scanning windows at each scale are considered in separate bags. Windows are sampled from each bag to obtain representative parts.

the local gradient information inside a cell. Rather than using the 36 dimensional vector directly, we project it onto a lower 32 dimensional space as described in [6].

Scanning windows are considered at multiple scales by scaling the image by a factor of $2^{\frac{1}{6}}$ and building the image pyramid (see figure 2). Features at the top of this pyramid capture coarse gradients histogrammed over fairly large areas of the input image while features at the bottom of the pyramid capture finer gradients histogrammed over small areas. Each window corresponds to a rectangular template at different levels of the HOG pyramid. In our experiments, we use a single window size of $w = 6$ and $h = 6$ which leads to a $w \times h \times 32$ dimensional feature vector. Further we noticed that there is more congruence in part matches when the levels of the pyramid for each object bounding box are explicitly restricted to a set of predefined image area sizes. Specifically we have considered part sizes corresponding to $[5\ 6\ 8\ 10\ 13\ 16\ 20\ 25\ 32\ 40\ 50]\%$ of the image area. We resize a given image bounding box to accommodate the largest part size possible by using the expression $t = \frac{sqrt(w \times h) * k}{sqrt(W \times H \times 0.05)}$. We ensure that every bounding box has a minimum of 3 scales, specifically for very low resolution object instances.

HOG features of the scanning window at each level are considered in separate bags. From each bag, we sample instances to be considered as the representative parts. We explore two strategies for the sampling. The first naive strategy is to randomly sample instances from each bag, while the second is to cluster all the windows using a k-means algorithm and pick the cluster centers. While clustering is computationally very expensive, the former is easy and feasible. In our experiments, we observed the former to be as good as the clustering method and thus have used the random sampling strategy to avoid the huge computational expense. The total number of parts considered is about 1000 to 5000 in our experiments with the number of points sampled from each bin being directly proportional on the bin size. Some of the parts sampled are displayed in figure 3. It must be observed that the discovered parts at various resolutions often correspond to different semantic object parts. For example, the high resolution parts correspond to torso, lower body, the mid resolution parts correspond to face, shoulder etc and the low resolution parts correspond to eyes, jaw etc.

3. Describing Objects using Spatial Part Models

In order to describe scanning windows in a new input image, we use a part-based spatial model based representation. The spatial model is built by warping the max detections for each part onto a canonical space and then quantizing the space to yield a compact parametrization.[1]

Filtering image using part templates The first step in this process involves running the part templates over given bounding

[1] We explore a simple parametrization in this work. It is also possible to use a Hough-transform based approach [10].

Figure 3. Discovered parts at various resolutions often correspond to different semantic object parts. For example, the high resolution parts correspond to torso, lower body, the mid resolution parts correspond to face, shoulder etc and the low resolution parts correspond to eyes, jaw etc.

boxes and recording the location of the max detection. This is performed by scoring each $w \times h$ scanning window of a HOG pyramid using a χ^2 distance function and picking the window corresponding to the lowest score. Rather than scoring all the windows at all scales, we restrict the scoring function to only consider windows that belong to either the exact same scale or a scale above or below of the query part. In our experiments, we found this to lead to better matches as this scheme avoids matching high-resolution parts (e.g., a human torso) to a a very low-resolution part template (e.g., person eye or nose). Also in our experiments, we observed that restricting to only the max detection leads to the matches for high-resolution parts being all concentrated in the center of the image bounding box. This is because high-resolution part templates match to the windows on the image boundary with a high-score. Although this is not a problem when multiple detections are used, it is of a major concern when only the max detection is considered. In order to cope with this problem, the image is padded by reflective padding (rather than conventional zero padding) which alleviates the problem. Figure 4 shows some of the matches obtained for the different part templates. In many cases, the retrieved matches are impressive. Of course looking locally at HOG features could in many cases lead to semantically poor matches (for e.g., matching human eye to a bicycle wheel).

Heat Map Parametrization Given the top detections across all the training image object bounding boxes, the goal here is to build a parametric representation that characterizes the spatial location of the given part template. This is done by first warping the location of the top K=2000 max detections onto a canonical 100×100 space. The canonical space is next quantized into a 8×8 coordinate frame using a bi-linear splatting algorithm. Given a new image window, we filter it using the HOG template of the part, record its max detection location and then project it onto the estimated 8×8 quantized space to compute its location feature value. This process is repeated for each part, which leads to a part location-based feature vector. Some of the heat maps are displayed in figure 4.

4. Tree-based modeling

The PASCAL training data consists of a large set of images with bounding boxes around each instance of an object. We reduce the problem of learning a detector with this data to a binary classification problem. Let D = ($< x_1, y_1 >, \ldots, <$

Figure 4. (a) Top 25 nearest neighbor matches obtained for the a few part templates using local HOG feature matching. In many cases, the retrieved matches are impressive. (b) The estimated spatial model in terms of heat map representation for each part.

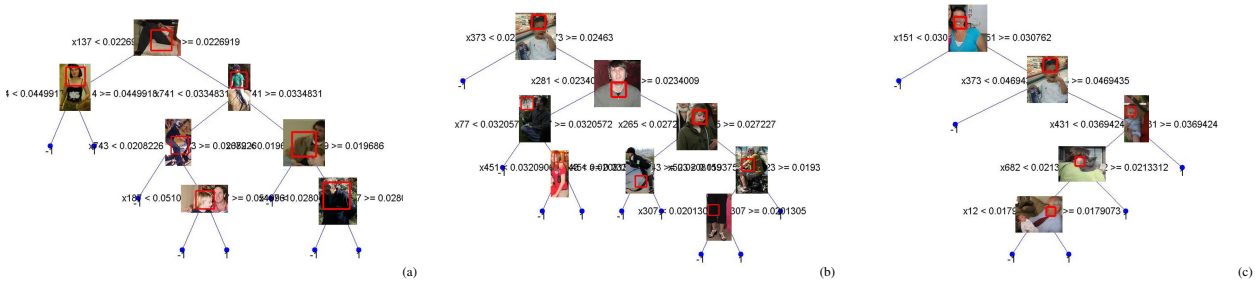

Figure 5. Some of the decision trees learned by our algorithm using part-based feature representation. Tree using (a) high-resolution parts (b) mid-resolution parts (c) low-resolution parts.

$x_n, y_n >$) be a set of labeled examples where $y_i \in -1, 1$ and x_i specifies the part-based features. We construct a positive example from each bounding box in the training set. The object windows are represented using the part-based spatial features. We use logistic regression version of Adaboost [2] with weak classifiers as 8-node decision trees [8] to learn the object models. Decision trees make good weak learners, since they not only provide automatic feature selection but allow interpretability of the learned models. Boosting allows us to learn multiple tree models each catering to a different "view" of the objects within the database. Thus we achieve great flexibility in automatically modeling the arbitrary object poses that are typical in real scene images unlike the approach of [6]. For negative datapoints, typically a cascade is employed starting from random negative windows and at the end of each iteration adding hard negatives. However due to computational issues, we resort to only one round of training using false positives obtained from a baseline detector [6]. Some of the decision trees learned are displayed in figure 5. It must be noticed that the learned trees are very interpretable and most often correspond to different object configurations. Further it must be noted that the parts need not always correspond to positive nodes.

5. Online Relative Parameter Representation

The spatial models computed in section 3 use absolute part locations. An interesting alternative would be to consider computing the models using the relative part features. For example, as illustrated in figure. 6, the detection of forehead in an image leads to a more peakier spatial model for the chest part.

To this end, we consider adding an additional set of P-dimensions to the feature vector for each window that would describe the relative spatial feature for each of the P-parts based on the parent part chosen in the decision tree. At the beginning of the training process, the relative features are all initialized to zero. The decision trees picks the most discriminative attribute amongst the absolute spatial part features and segregates the training data features into positive and negative sets. For the positive set, we update the latter half of its feature vector corresponding to the relative spatial part features based on the parent part picked by the tree. The decision tree now picks the most discriminative attribute from the augmented set of features to classify the data. This process leads to better spatial models being computed for each part and thus leads to improved discriminability of the features. However this comes at an expense as it is computationally intensive to recompute thousands of part spatial models at every node of the decision tree.

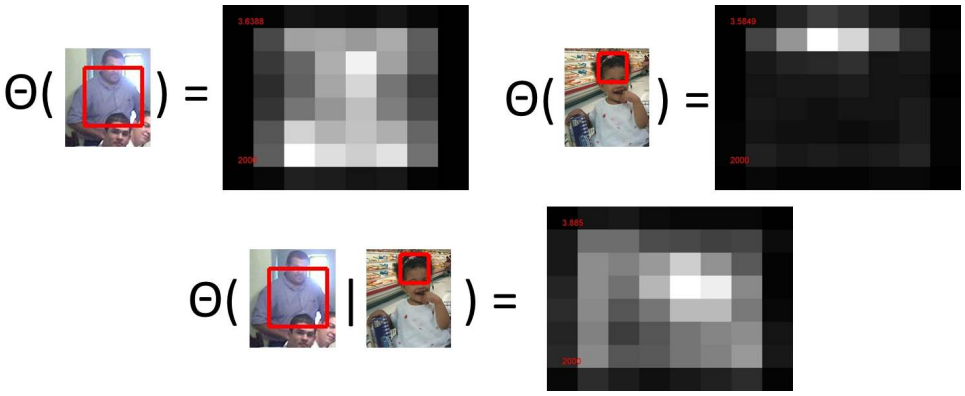

Figure 6. Relative Part attributes: The detection of forehead in an image leads to a more sharper spatial model for the chest detection

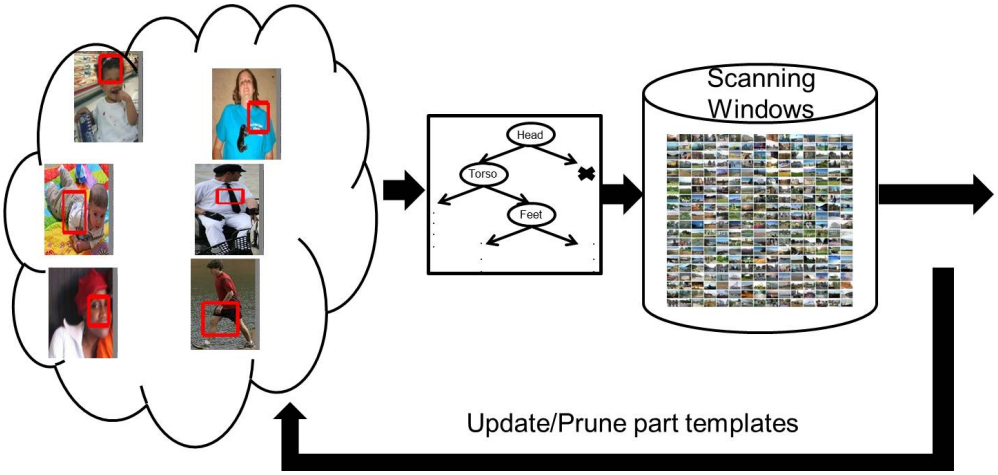

Figure 7. In order to prune redundant parts and also to enrich the part templates used by the decision tree, a feedback-step is considered.

6. Iterative Part Update

Many of the the initial part templates discovered in the part discovery step could be noisy. In order to prune redundant parts and also to enrich the part templates used by the decision tree, a feedback-step is considered as illustrated in figure. 7. The learned decision tree classifier is applied to all the positive bounding boxes in the training dataset and the object instances that pass the test at any given node are considered positive detections for the given part. The list of all instances across all nodes is recorded and are used to update the part HOG templates as in equation (1).

$$H_{new} = \alpha * H_{avg} + (1 - \alpha) * H_{original}, \tag{1}$$

$$where \;\; H_{avg} = \frac{\sum_{i=1}^{C} w_i * h_i}{\sum_{i=1}^{C} w_i}. \tag{2}$$

C is the number of positive instances recorded for each part and α is defined as $\frac{C}{C+2}$.

7. Experimental Results and Analysis

We evaluated our system using the PASCAL VOC 2007 comp3 challenge dataset and protocol. We refer to [4] for details, but emphasize that this challenge is widely acknowledged as difficult testbed for object detection. Each dataset contains several thousand images of real world scenes and 20 object categories. In our experiments, we specifically focus on the person class. The datasets specify ground-truth bounding boxes for several object classes, and a detection is considered correct when it overlaps more than 50% with a ground truth bounding box. One scores a system by the average precision

Figure 8. Some results from the PASCAL 2007 dataset. The first two rows show correct detections while the last row shows false positives. Our system is able to detect objects over a wide range of scales and poses. Our system can also detect partially occluded objects.

(AP) of its precision-recall curve across a test set. All our experiments have been run on the Microsoft High Performance Cluster cluster. The main computational intensive part is the running of 1000's of part templates over each sliding window. Table 1 displays the detection results for all the 20 VOC classes. Our approach achieves better results compared to the UoCTTI'07 result. In figure 8, we show some example detections for the 'person' class. Our approach detects objects not only over a wide range of scales and poses but also those that are partially occluded.

We conducted experiments varying the different parameters in our approach for the 'person' class. Our baseline method using 20 trees with 1000 parts obtains an A.P of 19.6%. Increasing the number of trees to 100 improves the result to 20.2%. Augmenting the baseline method with relative features improves the result to 20.1%. In contrast, updating the number of parts to 5000 drops the score to 17.0% and incorporating the feedback step also leads to a drop in score to 16.0%. This issue will be further investigated in our future work.

8. Conclusion

We introduced a new framework for representing objects using unsupervised parts-based attributes. We used this representation to build decision-tree based models to recognize objects in real world images. Experimental results on difficult benchmark data validates our approach.

Our system could be further enriched by using other types of features apart from HOG features to improve the part discovery as well as the nearest neighbor matching process. Further instead of using sliding windows, a segmentation based approach could be considered. Also a branch and bound scheme could be employed to speed up the algorithm.

References

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

[2] M. Collins, R. Schapire, and Y. Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3), 2002. 5

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 2

[4] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge results, 2009. http://pascallin.ecs.soton.ac.uk/challenges/VOC. 6

[5] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 1

Objects	Our Approach	UoCTTI'07 result	UoCTTI'09 result
aeroplane	25.6	20.6	28.9
bicycle	33.0	36.9	59.5
bird	6.8	9.3	10.0
boat	3.2	9.4	15.2
bottle	16.3	21.4	25.5
bus	47.7	23.2	49.6
car	37.9	34.6	57.9
cat	14.0	9.8	19.3
chair	0.9	12.8	22.4
cow	9.6	14.0	25.2
diningtable	17.0	0.2	23.3
dog	11.5	2.3	11.1
horse	23.3	18.2	56.8
motorbike	32.5	27.6	48.7
person	19.8	21.3	41.9
pottedplant	5.3	12.0	12.2
sheep	29.9	14.3	17.8
sofa	18.0	12.7	33.6
train	16.7	13.4	45.1
tvmonitor	32.1	28.9	41.6
Mean	20.1	17.1	32.3

Table 1. Detection Results on PASCAL VOC 2007 testset. The first column is the average precision (A.P.) obtained using our approach. The second column is UoCTTI detector result from VOC2007 challenge [11] and the third column is result using UoCTTI latest detector [7].

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

[7] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/. 8

[8] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2), 2000. 5

[9] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1

[10] S. Maji and J. Malik. Object detection using a max-margin hough transform [pdf] [ppt]. In *CVPR*, 2009. 3

[11] The pascal object recognition voc 2007 competition. Website, 2007. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/results/index.shtml. 8