A Segmentation-aware Object Detection Model with Occlusion Handling

Tianshi Gao[1] Benjamin Packer [2] Daphne Koller[2]
[1] Department of Electrical Engineering, Stanford University
[2] Department of Computer Science, Stanford University
{tianshig,bpacker,koller}@cs.stanford.edu

Abstract

The bounding box representation employed by many popular object detection models [3, 6] implicitly assumes all pixels inside the box belong to the object. This assumption makes this representation less robust to the object with occlusion [16]. In this paper, we augment the bounding box with a set of binary variables each of which corresponds to a cell indicating whether the pixels in the cell belong to the object. This segmentation-aware representation explicitly models and accounts for the supporting pixels for the object within the bounding box thus more robust to occlusion. We learn the model in a structured output framework, and develop a method that efficiently performs both inference and learning using this rich representation. The method is able to use segmentation reasoning to achieve improved detection results with richer output (cell level segmentation) on the Street Scenes and Pascal VOC 2007 datasets. Finally, we present a globally coherent object model using our rich representation to account for object-object occlusion resulting in a more coherent image understanding.

1. Introduction

Object detection and segmentation are two fundamental problems in computer vision. These two tasks are generally solved independently, each with its own representation. The most popular object detection methods use bounding box approaches [3, 6], which do not model which pixels actually support the object, but can achieve high detection performance. On the other hand, object segmentation methods [12, 21] usually adopt Markov Random Field (MRF) models, which are capable of assigning a label for each pixel, but are not efficient and effective enough for the detection task. Intuitively, jointly solving these two tasks can help each benefit from each other: a bounding box object detector can offer the segmentation model good location and shape priors [12, 22]; a segmentation model can provide the detector cleaner features, especially in the face of clutter [8]. In this paper, we focus on the latter case, *i.e.*, using segmentation reasoning to help the detection.

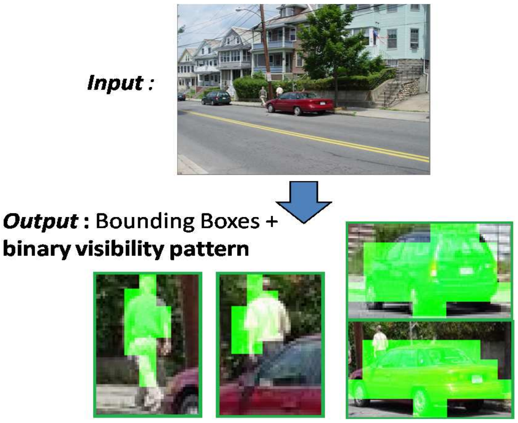
Input **:**
Output **: Bounding Boxes +**
binary visibility pattern

Figure 1: Demonstration of our representation and inference results. We introduce a binary variable for each cell in the bounding box indicating whether the pixels inside it belong to the object. The values of these variables need to be inferred when performing detection. Color shaded pixels indicate they belong to the object.

The importance of having a segmentation-aware model to explicitly reason about which pixels belong to the object in the bounding box when performing detection is reflected by occlusion. Occlusion is a common problem in real world images and videos and is a major obstacle for object detection. In the Caltech Pedestrian Dataset [16], for example, over 70% of pedestrians are occluded in at least one frame of a video sequence and 19% are occluded in all frames, where the occlusion was categorized as heavy in nearly half of these cases. Dollar et al. [16] show that detection performance for standard methods drops significantly even under partial occlusion, and drastically under heavy occlusion. An object detector that has the ability to learn and infer visibility patterns may therefore yield more robust results.

We propose a rich representation for object detection and segmentation. Specifically, we introduce binary variables for each cell in the bounding box indicating whether the pixels in the cell are from the object centered in the bounding box, as shown in Figure 1. These variables are observed during training but must be inferred at test time to provide a segmentation. To address the challenges of an exponentially large label space, we use a structured output framework, employing a structural SVM [10]. We also present an efficient method for inference — utilized during both training and testing — that uses graph cuts within the bounding box, combined with a global branch-and-bound procedure.

Furthermore, we provide a global coherent model aiming to do multi-object occlusion reasoning, allowing the inferred presence and visibility of one object to affect the likelihood that an object behind it is occluded. This results in a more coherent explanation of the scene, since the constraint that a pixel cannot be occupied by multiple objects is satisfied by our segmentation reasoning. We test our approach on the Street Scenes [1] and PASCAL VOC 2007 [5] datasets, and show significant improvements.

2. Background and Related Work

There have been several attempts to combine object detection and segmentation. The OBJCUT method of Kumar *et al*. [12] uses part detectors to get object-category specific priors for the segmenter. Winn *et al*. [21] use local and layout-sensitive pairwise potentials within a part-based conditional random field framework to segment an image into separate objects while handling arbitrary occlusion. However, the former method does not deal with occluded objects, and neither method has been shown to be efficient or have high detection performance on a variety of categories.

Recently, both Gould *et al*. [8] and Ladicky *et al*. [13] combined an object detector with an image segmenter. These methods use the object detector as a black box to propose bounding box candidates. While the methods do perform segmentation, the black box detector is fixed and does not reason about the segmentation within the bounding box, leaving it susceptible to errors due to clutter and occlusion. Yang *et al*. [22] represent ordered layers of object detections to estimate refined object appearance, similar to the order-based energy that we introduce in our global coherent model (Section 5). They use this ordering to improve only segmentation performance but not detection. Eichner *et al*. [4] also used depth ordering reasoning to handle occlusion for specific task of joint pose estimation.

There are also works whose detectors use segmentation inference directly to achieve more robust performance. Wang *et al*. [20] use a property of HOG features [3] on the human class to infer occluded pixels in a detection window. Since this approach is strongly tied to both the specific object class and features used, it does not generalize to other settings. Vedaldi and Zisserman [19] is the work most similar to ours. They propose using binary variables to indicate the visibility of the cells inside a detection window. However, they only model occlusion on image borders, and rather than inferring the values of those variables, they treat them as a deterministic function of the position of the bounding boxes. Specifically, when the bounding box is partially outside of the image, those variables corresponding to the cells that are outside of the image are set to be invisible. We generalize the visibility variables to allow them to represent arbitrary masks occurring anywhere in the image, and introduce terms that specifically inform these variables based on features from the image.

3. Model

Given an image $\mathbf{x}$ and an object category, our goal is to predict a bounding box indicating where the object of interest is as well as inferring the supporting pixels for the object (visibility pattern). We begin the description of our model by introducing the representation.

The first component of our representation corresponds to the bounding box and the view. A bounding box is specified by $\mathbf{p} = (p_x,\ p_y,\ p_s)$, where $(p_x,\ p_y) \in [1, W_s] \times [1, H_s]$ is the position of the box where W_s and H_s are the width and height of the image at the sth level of the pyramid. $p_s \in [1, S]$ specifies the scale, where S is the total number of levels of the image pyramid. We use $a \in [1, A]$ to index the view point (aspect) of the object instance, where A is the total number of view points in the model. For example, a horse can be labeled as being frontal view, left-facing view, right-facing view and rear view.

The object representation that we use is based on the HOG descriptor [3]. The image is decomposed into cells, and the descriptor of a cell is a histogram of the orientation of the image gradient inside the cell normalized by its different neighbors' gradient energy [6]. The HOG descriptor of a bounding box is the concatenation of cell descriptors within the box.

In addition to the parameters specifying the bounding box, we further define a set of binary variables, each of which corresponds to a cell in the bounding box, indicating whether the pixels of this cell belong to the object centered in the bounding box. Suppose that for a specific view a the HOG template size in cells is $w_a \times h_a$. We define a set of binary variables $\mathbf{v} = (v_1, ..., v_{w_a h_a})$ such that if $v_i = 1$ then the pixels in the i-th cell are visible; otherwise they're not. Therefore, the labeling of our model is defined by a vector $\mathbf{y} = (p_x,\ p_y,\ p_s,\ a,\ \mathbf{v})^T$.

Given a labeling $\mathbf{y}$ and the image $\mathbf{x}$, we have a joint feature vector $\Psi(\mathbf{x}, \mathbf{y})$ which is built from HOG descriptors [3] and is a mixture model consisting of multiple views. For a given view a and bounding box $\mathbf{y}\langle\mathbf{p}\rangle$, we extract the HOG descriptor for each cell in the bounding box, and stack these together to form a single feature vector $\phi_a(\mathbf{x}, \mathbf{y}\langle\mathbf{p}\rangle)$, where $\mathbf{y}\langle\cdot\rangle$ denotes the assignment to the component $\cdot$ in the label $\mathbf{y}$. Given these representations, we now define a baseline model similar to that of Vedaldi and Zisserman [19] aiming to model truncation at the image boundary, and then introduce our segmentation-aware model which is able to handle arbitrary visibility pattern within the bounding box.

3.1. Baseline model

In the baseline model, $\mathbf{v}$ is a deterministic function of the bounding box $\mathbf{p}$: given a bounding box specified by $\mathbf{p}$, $v_i = 0$ for those cells i which are outside of the image, and $v_j = 1$ for those cells which are inside the image. Therefore, we write $\mathbf{v}$ as $\mathbf{v}(\mathbf{y}\langle\mathbf{p}\rangle)$. Based on the value of $\mathbf{v}$, we zero out the feature components from those cells which are labeled as invisible. We denote this modified feature vector as $(\mathbf{v}(\mathbf{y}\langle\mathbf{p}\rangle) \otimes \mathbf{1}_{N_h}) \odot \phi_a(\mathbf{x}, \mathbf{y}\langle\mathbf{p}\rangle)$, where $\odot$ is the component-wise multiplication operator between two vectors, $\otimes$ is the Kroneker product, and N_h is the dimension of the HOG descriptor for a cell. Note that any feature representation that decomposes into cells can be used in our framework. Similar to Vedaldi and Zisserman [19], we use the number of invisible cells, $w_a h_a - \sum v_i(\mathbf{p})$, as the feature summarizing the invisible part, where w_a and h_a are the width and height for the HOG template of view a. To calibrate the scores from different views, we have a constant 1 appended to the feature vector serving as the bias for the view a. Putting these three parts together, we have $\Psi_a(\mathbf{x}, \mathbf{y}) = [(\mathbf{v}(\mathbf{y}\langle\mathbf{p}\rangle) \otimes \mathbf{1}_{N_h}) \odot \phi_a(\mathbf{x}, \mathbf{y}\langle\mathbf{p}\rangle); w_a h_a - \sum v_i(\mathbf{p}); 1]^T$. We define such a feature set for each possible view, and stack them to obtain our final feature vector.

3.2. The segmentation-aware object model

The baseline model allows for truncation of the bounding box by the image frame, but still assumes that all pixels of a bounding box inside the image should be associated with the candidate object. However, if a person is occluded by a car, only the pixels from the upper part of the bounding box are supporting evidence for the presence of the person and the pixels from the lower part may serve as noise and adversely affect the prediction. To address this issue, we modify the definition of our visibility variables $\mathbf{v}$ so that they are no longer determined by the placement of the bounding box within the image, but rather inferred as part of the algorithm. The assignments to $\mathbf{v}$ correspond to a particular occlusion pattern, and are observed in the training set, but must be inferred for test instances.

Similarly to the base model, the joint model is a mixture of multiple components, each of which corresponds to a view a. For each view, we have 5 components.

The first component of the feature vector corresponds to the HOG features but we arrange features so that features from the cells which are labeled as visible and those from the cells which are labeled as invisible lie in two different subspaces in the joint feature vector as follows:

$$\begin{bmatrix} (\mathbf{v} \otimes \mathbf{1}_{N_h}) \odot \phi_a(\mathbf{x}, \mathbf{y}\langle\mathbf{p}\rangle) \\ (\mathbf{1}_{w_a h_a} - \mathbf{v} \otimes \mathbf{1}_{N_h}) \odot \phi_a(\mathbf{x}, \mathbf{y}\langle\mathbf{p}\rangle) \end{bmatrix} \tag{1}$$

The first line of Equation 1 is computed by extracting the HOG features from the bounding box and zeroing out those components from cells which are labeled as invisible. The second line is computed by zeroing out those components from cells which are labeled as visible. We learn two different sets of weights for the visible and invisible parts to separate out the potential noise from pixels in the occluded parts. Note that a sensible alternative is to completely discard those components from the invisible parts. We tried this configuration in early experiments, but found that the performance degraded when the invisible parts are ignored, since features from both the background and occluding objects do provide some useful signal.

The second component corresponds to the prior for each cell to belong to the object. As shown in Dollar *et al*. [16], the distribution of occlusion over the cells in the bounding box is not uniform. For example, in street scenes, the lower part of the pedestrian is more likely to be occluded than the upper part. To allow the model to learn the visibility prior for each cell, we include $[\mathbf{1}_{w_a h_a} - \mathbf{v}]$ in our feature vector.

The third component corresponds to truncation by the image frame. As in the baseline model, we summarize the truncation due to the limited field of view by counting the number of cells $c(\mathbf{p})$ that lie outside of the image.

The fourth component encourages the smoothness of the visibility labeling. We employ different smoothness terms in the horizontal and vertical directions; using $\mathcal{E}_v$ and $\mathcal{E}_h$ to denote the vertical and horizontal edges, respectively, we define a pairwise feature:

$$\begin{bmatrix} \sum_{(i,j)\in\mathcal{E}_v} \mathbf{1}\{v_i = v_j\} \\ \sum_{(i,j)\in\mathcal{E}_h} \mathbf{1}\{v_i = v_j\} \end{bmatrix} \tag{2}$$

We sum over all pairs in a given orientation due to the sharing of the weights for each direction. If the weights for the pairwise term are positive, then the model encourages the smoothness and continuity of the visibility labeling.

The last component is a prior for the specific view which is represented as a constant 1 appended in the feature vector. Finally, we stack the features from different views to form a joint sparse feature vector, as was done in Section 3.1.

3.3. Structured training

For both the baseline model and the flexible occlusion model, our goal is to learn a discriminative function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that predicts the annotation $\mathbf{y} \in \mathcal{Y}$ given an image $\mathbf{x} \in \mathcal{X}$. We learn this mapping in the structured learning framework [18, 17, 10] as:

$$f(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x},\ \mathbf{y}; \mathbf{w}), \quad F(\mathbf{x},\ \mathbf{y}; \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}) \tag{3}$$

where $\Psi(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d$ is our joint feature map and $\mathbf{w} \in \mathbb{R}^d$ is the vector of model parameters. Given a set of training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, we learn the model parameters in the max-margin framework with the 1-slack formulation [10]:

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\xi \tag{4}$$
$$\text{s.t. } \frac{1}{N} w^T \sum_{i=1}^{N} (\Psi(\mathbf{x}_i,\ \mathbf{y}_i) - \Psi(\mathbf{x}_i,\ \hat{\mathbf{y}}_i)) \geq$$
$$\frac{1}{N} \sum_{i=1}^{N} \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) - \xi,\ \forall (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N) \in \mathcal{Y}^N$$

where $\Delta(\mathbf{y}, \mathbf{y}') \in [0, 1]$ is a loss function measuring the distance between two annotations. Intuitively, for a given image $\mathbf{x}_i$ we want the score of the ground truth annotation $F(\mathbf{x}_i,\ \mathbf{y}_i; \mathbf{w})$ to be higher than $F(\mathbf{x}_i,\ \mathbf{y}; \mathbf{w})$ for any other annotation $\mathbf{y}$ by the distance between $\mathbf{y}_i$ and $\mathbf{y}$, *i.e.* $\Delta(\mathbf{y}_i, \mathbf{y})$. An important subtlety that arises in our setting is that there may be more than one detection of the given class per image, and thus more than one right answer in Equation 3. Thus, when formulating our margin constraints in Equation 4, the loss $\Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ of a detection $\hat{\mathbf{y}}_i$ is defined according to the ground truth (of the right class) that has the largest overlap with $\hat{\mathbf{y}}_i$.

The standard evaluation criterion for object detection is the overlap between the candidate bounding box and the ground truth bounding box using the ratio of the intersection and union areas between them. To match the test criterion, we use the same measure as the loss [2]:

$$\Delta(\mathbf{y}_i, \mathbf{y}) = \begin{cases} 0 & \text{, if } \mathbf{y}_i = \mathbf{y} = \varnothing \\ 1 & \text{, if } \mathbf{y}_i = \varnothing,\ \mathbf{y} \neq \varnothing \\ & \text{ or } \mathbf{y}_i \neq \varnothing,\ \mathbf{y} = \varnothing \\ 1 - \frac{\text{area}(\mathbf{y}\langle p \rangle \cap \mathbf{y}_i\langle p \rangle)}{\text{area}(\mathbf{y}\langle p \rangle \cup \mathbf{y}_i\langle p \rangle)} & \text{, otherwise} \end{cases}$$

where the value $\varnothing$ means that there is no object of interest in the image (we define $\Psi(\mathbf{x}, \varnothing) = \mathbf{0}$) and area$(p \cap p')$ and area$(p \cup p')$ are the intersection and union areas between two bounding boxes p and p' respectively. In other words, if the ground truth and the proposed $\mathbf{y}$ both label the image as no object, then the loss is 0; if they disagree with the presence of the object, then the loss is 1; otherwise, the higher the overlap between two labelings, the smaller the loss.

4. Inference

We now describe how to perform efficient MAP inference on the joint model, that is, solve the optimization problem of Equation 3. The inference task is necessary also within the learning algorithm, since the key step of the cutting-plane algorithm [10] for solving Equation 4 is to compute the most violated constraint for each training sample $\mathbf{x}_k$ given the current model parameters, i.e.,

$$\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}}\ w^T \Psi(\mathbf{x}_k,\ \mathbf{y}) + \Delta(\mathbf{y}_k, \mathbf{y}) \tag{5}$$

It is convenient to view the label space in terms of three subspaces: $\mathbf{y} \in \{1, 2, \ldots, A\} \times \mathcal{P} \times \mathcal{V}$, where $\{1, 2, \ldots, A\}$ is the index set of different views, $\mathcal{P}$ is the set of all bounding boxes we search over, and $\mathcal{V} = \{0, 1\}^{wh}$ is the exponential space of assignments to the binary visibility variables. Clearly, naive brute-force search is intractable due to the exponentially large space $\{0, 1\}^{wh}$. We present two inference algorithms: one uses graph-cut for efficient inference over $\mathcal{V}$ and the other further improves the efficiency by a novel branch-and-bound procedure.

4.1. Basic inference algorithm

Since A is usually small, and for each view a different subspace of the weights is activated, we enumerate over views $a \in \{1, 2, \ldots, A\}$, handling each separately. Given the view, we decompose Equation 5 according to the five feature components introduced in Section 3.2 (the decomposition for Equation 3 is analogous):

$$\begin{aligned} &F(\mathbf{x}_k,\ \mathbf{y}; \mathbf{w}) + \Delta(\mathbf{y}_k, \mathbf{y}) \\ &= F^h(\mathbf{x}_k, \mathbf{p}, \mathbf{v}) + F^{\hat{h}}(\mathbf{x}_k, \mathbf{p}, \mathbf{v}) + F^{prior}(\mathbf{v}) \\ &\quad + F^c(\mathbf{x}_k, \mathbf{p}) + F^{pair}(\mathbf{v}) + F^{loss}(\mathbf{x}_k, \mathbf{p}_k, \mathbf{p}) \end{aligned} \tag{6}$$

where we use $\mathbf{p}$ and $\mathbf{v}$ for $\mathbf{y}\langle\mathbf{p}\rangle$ and $\mathbf{y}\langle\mathbf{v}\rangle$ to avoid notation clutter. Here, $F^h(\mathbf{x}_k, \mathbf{p}, \mathbf{v})$ and $F^{\hat{h}}(\mathbf{x}_k, \mathbf{p}, \mathbf{v})$ are the scores from the HOG features of the visible and invisible cells respectively, i.e., $F^h(\mathbf{x}_k, \mathbf{p}, \mathbf{v}) = \sum_{i=1}^{wh} F_i^h(\mathbf{x}_k, p, v_i) = \sum_{i=1}^{wh} \left(\mathbf{w}_{h,i}^T\, \phi_i(\mathbf{x}_k, \mathbf{p})\right) \cdot v_i$ and $F^{\hat{h}}(\mathbf{x}_k, \mathbf{p}, \mathbf{v}) = \sum_{i=1}^{wh} F_i^{\hat{h}}(\mathbf{x}_k, p, v_i) = \sum_{i=1}^{wh} \left(\mathbf{w}_{\hat{h},i}^T\, \phi_i\,(\mathbf{x}_k, \mathbf{p})\right) \cdot (1 - v_i)$ where $\phi_i(\mathbf{x}_k, \mathbf{p})$ is the HOG feature vector from the ith cell and $\mathbf{w}_{h,i}$ and $\mathbf{w}_{\hat{h},i}$ are the weights for the HOG features in the ith cell when it is visible and invisible, respectively. The prior is $F^{prior}(\mathbf{v}) = \sum_{i=1}^{wh} F_i^{prior}(v_i) = \sum_{i=1}^{wh} w_{p,i} \cdot (1 - v_i)$ where $w_{p,i}$ is the bias for the ith cell, independent of the bounding box location $\mathbf{p}$. The term $F^c(\mathbf{x}_k, \mathbf{p}) = w_c \cdot c(\mathbf{p})$ is the truncation component score. If $\mathbf{p}$ is completely inside the image ($c(\mathbf{p}) = 0$), then $F^c(\mathbf{x}_k, \mathbf{p}) = 0$. The pairwise term over $\mathbf{v}$ is $F^{pair}(\mathbf{v}) = \sum_{(i,j)\in\mathcal{E}_v} w_{e,v} \cdot \mathbf{1}\{v_i = v_j\} + \sum_{(i,j)\in\mathcal{E}_h} w_{e,h} \cdot \mathbf{1}\{v_i = v_j\}$, where $w_{e,v}$ and $w_{e,h}$ are vertical and horizontal edge weights respectively. The final score component is $F^{loss}(\mathbf{x}_k, \mathbf{p}_k, \mathbf{p}) = 1 - \frac{\text{area}(\mathbf{p} \cap \mathbf{p}_k)}{\text{area}(\mathbf{p} \cup \mathbf{p}_k)}$, which is independent of both $\mathbf{v}$ and $\mathbf{w}$.

If we only consider a single bounding box $\mathbf{p} \in \mathcal{P}$, then only the value of $\mathbf{v}$ needs to be inferred. This reduces to a MAP problem on a grid Markov Random Field (MRF):

$$\begin{aligned} \hat{\mathbf{v}} = \text{argmax}_{\mathbf{v}} \Bigg(&\sum_{i=1}^{wh} \beta_i(v_i) + \sum_{(i,j)\in\mathcal{E}_v} \beta_{v,ij}(v_i, v_j) \\ &+ \sum_{(i,j)\in\mathcal{E}_h} \beta_{h,ij}(v_i, v_j) + \text{const}(\mathbf{p}) \Bigg) \end{aligned} \tag{7}$$

where $\beta_i(v_i) = F_i^h(\mathbf{x}_k, \mathbf{p}, v_i) + F_i^{\hat{h}}(\mathbf{x}_k, \mathbf{p}, v_i) + F_i^{prior}(v_i)$ and $\beta_{v,ij}(v_i, v_j) = w_{e,v} \cdot \mathbf{1}\{v_i = v_j\}$ and $\beta_{h,ij}(v_i, v_j) = w_{e,h} \cdot \mathbf{1}\{v_i = v_j\}$. If $w_{e,v}$ and $w_{e,h}$ are non-negative then the pairwise terms are submodular. In this case, Equation 7 can be solved efficiently using a single s-t min cut. If the pairwise terms are nonsubmodular, we use QPBO [11] to get an approximate solution. QPBO converts a nonsubmodular energy to a graph cut problem over a larger graph, and is effective when the energy is close to submodular. In our experiments, we found that only the first several iterations of the cutting-plane learning procedure may result in a nonsubmodular energy due to lack of constraints but in later iterations the energy is always submodular. Furthermore, even if the energy is nonsubmodular, QPBO tends to find global optimal assignments for most of our model variables.

Based on the above observation, our basic algorithm to maximize Equation 6 is to enumerate all $(a, \mathbf{p}) \in \{1, 2, \ldots, A\} \times \mathcal{P}$ and use a graph-cut based technique to maximize Equation 7 to get $\hat{\mathbf{v}}$.

4.2. Efficient inference with branch-and-bound

Since $|\mathcal{P}|$ is usually on the order of 10^6 for a typical million-pixel image, the basic method is still computationally expensive. We then propose a best-first branch-and-bound algorithm inspired by [14, 15] to prune the space, resulting in a more efficient inference algorithm. We hierarchically split the space $\mathcal{P} \times \mathcal{V}$ along $\mathcal{P}$ into disjoint subspaces $\{(\tilde{\mathcal{P}}, \mathcal{V})\}$ and use a function $U(\tilde{\mathcal{P}} \times \mathcal{V})$ to upper-bound the score over each subspace $\tilde{\mathcal{P}} \times \mathcal{V}$. We then iteratively examine the subspace with the highest upper bound and split it further, until we find a solution. The hope is that many subspaces with low upper bounds will be worse than the solution found, and we can therefore avoid expanding them entirely. We refer the reader to [14] for details of the general branch-and-bound procedure, and to the supplementary material [1] for further description of the algorithm.

We derive a bound by pushing the maximization over $\mathbf{p} \in \tilde{\mathcal{P}}$ into individual terms of the objective Equation 6. There are three components of the bounds corresponding to different groups of terms in Equation 6. The first component corresponds to all terms involving $\mathbf{v}$ (please refer to the supplemental material for the derivation):

$$\begin{aligned} U^v(\tilde{\mathcal{P}} \times \mathcal{V}) = \max_{\mathbf{v}} \Bigg(& \sum_{i=1}^{wh} \max_{\mathbf{p}} \left(\mathbf{w}_{h,i}^T \, \phi_i(\mathbf{x}_k, \mathbf{p})\right) \cdot v_i \\ & + \sum_{i=1}^{wh} \max_{\mathbf{p}} (\mathbf{w}_{\hat{h},i}^T \, \phi_i(\mathbf{x}_k, \mathbf{p})) \cdot (1 - v_i) \\ & + F^{prior}(\mathbf{v}) + F^{pair}(\mathbf{v}) \Bigg) \end{aligned} \tag{8}$$

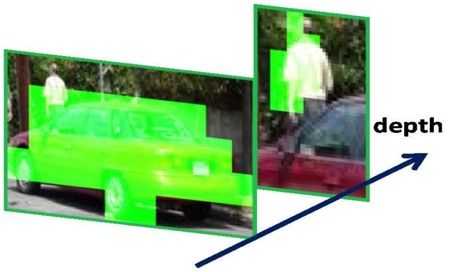

Figure 2: The visibility patterns from multiple objects should be consistent with respect to their relative depth ordering.

This MAP problem over $\mathbf{v}$ can be solved using a single graph cut by the same construction of the graph and potentials as Equation 7; the cost of this computation is independent of $|\tilde{\mathcal{P}}|$ (number of bounding boxes in $\tilde{\mathcal{P}}$).

The second component of the bound corresponds to the empty cell count, which is

$$U^c(\tilde{\mathcal{P}}) = \max_{\mathbf{p}} F^c(\mathbf{x}_k, \mathbf{p}) = \max_{\mathbf{p}} \left(w_c \cdot c(\mathbf{p})\right) \tag{9}$$

Thus, we use $\max_{\mathbf{p}} c(\mathbf{p})$ if $w_c \geq 0$, and $\min_{\mathbf{p}} c(\mathbf{p})$ otherwise. Finally, the bound corresponding to the loss is:

$$U^{loss}(\tilde{\mathcal{P}}) = \max_{\mathbf{p}} F^{loss}(\mathbf{x}_k, \mathbf{p}_k, \mathbf{p}) \leq 1 - \frac{\min_{\mathbf{p}} \text{area}(\mathbf{p}_k \cap \mathbf{p})}{\max_{\mathbf{p}} \text{area}(\mathbf{p}_k \cup \mathbf{p})} \tag{10}$$

Combining the three bound components, we get an upper bound of $\max_{(\mathbf{p},\mathbf{v}) \in \tilde{\mathcal{P}} \times \mathcal{V}} \left(F(\mathbf{x}_k, \ \mathbf{y}; \mathbf{w}) + \Delta(\mathbf{y}_k, \mathbf{y})\right) \leq U^v + U^c + U^{loss} = U(\tilde{\mathcal{P}} \times \mathcal{V})$. Note that, if $\tilde{\mathcal{P}}$ only has a single element, the bound is simply the energy function at that point. Thus, when branch-and-bound considers a partition consisting of a single bounding box, its value is guaranteed to be better than the value of any other bounding box in the image, including those in unexpanded partitions.

5. Globally Coherent Object Model

The model described in Section 3.2 considers the visibility pattern for each candidate detection independently from that of any other detection. However, one of the most indicative cues that a pixel in the image is occluded is when we detect an object that is occluding that pixel. In this section, we build on this intuition to provide a globally coherent explanation for pixels in different detections.

5.1. Consistency-enforcing object model

Specifically we introduce a new term to force the model to commit to an explanation for that cell that is consistent with the other objects in the image.

To facilitate a global understanding of the visibility of all objects in the image, we introduce a variable π that denotes the depth ordering of overlapping detections in a particular region of the image. An example is shown in Figure 2. Formally, π is a mapping from detection index o_k to

[1] www.stanford.edu/~tianshig/papers/segObj-CVPR11-supp.pdf

a depth ranking, where a higher ranking means closer to the viewer. Using this mapping, we can now explicitly take into account what is in front of a particular object by modeling mutual exclusion, which enforces a penalty if two objects o_k and o_l are both labeled visible for the same cell. This pairwise term between overlapping cells of two objects is defined as: $\psi^{\text{mi}}_{o_k,i}(v_i^{o_k}, \vec{v_i}^{-k}, \pi) = \theta^{mi}\mathbf{1}\{v_i^{o_k} = 1\}$ if exists l s.t. $\pi(o_l) > \pi(o_k)$ and $v_i^{o_l} = 1$, otherwise 0, where $\vec{v_i}^{-k}$ is the vector of visibility variables for all objects other than o_k that cover cell i. The full energy for our global model add this term to those in our original energy function Equation 7 described in Section 3.2.

5.2. Inference

We propose to approximately optimize this energy function using a simple greedy algorithm which is simple to implement and computationally efficient.

We begin by running the baseline object detector for all classes over the image, to generate a set of candidate detections. For computational efficiency, we prune the set of candidates to the highest scoring 20 after non-maximum suppression. We then divide the candidates into regions of possible confusion using a connected components algorithm. Thus, each connected component consists of a set of detections that may be occluding each other.

Within each connected component, we enumerate all possible orderings π. Since we pruned the number of candidate detections, we typically have at most 6 detections within a connected component, so that the $6!$ possible orderings are reasonably tractable to enumerate exhaustively. The algorithm now infers the visibility variables within the global model for each such connected component given the depth ordering π. We initialize all visibility variables for all detections to be 0, and then incrementally infer the values of the visibility variables for each object one at a time, moving from front to back. Despite the complex nature of the energy terms, given the visibility values of all objects in front of the current object o^k, and the fact that all visibility variables for objects in the back of o^k are 0, the energy terms for o^k simplify significantly. Specifically, we have only singleton potentials for the individual visibility variables and pairwise smoothness terms that are submodular. Thus, this (greedy) step of the optimization can be performed efficiently and exactly by graph cut inference.

6. Experimental Results

We conduct experiments on two challenging datasets: (1) Street Scenes from [1]; (2) Pascal VOC 2007 [5] on the bicycle, horse, motorbike, aeroplane and train classes. Our supervised training regime requires the labeling of the bounding box corresponding to the full extent of the object (not a bounding box just for the visible part of the object), the object pixels within the bounding box, and the view (*e.g.*, left, right, frontal and rear). We used Amazon Mechanical Turk to label the images. The labeling of the visibility binary variables are converted from the pixel labeling mask.

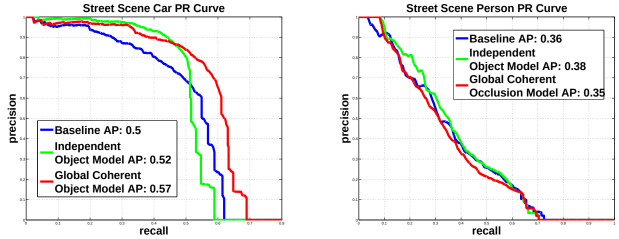

Figure 3: PR curves for car (left) and person(right) on the Street Scene dataset. Average precisions for baseline, independent, and global are: cars — 0.5, 0.52, $\mathbf{0.57}$; person — 0.36, $\mathbf{0.38}$, 0.35.

6.1. The segmentation-aware object model

We first tested our segmentation-aware object model described in Section 3.2. The baseline model is introduced in Section 3.1. For the Street Scenes dataset, we randomly sampled 400 images of size 960×720. The dataset contains 1219 cars and 445 persons, and for both categories, around 50% of the instances are occluded by various degrees. We used 5-fold cross-validation to report our results. As shown in Figure 3 (point-wise average of PR curves from 5 folds), our model performs 2% better on both person and car than the baseline model. The performance boost is consistent on 4 out of 5 folds. We also tested on the horse, bicycle, motorbike, aeroplane and train classes in the Pascal VOC2007 dataset. As shown in Table 1, our model did better on bicycle, motorbike, aeroplane and train but similarly on the horse. In addition to a better detection performance, our model has a richer output than just a bounding box for the detection. Some sample detections from our model are shown in Figure 4. Note that the focus of this work is on the detection task, and the model was trained with a loss function tailored to this task. Therefore, we do not expect the model to produce competitive pixel-level segmentation, though one could augment the loss to penalize segmentation errors in our framework.

Object	baseline	segmentation-aware model
Bicycle	0.234	$\mathbf{0.291}$
Horse	$\mathbf{0.235}$	0.227
Motorbike	0.155	$\mathbf{0.168}$
Aeroplane	0.080	$\mathbf{0.091}$
Train	0.125	$\mathbf{0.137}$

Table 1: Average precisions on Pascal VOC2007 dataset.

Since our model explicitly infers the cells that support the object, we explored whether it is more robust to occlusion by examining how the detector performance changes

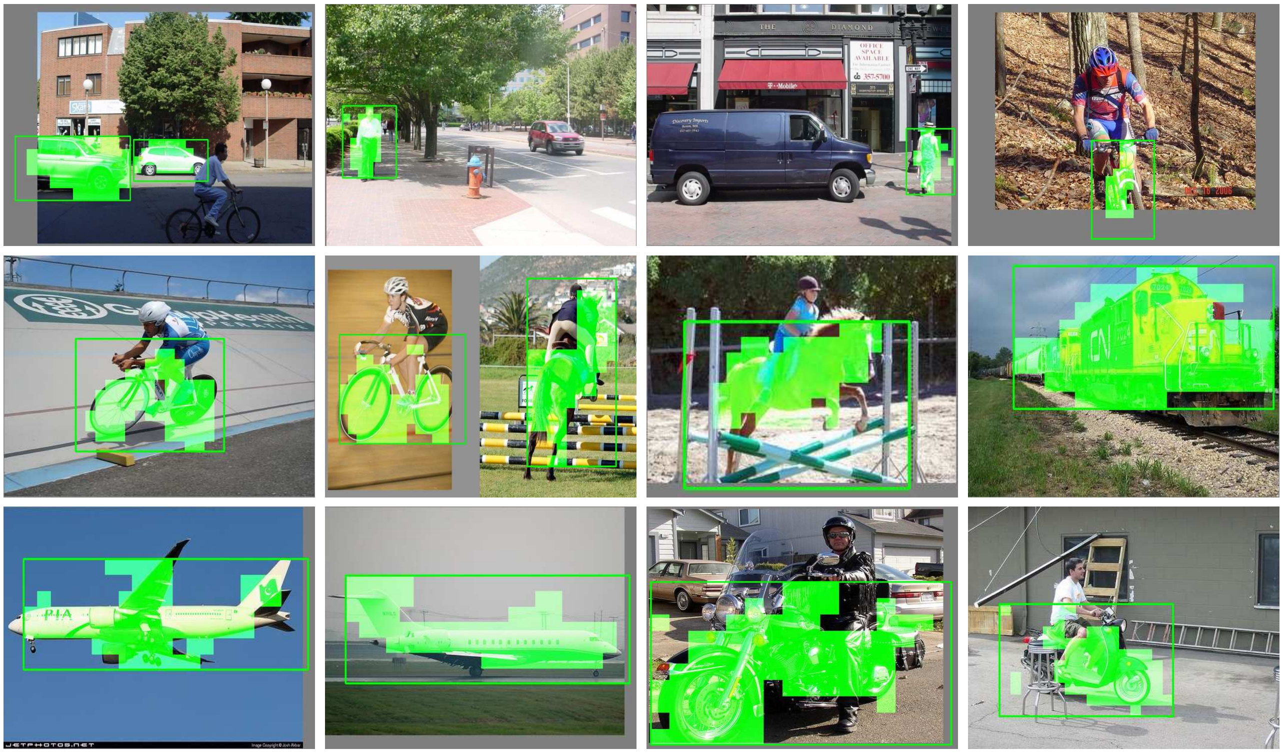

Figure 4: Example images from the Street Scenes and the Pascal VOC 2007 datasets demonstrating the output of our segmentation-aware object model for each individual object category. Shown are the inferred bounding boxes and visibility masks. Note that our model handles objects with multiple views and truncations at the image boundary.

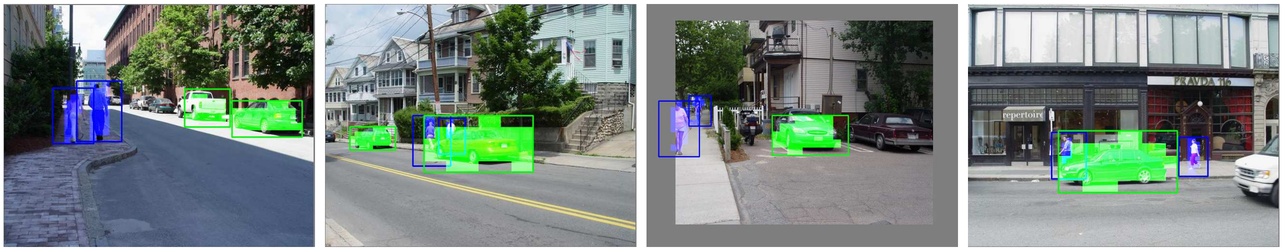

Figure 5: Sample detections of our global coherent object model on the Street Scenes. Shown are the inferred bounding boxes and visibility patterns. The color encodes the object category. Note that the global coherent model considers both the car and person simultaneously. For the second and fourth images, some persons are correctly inferred as behind the cars. Furthermore, the visibility masks for the occluded persons and the occluding cars are consistent. In this case, a normal sliding window detector rejects the persons behind the cars, since the bounding boxes are overlapping significantly (but not their visibility masks in our model).

when the degree of occlusion (DoO) varies. Similar to [16], we split the samples into two bins based on the percentage of the pixels belonging to the object inside the bounding box according to the ground truth. The first bin corresponds to DoO $< 50\%$ and the other corresponds to DoO $\geq 50\%$. On the street scene dataset, for the person, our model outperforms the baseline by 2% in both bins. For the car, our model outperforms the baseline by 1% in the first bin, but by 6% in the second bin with higher DoO. Our model thus outperforms the baseline when there is both partial and heavy occlusion, but the benefit is more pronounced in the latter case. For Pascal, we cannot perform this particular analysis due to the lack of segmentation labels in the test set.

As for inference, the way we split the bounding box space in our experiment is as follows: for a bounding box space specified by a subspace of the image pyramids $[s_1, s_2] \times [w_1, w_2] \times [h_1, h_2]$ where s, w and h are the scale, width and height (normalized to $[0, 1]$), we split the dimension with the largest interval into half. We compare the speeds by recording the number of graph-cuts used by enumerating all bounding boxes and by our branch-and-bound (BB) algorithm. This number is independent of implementation details and different physical machines in use. Empirically, the inference is $1.5 - 3$ times faster with BB. For example, for car, BB reduces the number of graph-cuts from 150K (enumerating all boxes) to 56.3K per image; for person, it is reduced from 54K to 35K. The absolute average running time of a 3-view object model on a typical Pascal image is around 2 seconds using a 2.66GHz CPU.

6.2. Globally coherent object model

We also tested the globally coherent object model described in Section 5. We estimated the potential parameter θ^{mi} by varying it from 0.1 to 1000 (increment by 10x a time) and choosing the one yielding the best performance on the training set. As shown in Figure 3, our globally coherent object model did much better on cars than both the baseline and the independent model. For person, it performed slightly worse. This may because that in this dataset, many persons are occluded by cars, and in this case, once a cell is declared by a car in front, a person has to pay a cost to turn it on. However, due to cell quantization and different scales of the objects, a cell at the boundary of a car may cover several cells from the person, influencing the scores of the person in a more noisy way. A finer quantization or a better calibration of the scores from different object classes may help alleviate this problem. Some example detections are shown in Figure 5. Note that in the global coherent model, we consider multiple object classes, *e.g.*, both person and car, at the same time. As can be seen, our model correctly inferred the relative depth ordering between occluding cars and occluded persons with consistent visibility masks.

7. Discussion

We have provided a segmentation-aware object detection model which solves the detection and segmentation simultaneously. The rich representation makes the detection more robust to occlusion and offers a richer output. Our inference algorithm combining graph-cut and branch-and-bound is novel and efficient. Finally, our globally coherent object model incorporates semantic information of the object-object spatial relationship by jointly inferring the relative depth ordering of multiple detections. Directions for future work include using the rich representation on the deformable parts-based model [6], treating the binary visibility variables as hidden during the training, the use of a less greedy inference algorithm for the globally consistent object model, and the incorporation of additional cues regarding occlusion, such as occlusion boundaries [9] or cues regarding the relative position of the object on the ground [7].

Acknowledgements. This work was supported by the NSF under grant No. RI-0917151 and MURI contract N000140710747. We thank Stephen Gould and Pawan Kumar for helpful discussions.

References

[1] S. Bileschi and L. Wolf. A unified system for object detection, texture recognition, and context analysis based on the standard model feature set. In *BMVC*, 2005. 1362, 1366

[2] M. Blaschko and C. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008. 1364

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1361, 1362

[4] M. Eichner and V. Ferrari. We are family: joint pose estimation of multiple persons. In *ECCV*, 2010. 1362

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007. 1362, 1366

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. In *PAMI*, 2009. 1361, 1362, 1368

[7] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 1368

[8] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *NIPS*, 2009. 1361, 1362

[9] D. Hoiem, A. Stein, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007. 1368

[10] T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009. 1362, 1363, 1364

[11] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *PAMI*, 2007. 1365

[12] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Objcut: Efficient segmentation using top-down and bottom-up cues. *PAMI*, 32(3):530–545, 2010. 1361, 1362

[13] L. Ladicky, P. Sturgess, K. Alahari, C. Russell, and P. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*, 2010. 1362

[14] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows:object localization by efficient subwindow search. In *CVPR*, 2008. 1365

[15] V. Lempitsky, A. Blake, and C. Rother. Image segmentation by branch-and-mincut. In *ECCV*, 2008. 1365

[16] B. S. P. Dollar, C. Wojek and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009. 1361, 1363, 1367

[17] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, Vancouver, Canada, 2004. 1363

[18] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 1363

[19] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial truncation. In *NIPS*. 2009. 1362, 1363

[20] X. Wang, T. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009. 1362

[21] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006. 1361, 1362

[22] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object detection for multi-class segmentation. In *CVPR*, 2010. 1361, 1362