Discriminative models for multi-class object layout

Chaitanya Desai Deva Ramanan Charless Fowlkes
Department of Computer Science
U C Irvine
{desaic,dramanan,fowlkes}@ics.uci.edu

Abstract

Many state-of-the-art approaches for object recognition reduce the problem to a 0-1 classification task. Such reductions allow one to leverage sophisticated classifiers for learning. These models are typically trained independently for each class using positive and negative examples cropped from images. At test-time, various post-processing heuristics such as non-maxima suppression (NMS) are required to reconcile multiple detections within and between different classes for each image. Though crucial to good performance on benchmarks, this post-processing is usually defined heuristically.

We introduce a unified model for multi-class object recognition that casts the problem as a structured prediction task. Rather than predicting a binary label for each image window independently, our model simultaneously predicts a structured labeling of the entire image. Our model learns statistics that capture the spatial arrangements of various object classes in real images, both in terms of which arrangements to suppress through NMS and which arrangements to favor through spatial co-occurrence statistics.

We formulate parameter estimation in our model as a max-margin learning problem. Given training images with ground-truth object locations, we show how to formulate learning as a convex optimization problem. We employ a cutting plane algorithm similar to [14] to efficiently learn a model from thousands of training images. We show state-of-the-art results on the PASCAL VOC benchmark that indicate the benefits of learning a global model encapsulating the spatial layout of multiple object classes.

1. Spatial Interactions in Object Detection

A contemporary and successful approach to object recognition is to formulate it as a classification task, e.g. “Does an image window at location i contain a given object o?”. The classification formulation allows immediate application of a variety of sophisticated machine learning techniques in order to learn optimal detectors from training data. Such methods have the potential to encapsulate those subtle statistical regularities of the visual world which separate object from background. As a result, learning approaches have often yielded detectors that are more robust and accurate than their hand built counterparts for a range of applications, from edge and face detection to general purpose object recognition [25, 6, 8].

In contrast to the well founded techniques used for classification of individual image patches, the problem of correctly detecting and localizing multiple objects from multiple classes within an image of a scene has generally been approached in a far more ad-hoc manner. For example, *non-max suppression* (NMS) is required to remove some detections returned by a classifier based on overlap criteria or more complicated heuristics (e.g. the mode finding approach of [6]). Such tricks of the trade are essential to good performance on benchmarks designed to penalize multiple non-localized detections, however, they highlight a clear disconnect between training and testing phases. The objective optimized during learning only characterizes a subcomponent of the final system used at runtime.

Furthermore, there is a wide range of possible interactions between object detections which is not fully captured by ad-hoc approaches. In street-level views, pedestrians are likely to occur standing next to each other, nearly overlapping, but unlikely to occur directly above or below each

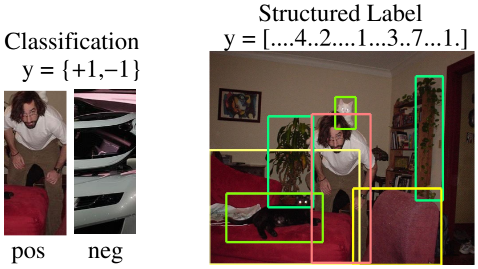


Figure 1. Our framework. Classification-based approaches for recognition predict a binary label for a cropped window (**left**). We formulate the recognition problem as predicting a sparse, structured label vector specifying which windows, if any, contain particular objects in an entire input image. The latter allows our model to capture a wide range of contextual constraints among objects as described in Table 1 and Fig 2.

	within-class	**between-class**
activation	textures of objects	spatial cueing
inhibition	NMS	mutual exclusion
global	expected counts	co-occurrence

Table 1. A taxonomy of interactions captured in our model. **Within** a single object class, our model can favor typical spatial layouts of objects (people often stand in crowds) while directly learning how to inhibit overlapping detections in such cases (NMS). Our model also captures long-range interactions between objects, such as the constraint that there exists at most one object instance (counting). Analogous interactions exist **between** object classes, including typical spatial relations between objects (bottles sit on tables), mutual exclusion (dog and cat detectors should not respond to the same image region), and co-occurrence (couches and cars do not commonly co-occur).

other (Fig.2). In general, spatial object-object interactions may be arbitrarily complex and depend on latent information which is not readily available from single image. As an extreme example, studies of *proxemics* [11], the body spacing and pose of people as they interact, shows that physical spacing between people depends in complicated ways on their "social distance". While such complex interactions are difficult to encode, we argue there does exist useful information that is being ignored by current ad-hoc approaches to NMS.

NMS is generally described in terms of intra-class inhibition, but can be generalized to suppression of overlapping detections between different classes. We refer to this more general constraint, that two objects cannot occupy the same 3D volume at the same time, as *mutual exclusion*. As seen in a 2D image projection, the exact nature of this constraint depends on the object classes. Fig.2(right) shows an example of ground-truth labelings in the PASCAL VOC dataset in which strict mutual-exclusion would produce suboptimal performance.

Object detections can also serve to enhance rather than inhibit other detections within a scene. This has been an area of active research in object recognition over the last few years [22, 18, 10, 12, 13, 4, 15]. For example, different object classes may be likely to co-occur in a particular spatial layout. People ride on bikes, bottles rest on tables, and so on. In *contextual cueing*, a confident detection of one object (a bike) provides evidence that increases the likelihood of detecting another object (a person above the bike) [4, 10, 15]. Contextual cueing can also occur within an object category, e.g., a crowd of pedestrians reinforcing each other's detection responses. An extreme example of this phenomena is *near-regular texture* in which the spatial locations of nearly identical elements provides a strong prior on the expected locations of additional elements, lowering their detection threshold [17].

In Table 1 we outline a simplified taxonomy of different types of object-object interactions, both positive and negative, within and between classes. The contribution of this paper is a single model that incorporates all interactions from Table 1 through the framework of structured prediction. Rather than returning a binary label for a each image window, our model simultaneously predicts a set of detections for multiple objects from multiple classes over the entire image. Given training images with ground-truth object locations, we show how to formulate parameter estimation as a convex max-margin learning problem. We employ the cutting plane algorithm of [14] to efficiently learn globally optimal parameters from thousands of training images.

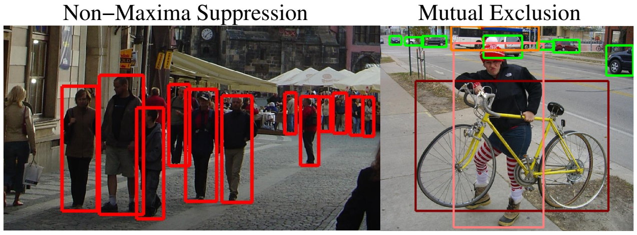


Figure 2. Our novel contributions include the ability to learn inhibitory intra-class constraints (NMS) and inhibitory inter-class constraints (Mutual Exclusion) in a single unified model along with contextual cuing and spatial co-occurrence. Naive methods for NMS or mutual exclusion may fail for objects that tend to overlap themselves (**left**) and other objects (**right**). In contrast, our framework *learns* how best to enforce such constraints from training data. We formulate the tasks of NMS and Mutual Exclusion using the language of structured prediction. This allows us to compute an optimal model by minimizing a convex objective function.

In the sections that follow we formulate the structured output model in detail, describe how to perform inference and learning, and detail the optimization procedures used to efficiently learn parameters. We show state-of-the-art results on the PASCAL 2007 VOC benchmark[7], indicating the benefits of learning a global model that encapsulates the layout statistics of multiple objects classes in real images. We conclude with a discussion of related work and future directions.

2. Model

We describe a model for capturing interactions across a family of object detectors. To do so, we will explicitly represent an image as a collection of overlapping windows at various scales. The location of the i^{th} window is given by its center and scale, written as $l_i = (x, y, s)$. The collection of M windows are precisely the regions scored by a scanning-window detector. Write x_i for the features extracted from window i, for example, a histogram of gradient features [6]. The entire image can then be represented as the collection of feature vectors $X = \{x_i : i = 1 \ldots M\}$

Assume we have K object models. We write $y_i \in \{0 \ldots K\}$ for the label of the i^{th} window, where the 0 la-

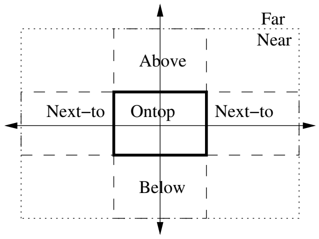


Figure 3. A visualization of our spatial histogram feature d_{ij}. We consider the location of the center of window j with respect to a coordinate frame defined by window i, denoted by the thickly outlined box. The dashed and dotted rectangles represent regions over which the center of window j are binned. The relative location of j must either be `far` or `near`. For near windows, we consider `above`, `ontop`, `below`, and symmetric `next-to` bins as shown. To allow our model to reproduce the behavior of baseline modules that perform NMS with a criteria of 50% relative overlap, we also include a binary overlap feature. This makes d_{ij} a 7 dimensional sparse binary vector.

bel designates the background. Let $Y = \{y_i : i = 1 \dots M\}$ be the entire label vector for the set of all sub-windows in an image. We define the score of labeling image X with vector Y as:

$$S(X,Y) = \sum_{i,j} w_{y_i,y_j}^T d_{ij} + \sum_i w_{y_i}^T x_i \tag{1}$$

where w_{y_i,y_j} represent weights that encode valid geometric configurations of object classes y_i and y_j, and w_{y_i} represents a local template for object class y_i. d_{ij} is a spatial context feature that bins the relative location of window i and j into one of D canonical relations including above, below, overlapping, next-to, near, and far (Fig3). Hence d_{ij} is a sparse binary vector of length D with a 1 for the kth element when the kth relation is satisfied between the current pair of windows. w_{y_i,y_i} encodes the valid geometric arrangements of a single class. For example, if people occur beside one another but not above, the weight from w_{y_i,y_i} associated with next-to relations would then be large.

Local model: In our current implementation, rather than learning a local template, we simply use the output of the local detector as the single feature. To learn biases between different object classes, we append a constant 1 to make x_i two-dimensional.

Background class. Since we are concerned only with the relative difference in scores between labelings, we have an extra degree of freedom in defining the weights. We constrain local and pairwise background weights w_0 and w_{i0} and w_{0i} to be 0. Since the majority of windows in an image will be labelled as background, this significantly speeds up computations with the model.

3. Inference

Computing $\arg\max_Y S(X,Y)$ is NP hard unless the pairwise potentials happen to have some particular structure (e.g. sub-modularity with $K = 2$). For more general cases, one must resort to search techniques such as branch-and-bound or A* to find exact minima. In our experiments, we use a simple greedy forward search.

3.1. Greedy Forward Search

Our algorithm for optimizing (1) is analogous to greedy algorithms traditionally used for NMS [16]. (1) Initialize the label vector Y to the background class for each window. (2) Greedily select the single window that, when labelled as a non-background class, increases the score S by the largest amount. (3) Stop when instancing any other detection decreases the total score. Naively re-computing the score at each step of the algorithm takes excessively long but we can track the potential gain of adding each detection incrementally.

We write I for a particular set of instanced window-class pairs $\{(i,c)\}$ and write $Y(I)$ for the associated label vector where $y_i = c$ for all pairs in I and $y_i = 0$ otherwise. We define the change in score obtained by adding window-class pair (i,c) to the set of instances I as

$$\Delta(i,c) = S(X, Y(I \cup \{(i,c)\})) - S(X, Y(I))$$

Initialize $I = \{\}, S = 0$ and $\Delta(i,c) = w_c^T x_i$ and repeat:

1. $(i^*, c^*) = \arg\max_{(i,c) \notin I} \Delta(i,c)$
2. $I = I \cup \{(i^*, c^*)\}$
3. $S = S + \Delta(i^*, c^*)$
4. $\Delta(i,c) = \Delta(i,c) + w_{c^*,c}^T d_{i^*,i} + w_{c,c^*}^T d_{i,i^*}$

until $\Delta(i^*, c^*) < 0$ or all windows are instanced. In Step 4, we update $\Delta(i,c)$ for un-instanced window-class pairs by adding in the pairwise costs due to the newly instanced pair (i^*, c^*). For additional speed ups, we ran the above algorithm on a set of windows that passed an initial minimal threshold and conservative NMS step. This substantially reduces the number of windows the algorithm must consider.

Effectiveness: On small-scale problems where the the brute-force solution is computable, we find that the above algorithm tends to produce solutions whose score is quite close to the score of the optimal solution, though the solutions may differ in the labels of 1-3 windows. This suggests our algorithm performs reasonably well in practice, though we are investigating other optimization methods.

3.2. Marginals

Many object recognition benchmarks such as PASCAL are scored by ranking detections with a precision-recall curve. This means we need to associate a score with each detected window. To obtain a score, we can appeal to a probabilistic version of our model, which would correspond to a conditional random field (CRF) written as $P(Y|X) =$

$\frac{1}{Z(X)}e^{S(X,Y)}$. The natural score for an individual detection would be the posterior marginal or log-odds ratio

$$m(y_i = c) \;=\; \log \frac{P(y_i = c|X)}{P(y_i \neq c|X)} \qquad (2)$$
$$= \log \frac{\sum_{y_{\mathbf{r}}} P(y_i = c, y_{\mathbf{r}}|X)}{\sum_{y_{\mathbf{s}}, c' \neq c} P(y_i = c', y_{\mathbf{s}}|X)}$$

We write $y_{\mathbf{r}}$ for a $M-1$ vector of labels for the remaining $M-1$ windows other than i. We similarly define $y_{\mathbf{s}}$. Both sums above require marginalizing out an exponential number of labels. Let us assume the posterior mass inside each sum is dominated by the most probable label $\mathbf{r}^*$ and the second best label $\mathbf{s}^*$ with class c^* respectively.

$$\mathbf{r}^* = \arg\max_{\mathbf{r}} S(X, y_i = c, y_{\mathbf{r}}) \qquad (3)$$
$$(\mathbf{s}^*, c^*) = \arg\max_{(\mathbf{s}, c' \neq c)} S(X, y_i = c', y_{\mathbf{s}})$$

Then the marginal log-odds ratio (2) can approximated by

$$m(y_i = c) \approx \log \frac{P(y_i = c, y_{\mathbf{r}^*}|X)}{P(y_i = c^*, y_{\mathbf{s}^*}|X)}$$
$$= S(X, y_i = c, y_{\mathbf{r}^*}) - S(X, y_i = c^*, y_{\mathbf{s}^*})$$

It is straightforward to extend our greedy maximization procedure for optimizing equation (1) to solve equation (3). This is used for the per detection scoring presented in the result section.

4. Learning

In order to describe the learning algorithm, we first rewrite the score function from (1) in terms of a single linear parameter vector w. To do this, we encapsulate the effect of Y and X in a potential function, writing

$$S(X,Y) = \sum_{i,j} w_s^T \psi(y_i, y_j, d_{ij}) + \sum_i w_a^T \phi(x_i, y_i)$$

where w_s and $\psi()$ are vectors of length DK^2, and w_a and $\phi()$ are vectors of length KF, where D is the number of spatial relations, K is the number of classes and F is the length of feature vector x_i. In general, each object class may use a feature vector of different length. The vector $\psi()$ will contain at most D nonzero entries and the vector $\phi()$ will contain only F nonzero entries. We can then write the score as $S(X,Y) = w^T\Psi(X,Y)$ where

$$w = \begin{bmatrix} w_s \\ w_a \end{bmatrix} \quad \Psi(X,Y) = \begin{bmatrix} \sum_{ij} \psi(y_i, y_j, d_{ij}) \\ \sum_i \phi(x_i, y_i) \end{bmatrix} \qquad (4)$$

where our greedy inference procedure solves

$$Y^* = \arg\max_Y w^T \Psi(X,Y) \qquad (5)$$

4.1. Convex training

Assume we are given a collection of training images X_n and labels Y_n. We want to train a model w, that given a new image X_n, tends to produce the true label vector $Y_n^* = Y_n$. We formulate this as a regularized learning problem:

$$\arg\min_{w, \xi_n \geq 0} \quad w^T w + C \sum_n \xi_n \qquad (6)$$
$$\text{s.t.} \quad \forall n, H_n \quad w^T \Delta\Psi(X_n, Y_n, H_n) \geq l(Y_n, H_n) - \xi_n$$

where $\Delta\Psi(X_n, Y_n, H_n) = \Psi(X_n, Y_n) - \Psi(X_n, H_n)$. The constraint from (6) specifies the following: Consider the n^{th} training image X_n and its true label Y_n. We want the true label to score higher than all other hypothesized labelings $\{H_n\}$. However not all incorrect labelings are equally bad. The loss function $l(Y_n, H_n)$ measures how incorrect H_n is and penalizes the slack variable ξ_n in proportion. This loss function formulation from (6) is often called margin-rescaling [23].

We consider notions of loss that decompose across the M windows: $l(Y,H) = \sum_{i=1}^M l(y_i, h_i)$. One simple window-specific loss is 0-1:

$$l_{01}(y_i, h_i) = I(y_i \neq h_i)$$

Hence, the constraint from (6) requires that label Y scores much higher than those hypotheses H that differ from the ground-truth on many windows. However note that l_{01} incorrectly penalizes detections that overlap true positives as false positives. A more appropriate loss that handles overlap a bit better is:

$$l_{ov}(y_i, h_i) = \begin{cases} 1: & y_i \neq bg \wedge h_i \neq y_i \\ 1: & h_i \neq bg \wedge \neg \exists j \\ & \text{s.t.}[ov(i,j) > .5 \wedge y_j = h_i] \\ 0: & \text{otherwise} \end{cases} \qquad (7)$$

The top condition corresponds to a missed detection, while the second corresponds to a false positive (where we check to make there does not exist an overlapping true detection). One may also define a soft loss that assigns a value between 0 and 1 for partially overlapping windows, as in [5].

5. Cutting plane optimization

The training problem outlined in (6) is exactly the formulation addressed by the excellent SVMStruct package [14]. However, in order to allow us greater flexiblity in caching data vectors and scheduling model updates, we opted to develop our own solver implementing the 1-slack cutting plane algorithm [14]. We briefly outline our approach here using the notation from [21].

Consider the following unconstrained formulation that is equivalent to the constrained problem from (6):

$$w^* = \arg\min_w L(w) \quad \text{where} \quad L(w) = \frac{1}{2}||w||^2 + CR(w)$$

$$R(w) = \sum_n^N \max_H(0, l(Y_n, H) - w^T \Delta\Psi(X_n, Y_n, H))$$

In the above formulation, $R(w)$ is a convex function since it is the maximum of a set of linear functions. This proves that the overall objective function $L(w)$ in convex since it is the sum of two convex functions.

Following [21]'s notation, we call (6) the master problem and define the following reduced problem:

$$w_t = \arg\min_w L_t(w) \text{ where } L_t(w) = \frac{1}{2}||w||^2 + CR_t(w) \tag{8}$$

where the convex hinge loss R is approximated by a piecewise linear function R_t. The approximation is constructed from a small set of lower-tangent planes called *cutting planes*. Each cutting plane will be a sub-gradient g of the function $R(w)$ computed at a particular point w_j. The subgradient is computed as:

$$g(w_j) = -\sum_{n=1}^N \pi_n \Delta\Psi(X_n, Y_n, H_n^*)$$

$$\pi_n = \begin{cases} 1 & \text{if} \quad l(Y_n, H_n^*) - w_j^T \Delta\Psi(X_n, Y_n, H_n^*) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$H_n^* = \arg\max_H l(Y_n, H) - w^T \Delta\Psi(X_n, Y_n, H) \tag{9}$$

where H_n^* is the most violated constraint for image n under the current weight vector w. The subgradient provides a linear lower bound for $R(w)$.

$$R(w) \geq R(w_j) + g(w_j)^T(w - w_j) \quad \forall w \tag{10}$$

To obtain a tighter lower bound of $R(w)$, we will take the point-wise maximum of cutting planes computed at points $w_1 \ldots w_{t-1}$, adding the zero-plane to the set since the hinge loss R is nonnegative:

$$R_t(w) = \max(0, \max_{j=1\ldots t-1} w^T g(w_j) + b_j) \quad \forall w \tag{11}$$

5.1. Algorithm

Initialize $t = 0$ and the set of cutting planes to be empty. Iterate:

1 Compute w_t by solving the reduced problem (8). This can be solved with a dual QP with t variables. Since t is typically small (10-100), this can be solved with off-the-shelf solvers. We use the publicly available simplex solver from [1].

2 Add the new cutting plane $w^T g(w_t) + b_t$.

As in [21], we iterate until the stopping condition $L(w_t) - L_t(w_t) < \epsilon$. Define the optimal solution as $L^* = \min_w L(w)$. It is relatively straightforward to show that $\forall t$, we have the lower and upper bounds $L_t(w_t) \leq L^* \leq L(w_t)$. The iteration must terminate because the lower bound is non-decreasing $L_t(w_t) \geq L_{t-1}(w_{t-1})$ and there exists a finite number of cutting planes.

5.2. Finding most-violated constraint

In step (2) of (5.1), we need to compute the subgradient of $R(w)$ at the current w_t. To do so, we need to compute the most violated constraint for image n (9). Dropping the n subscript notation, we can rewrite (9) as

$$\begin{aligned} H^* &= \arg\max_H l(Y, H) + w^T \Psi(X, H) \\ &= \arg\max_H \sum_{i,j} w_{h_i,h_j}^T d_{ij} + \sum_i (w_{h_i}^T x_i + l(h_i, y_i)) \end{aligned} \tag{12}$$

Since the loss function decomposes into a sum over windows, solving (12) is very similar to the original maximization (1) except that the local match costs have been augmented by the loss function. Using the loss function (7), the local scores for invalid object labels for a given window are incremented by one. This makes these labels more attractive in the maximization, and so they are more likely to be included in the most-violated constraint H^*. We can compute an approximate solution of (12) with a greedy forward search as in Section 3.1. Our algorithm is an under-generating approximation [9], implying that formal garuntees of optimal learning no longer hold. However, as stated in Sec.3.1, greedy forward search tends to produce scores similar to the brute-force solution, and so we suspect our solutions are close to optimal.

6. Results

Testbed: We have focused our experimental results for multiclass object recognition on the PASCAL Visual Object Challenge. It is widely regarded as the most difficult available benchmark for recognition. We use the 2007 data which is the latest for which test annotations are available. The data consists of 10000 images spanning 20 object classes with a 50% test-train split. The images are quite varied, making this an especially difficult testbed for high-level contextual reasoning.

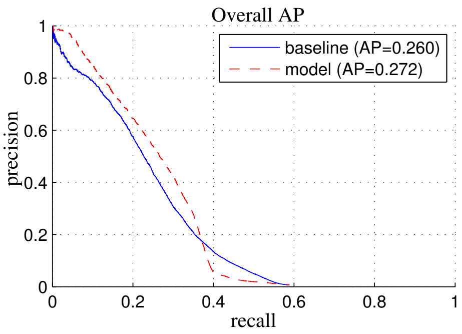
Baseline: State-of-the-art approaches tend to be scanning window detectors [7]. We use the publicly available code [2] as a baseline. It implements a intra-class NMS post-processing step. The code is an improved version of [8] that out-scores many of the previous best performers from the 2007 competition, suggesting it is a strong baseline for comparison.

Class	[7]	Baseline [8]	MC-NMS	Our model
plane	.262	0.278	0.270	**0.288**
bike	.409	0.559	0.444	**0.562**
bird	**.098**	0.014	0.015	0.032
boat	.094	**0.146**	0.125	0.142
bottle	.214	0.257	0.185	**0.294**
bus	**.393**	0.381	0.299	0.387
car	.432	0.470	0.466	**0.487**
cat	**.240**	0.151	0.133	0.124
chair	.128	**0.163**	0.145	0.160
cow	.140	0.167	0.109	**0.177**
table	.098	0.228	0.191	**0.240**
dog	**.162**	0.111	0.091	0.117
horse	.335	0.438	0.371	**0.450**
motbike	.375	0.373	0.325	**0.394**
person	.221	0.352	0.342	**0.355**
plant	.120	0.140	0.091	**0.152**
sheep	**.175**	0.169	0.091	0.161
sofa	.147	0.193	0.188	**0.201**
train	.334	0.319	0.318	**0.342**
TV	.289	**0.373**	0.359	0.354

Table 2. Per-class AP scores on PASCAL 2007 [7]. We show the winning score from the 2007 challenge in the **first** data column. This column is composed of various state-of-the-art recognition algorithms. The **second** column is our baseline obtained by running the code from [2]. It outperforms many of the 2007 entries, suggesting it is a strong baseline for comparison. The **third** column pools detections across multiple classes before applying NMS procedure from [2] (MC-NMS). The **third** column is our approach, which provides a stark improvement over MC-NMS and generally improves performance over classification-trained approaches.

Per-class scores: We follow the VOC protocol for reporting results [7]. A putative detection is considered correct if the intersection of its bounding box with the ground-truth bounding box is greater than 50% of their union. Multiple detections for the same ground-truth are considered false positives. We compute Precision-Recall (PR) curves and score the average precision (AP) across classes in Table 2. For twelve of the twenty classes, we achieve the best score when compared to the 2007 competition and the baseline model. We also compare to a version of [8] in which detections from multiple classes are pooled before applying NMS (MC-NMS). This tends to hurt performance, indicating the need for proper training of multiclass inhibition. The improvement over MC-NMS is generally large. In most cases, the improvement over the baseline is small, but for indoor classes such as tables and bottles and outdoor classes such as motorbikes and trains, the improvement is close to 10%.

Multi-class scores: Per-class APs do not score the consistency of detections across classes on an image, which is one of our goals for multi-class recognition. We consider two approaches for multiclass scores in Fig.4. First we pool detections across classes and images (running the default NMS procedure in [2] before pooling), and generate a single PR curve. Our model provides a noticeable improvement, particularly in the high precision - low recall regime. We also pool detections on a per image bases, generating a per-image multi-class AP. We average this AP across all images. Our model again provided a strong improvement of 10% over the baseline. This is because the baseline does not correctly reconcile detections from various classes due to the fact that the detectors were trained independently.



Average
Per–Image AP

Baseline	.377
Our model	.404

Figure 4. Multi-class AP scores on PASCAL 2007. On the **left**, we score overall AP. We construct the baseline curve by pooling detections across classes and images when computing PR curves. Our global model clearly provides a noticeable boost in performance in the low-recall high-precision regime. On the **right**, we pool detections on a per-image base, compute the per-image AP, and average the result over images. We see a noticeable improvement of 10% over our baseline [2].

Models: We visualize the pairwise weights learned in our models in both Fig.5 and Fig.6. These are trained discriminatively, taking into account the behavior of the local detector. For example, our model learns to aggressively compete bottle and person detections because local detectors confuse the two. This is contrast to simple co-occurrence weights that are trained by frequency counting as in [10, 4]. We also learn meaningful multiclass spatial layouts - e.g., bottles tend to occur above tables. We refer the reader to the captions for additional analysis. Fig.7 shows example multi-class detections from our model as compared to the baseline. Our model appears to produce better detections by understanding interactions between objects that spatially overlap, such as people when riding horses. It also learns how to correctly enforce mutual exclusion between classes, allowing people and sofas to overlap but not people and bottles.

7. Discussion and Related Work

There has been a wide variety of work in the last few years on contextual modeling in image parsing [22, 20, 13, 10, 19, 12, 3]. These approaches have typically treated the problem as that of finding a joint labeling for a set of pixels, super-pixels, or image segments and are usually formulated

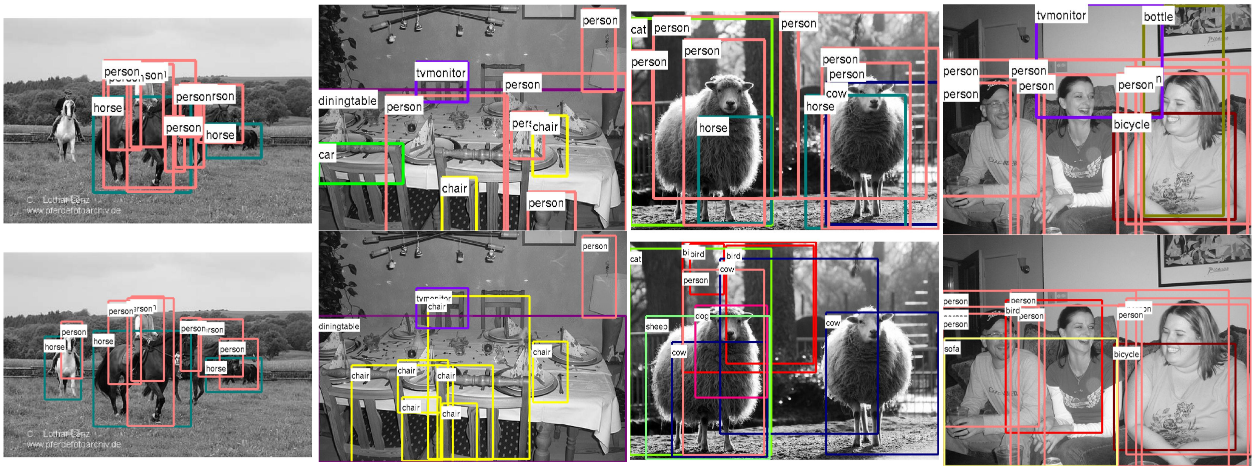


Figure 7. Example test images. On the **top** row, we show the top 10 detections from the baseline model after standard NMS. On the **bottom** row, we show the top 10 marginal detections from our global model. On the **left**, we see that horse and person detections are better localized by the globally tuned NMS model. In the **left center**, our model seems to favor patterns of chair detections that overlap, as maybe common in scenes of tables. In the **right center**, our model exploits co-occurrence cues favoring groups of animals. Finally, on the **right**, our model appears to be exploiting relational cues about sofas and people while enforcing mutual exclusion between the bottle and people detections.

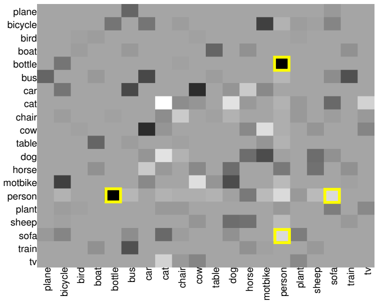


Figure 5. We visualize the weights for our overlap threshold across all our models. Light areas correspond to an increase in score. The structure in these weights indicate the subtlety required for applying mutual exclusion across classes. For example, because people and bottles have similar shapes, the local detectors we use [8] can confuse them. Our global model learns to strongly compete such overlapping detections using a negative weight. However, people and sofas tend to overlap because people partially occlude sofas when sitting down. In this case, we learn a positive weight that reinforces both detections.

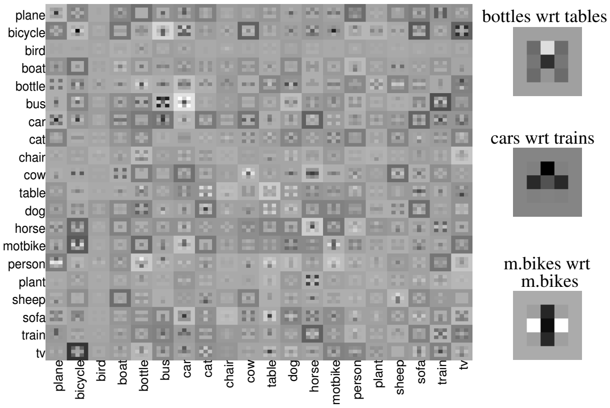


Figure 6. We visualize the pairwise spatial weights for each pair of classes as a 5×5 image (analogous to Fig3). Light areas indicate a favorable arrangement. We show a closeup for particular relations from classes where the global model helps performance. On the **top**, we see that bottles tend to sit above tables. In the **middle**, **cars** lie both near and far from trains, but rarely above or directly next to them. On the **bottom**, we see that motorbikes tend to occur next to one another in images.

as a CRF. Such CRFs for pixel/segment labeling use singleton potential features that capture local distributions of color, textons, or visual words. Pairwise potentials incorporate the labelings of neighboring pixels but in contrast to older work on MRFs these pairwise potentials may span a very large set of neighboring sites (e.g. [22, 24]). Learning such complicated potentials is a difficult problem and authors have relied primarily on boosting [19, 22, 24] to do feature selection in a large space of possible potential functions.

These approaches are appealing in that they can simultaneously produce a segmentation and detection of the objects in a scene. Thus they automatically enforce NMS and hard mutual exclusion (although as our examples show, this may not be entirely desirable). However, the discriminative power of these methods for detection seems limited. While local image features work for some object classes (grass, sky etc), a clear difficulty with the pixel/segment labeling approach is that it is hard to build features for objects defined primarily by shape. It still remains to be shown whether such approaches are competitive with scanning window templates on object detection benchmarks.

In principle, one could define unary potentials for CRFs using, say, HOG templates centered on individual pixels. However, the templates must score well when centered on every pixel within a particular segment. Thus templates will tend to be overly-smoothed. Our method is fundamentally different in that the output is **sparse**. A complete object detection is represented by the activation of a single pixel and so the unary potential can be quite strong. Furthermore, a detection in our model represses detections corresponding to small translations while, in the pixel labeling model, exactly the opposite has to happen. We thus make a tradeoff, moving to more powerful discriminative unary features but sacrificing tractable pairwise potentials.

Alternatively, [10, 15] group pixels into object-sized segments and then define a CRF over the labels of the segments. This approach has the advantage that unary potentials can now be defined with object templates, say, centered on the segment. However, the initial segmentation must be fairly accurate and enforces NMS and mutual exclusion without object-level layout models.

To our knowledge, the problem of end-to-end learning of multi-object detection (i.e. learning NMS) has not been explored. The closest work we know if is that of [5] who use structured regression to predict the bounding box of a single detection within an image. Both models are trained using images rather an cropped windows. Both are optimized using the structural SVM formalism of [23]. However, the underlying assumptions and resulting models are quite different. In the regression formalism of [5], one assumes that each training image contains a single object instance, and so one cannot leverage information about the layout of multiple object instances, beit from the same class or not. The models may not perform well on images without the object because such images are never encountered during training. In our model, we can use all bounding-box labels from all training images, including those that do not contain any object, to train a model that will predict those very labels.

8. Conclusion

We have presented a system for multi-class object detection with spatial interactions that can be efficiently trained in a discriminative, end-to-end manner. This approach is able to fuse the outputs of state of the art template based object detectors with information about contextual relations between objects. Rather than resorting to post-processing to clean up detections, our model learns optimal non-max suppression parameters and detection thresholds for each class. The resulting system outperforms published results on the PASCAL VOC 2007 object detection dataset.

Acknowledgements: Funding for this research was provided by National Science Foundation award IIS 0812428.

References

[1] http://cmp.felk.cvut.cz/~xfrancv/libqp/html. 5

[2] http://people.cs.uchicago.edu/~pff/latent. 5, 6

[3] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR*, pages II: 169–176, 2005. 6

[4] R. Baur, A. A. Efros, and M. Hebert. Statistics of 3d object locations in images. Technical Report CMU-RI-TR-08-43, Robotics Institute, Pittsburgh, PA, October 2008. 2, 6

[5] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, pages 2–15, Berlin, Heidelberg, 2008. Springer-Verlag. 4, 8

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005. 1, 2

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop. 2, 5, 6

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 1, 5, 6, 7

[9] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *Proceedings of the 25th international conference on Machine learning*, pages 304–311. ACM New York, NY, USA, 2008. 5

[10] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, Anchorage, AK, 2008. 2, 6, 8

[11] E. Hall. *The Hidden Dimension*. Anchor Books, 1966. 2

[12] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, volume 2. IEEE Computer Society; 1999, 2004. 2, 6

[13] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80(1):3–15, 2008. 2, 6

[14] T. Joachims, T. Finley, and C. Yu. Cutting plane training of structural SVMs. *Machine Learning*, 2009. 1, 2, 4

[15] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 2, 2005. 2, 8

[16] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 17–32, 2004. 3

[17] Y. Liu, W. Lin, and J. Hays. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics*, 23(3):368–376, 2004. 2

[18] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. *NIPS*, 16, 2003. 2

[19] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *Lecture Notes in Computer Science*, 3951:1, 2006. 6, 7

[20] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, pages II: 1331–1338, 2005. 6

[21] C. Teo, A. Smola, S. Vishwanathan, and Q. Le. A scalable modular convex solver for regularized risk minimization. In *SIGKDD*. ACM New York, NY, USA, 2007. 4, 5

[22] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. *NIPS*, 2004. 2, 6, 7

[23] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*. ACM New York, NY, USA, 2004. 4, 8

[24] Z. Tu. Auto-context and its application to high-level vision tasks. In *CVPR*, 2008. 7

[25] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 1